

SOLAR Testing Toolkit

Руководство пользователя

Версия 3.1.0

Дата

10.11.2023

Оглавление

1	0 приложение SOLAR Testing Toolkit	9
2	Работа с интерфейсом	10
2.1	Аутентификация	10
2.2	Панель управления	10
2.3	Строка навигации	11
2.4	Меню	11
2.5	Табличные формы	12
2.6	Формы смешанного формата	13
2.7	Всплывающие окна приложения	14
3	Основные объекты	16
3.1	Взаимодействие объектов SOLAR Testing Toolkit	16
3.2	Тестовые шаги	17
3.2.1	Направление тестового шага	17
3.2.2	Порядок выполнения тестовых шагов	17
3.2.3	Выполнение исходящего тестового шага	18
3.2.4	Выполнение входящего тестового шага	19
3.3	Основные схемы взаимодействия приложения и внешней системы	19
4	Конфигурация приложения	21
4.1	Каналы	21
4.1.1	Создание канала	21
4.1.2	Режим отправки в настройках канала	28
4.1.3	Настройка правил выбора входящего сообщения	29
4.1.4	Настройка шаблона Url для канала на основе диалекта HTTP	31
4.2	Функциональные области	31
4.3	Системы	33
4.3.1	Создание системы	33
4.3.2	Архивация версии системы	33
4.4	Окружения	33
4.4.1	Создание окружения	34
4.4.2	Экспорт окружения	34
4.4.3	Импорт окружения	34
4.5	Шаблоны действия	35
4.5.1	Создание группы шаблонов	35
4.5.2	Создание шаблона	36
4.5.3	Добавление шаблона к группе	38
4.5.4	Копирование шаблона	38
4.5.5	Экспорт шаблона	38

4.5.6 Импорт шаблона	39
4.6 Наборы параметров	40
4.6.1 Создание набора параметров	40
4.6.2 Загрузка набора параметров из базы данных	41
4.6.3 Экспорт набора параметров	42
4.6.4 Импорт набора параметров	43
4.7 Пользовательские настройки (окружение)	43
4.8 Управление пользователями и правами доступа	43
4.8.1 Изменение пароля и данных учетной записи пользователем	43
4.8.2 Стандартные группы пользователей	44
5 Работа с проектами	46
5.1 Создание проекта	46
5.1.1 Копирование проекта	47
5.1.2 Экспорт проекта	47
5.1.3 Импорт проекта	48
5.2 Привязка набора параметров к проекту	49
5.3 Тестовые сценарии	50
5.3.1 Режим запуска сценариев с блокировкой	50
5.3.2 Приоритеты и порядок выполнения сценариев	51
5.3.3 Создание тестового сценария	51
5.3.4 Копирование тестового сценария	53
5.3.5 Экспорт тестового сценария	53
5.3.6 Импорт тестового сценария	54
5.3.7 Разметка сценариев с помощью функциональных областей	55
5.4 Тестовые шаги	55
5.4.1 Создание тестового шага	55
5.4.2 Создание тестового шага на основе шаблона	59
5.4.3 Создание финального тестового шага	60
5.4.4 Копирование шагов	60
5.4.5 Соответствие тестового шага на основе шаблона и вызываемого шаблона	61
5.4.6 Контекстные переменные	61
5.4.7 Проверки	63
5.4.8 Контекстные функции	66
5.4.9 Типы шагов	71
5.4.10 Использование шаблона действия	82
5.5 Планы выполнения	83
5.5.1 Создание плана выполнения	84
5.5.2 Переопределение набора параметров	85
5.5.3 Запуск плана выполнения	86

5.6 Требования	86
5.6.1 Создание требования	88
5.6.2 Импорт требований	88
5.6.3 Формирование отчета о покрытии требований	89
5.7 Форматирование	89
5.7.1 Форматирование даты и времени	89
5.7.2 Форматирование чисел	89
5.8 Запуск проекта	90
5.9 Отчеты о выполнении	90
5.9.1 Формирование отчета о выполнении	91
5.10 Просмотр истории изменений	91
6 Работа со сценарием	92
6.1 Запуск сценария	92
6.2 Просмотр результатов запуска	93
6.3 Анализ результатов запуска	95
6.3.1 Статус выполнения	95
6.3.2 Детали выполнения сценария	96
6.3.3 Детали выполнения шага	97
6.4 Удаление устаревших результатов запуска	100
6.4.1 Поддерживаемые таблицы	101
7 Работа с файлами	102
7.1 Загрузка файлов	102
7.1.1 Основные сведения о загрузке файлов	102
7.1.2 Настройка загрузки файлов	102
7.1.3 Поддерживаемые форматы загружаемых файлов	104
7.1.4 Загрузка файла и проверка результатов	112
7.1.5 Проверки загруженных файлов	112
7.1.6 Настройка валидации XML-файлов по XSD-схеме	119
7.2 Выгрузка файлов в файловую систему	120
7.2.1 Поддерживаемые форматы выгружаемых файлов	120
7.2.2 Выгрузка файла произвольного формата	120
7.2.3 Выгрузка файла заданного формата	120
7.3 Проверка наличия файла в файловой системе	121
8 Поддержка криптографических действий	122
8.1 Тестовый ключ LMK	122
8.2 Функции deriveIccSessionKey и deriveIccMasterKey	125
8.3 Функции rawGenerateAC и isoGenerateAC	126
8.4 Функция svc3	127

8.5 Функция encryptClearPinBlockIso0	128
9 Нагрузочное тестирование	129
9.1 Основные понятия	129
9.2 Настройка профиля нагрузки	130
9.2.1 Создание нагрузочного проекта	130
9.2.2 Привязка набора параметров	130
9.2.3 Нагрузочный сценарий	130
9.2.4 Создание нагрузочного сценария	131
9.3 Проверки метрик производительности	132
9.3.1 Настройка проверок метрик производительности	133
9.3.2 Пример проверки метрики «Минимальное количество операций в секунду»	133
9.4 Просмотр результатов выполнения	134
9.5 Динамическое управление нагрузкой	135
9.6 Отчет о выполнении проекта	136
9.7 Копирование нагрузочного проекта	136
10 Compliance-тестирование процессинговой системы	137
11 Поддержка протокола NDC	138
11.1 Настройка проекта тестирования	138
11.2 Данные шага для преобразования в формат NDC	138
11.3 Перечень поддерживаемых команд NDC	139
11.4 Перечень управляющих команд	139
11.5 Примеры эмуляции команд	140
11.5.1 Пример команды Transaction Request Message для операции Cash-in	140
11.5.2 Пример команды Transaction Request Message для выдачи наличных	140
11.5.3 Ready Status Message (Transaction Reply Confirmation)	140
11.5.4 Ready Status Message (Host Command Confirmation)	141
11.5.5 Пример Solicited message (Send Basic Supply Counters Message)	141
11.5.6 Пример Unsolicited message (Change supervisor sensor status)	141
11.5.7 Переключение между режимами работы	142
12 Работа с клиринговыми файлами	143
12.1 Настройка загрузки и выгрузки клиринговых файлов	143
12.1.1 Настройка зависимостей в файле pom.xml	143
12.1.2 Создание канала	144
12.2 Структура клирингового файла Mastercard	144
12.2.1 Поля клирингового диалекта Mastercard	146
12.3 Структура клирингового файла МИР	147
12.3.1 Поля клирингового диалекта МИР	148
12.4 Структура клирингового файла Visa	150

12.4.1 Поля клирингового диалекта Visa	151
12.5 Структура клирингового файла UPI	151
12.5.1 Поля клирингового диалекта UPI	160
12.6 Выгрузка клирингового файла	161
12.7 Загрузка клирингового файла	161
13 ISO Responder	162
13.1 Настройка работы с авторизационным респондером	162
13.2 Настройка правила генерации ответа ISO 8583	162
13.3 Настройка онлайн-детокенизации	165
13.4 Формирование ответа в режиме онлайн-детокенизации	165
13.5 Просмотр журнала сообщений ISO 8583	166
13.6 Загрузка клиринга	167
13.7 3DS1 Responder	169
13.8 Настройка работы с 3DS1 Responder	169
13.9 Настройка правила генерации ответа 3DS1	169
13.10 Примеры сообщений VERes	172
13.11 Примеры сообщений PARes	174
13.12 Просмотр журнала сообщений 3DS1	177
13.13 3DS2 Responder	180
13.13.1 Настройка правила генерации ответа 3DS2	180
13.13.2 Примеры ARes	183
13.13.3 Примеры сообщений RReq	186
13.13.4 Пример CRes	188
13.13.5 Просмотр журнала сообщений 3DS2	188
Приложение А. Справочник контекстных функций	193
А.1 Общие контекстные функции платформы SOLAR	193
A.1.1 add	193
A.1.2 caseOf	193
A.1.3 concat	193
A.1.4 divide	193
A.1.5 lowerCase	194
A.1.6 multiply	194
A.1.7 subtract	194
A.1.8 upperCase	194
A.1.9 abs	194
A.1.10 addDays	195
A.1.11 dateTime	195
A.1.12 format	195

A.1.13 listOf	196
A.1.14 padCenter	196
A.1.15 padLeft	196
A.1.16 padRight	197
A.1.17 repeat	197
A.1.18 substring	197
A.1.19 subtractDays	197
A.2 Основные контекстные функции SOLAR Testing Toolkit	198
A.2.1 base64Decode	198
A.2.2 base64Encode	198
A.2.3 convert	198
A.2.4 deflateDecode	199
A.2.5 deflateEncode	199
A.2.6 random	199
A.2.7 sequence	199
A.2.8 sysdate	200
A.2.9 unicodeDecode	200
A.2.10 urlDecode	201
A.2.11 urlEncode	201
A.2.12 uuid	201
A.2.13 xPath	201
A.3 Контекстные функции для тестирования ISO 8583	202
A.3.1 isoGenerateField	202
A.4 Контекстные функции для тестирования 3DS 1.0	202
A.4.1 pares	202
A.4.2 pareqEncode	204
A.4.3 paresDecode	204
A.5 Контекстные функции для тестирования 3DS 2.0	205
A.5.1 base64UrlDecode	205
A.5.2 base64UrlEncode	205
A.5.3 generateEcdhKeyPair	205
A.5.4 buildEcdhSharedSecretKey	205
A.5.5 buildAcsSignedContent	206
A.5.6 parseAcsSignedContent	206
A.5.7 encodeAppCreq	206
A.5.8 decodeAppCreq	207
A.6 Контекстные функции модуля криптографии	207
A.6.1 cvc3	207
A.6.2 deriveIccSessionKey	208

A.6.3 deriveIccMasterKey	208
A.6.4 encryptClearPinBlockIso0	209
A.6.5 isoGenerateAC	209
A.6.6 rawGenerateAC	210
Приложение В. Интерфейс saveParameterSetRowBatch	212
В.1 Состав заголовка XML-сообщения	212
В.2 Состав тела сообщения saveParameterSetRowBatchRequest	213
В.3 Структура файла saveParameterSetRowBatchRequest	214
В.4 Настройка взаимодействия с интерфейсом saveParameterSetRowBatch	215
История изменений документа	216

1 О приложении SOLAR Testing Toolkit

SOLAR Testing Toolkit — это многофункциональное приложение (далее — приложение), позволяющее осуществлять следующие виды тестирования процессинговой системы с помощью интерфейсов и API, используемых в промышленной работе:

- функциональное,
- регрессионное,
- нагрузочное,
- проверка соответствия требованиям платежных систем.

Приложение предоставляет следующие возможности:

- генерация сообщений и файлов,
- получение, разбор и проверка входящего потока информации,
- поддержка криптографических действий,
- загрузка и выгрузка файлов.

Поддерживаемые интерфейсы:

- Онлайн интерфейсы:
 - ISO 8583, например, ISO Host-2-Host, VISA BASE I, MasterCard (входящие и исходящие сообщения), Mir;
 - XML-сообщения (в том числе SOAP) через HTTP/JMS (входящие и исходящие сообщения),
 - Вызов REST API через HTTP/JMS,
 - DB Adapter,
 - gRPC Adapter.
- Офлайн интерфейсы:
 - MC Clearing, Mir Clearing, VISA BASE II (загрузка и выгрузка файлов),
 - File In, File Out, RAW Out (загрузка и выгрузка файлов).

2 Работа с интерфейсом

2.1 Аутентификация

Для получения доступа к приложению требуется пройти аутентификацию. Для этого введите логин и пароль учетной записи, зарегистрированной в приложении (см. рисунок 1). Если вы не получили логин и пароль, обратитесь к администратору, который отвечает за внедрение приложения в вашу информационную систему.

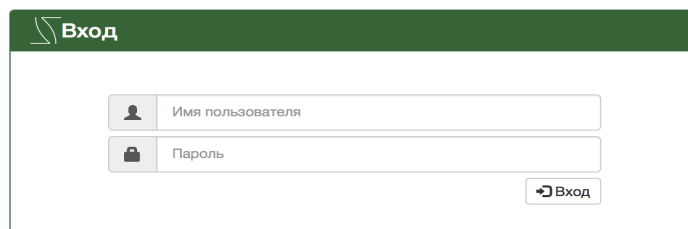


Рисунок 1 Окно авторизации

Для завершения работы под текущей учетной записью нажмите кнопку **Выход** в раскрывающемся списке в правом верхнем углу страницы, как показано на рисунке 2. Чтобы исключить несанкционированный доступ или непреднамеренное изменение данных, необходимо выходить из учетной записи после каждого сеанса работы с приложением.

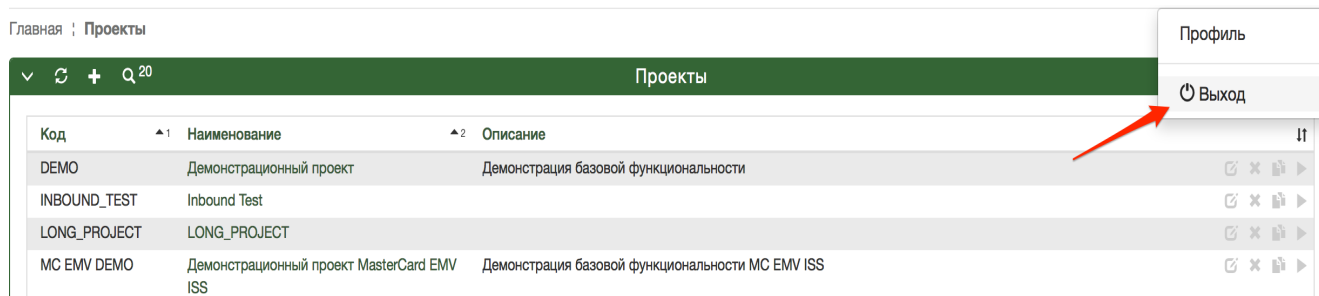


Рисунок 2 Завершение сеанса работы

2.2 Панель управления

Панель управления находится в верхней части окна приложения, как показано на рисунке 3.

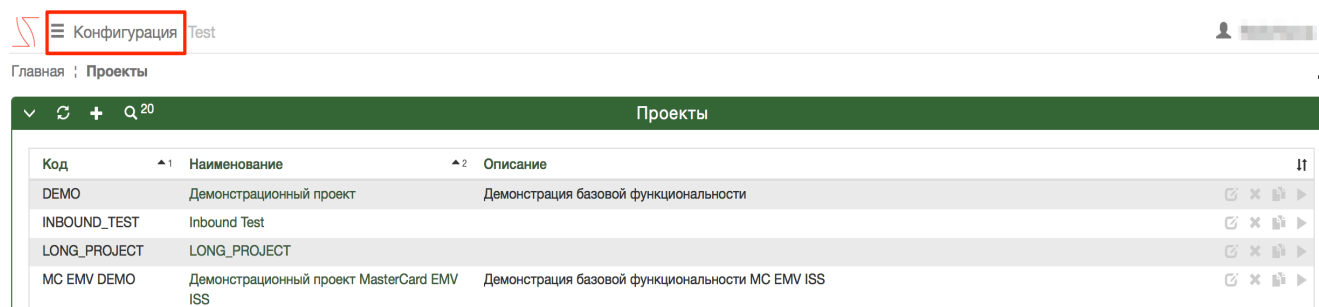


Рисунок 3 Панель управления

В панели управления находятся следующие элементы:

- меню;
- название окружения, в котором происходит работа. Например, тестовое, производственное или другое — в зависимости от настроек, установленных администратором вашей информационной системы (необязательное поле);
- панель пользователя, в которой отображается имя текущего пользователя, ссылка на профиль и кнопка **Выход**.

Подробнее о меню см. в разделе 2.4 «Меню».

2.3 Строка навигации

Строка навигации, известная также как «хлебные крошки» – элемент управления, предназначенный для быстрой навигации между вложенными объектами интерфейса пользователя.

Строка навигации расположена под панелью управления (см. рисунок 4). Все элементы строки навигации, кроме последнего, являются ссылками, по которым можно перейти к родительским страницам. При запуске приложения строка навигации содержит единственный элемент **Главная**, который является ссылкой на стартовую страницу приложения.

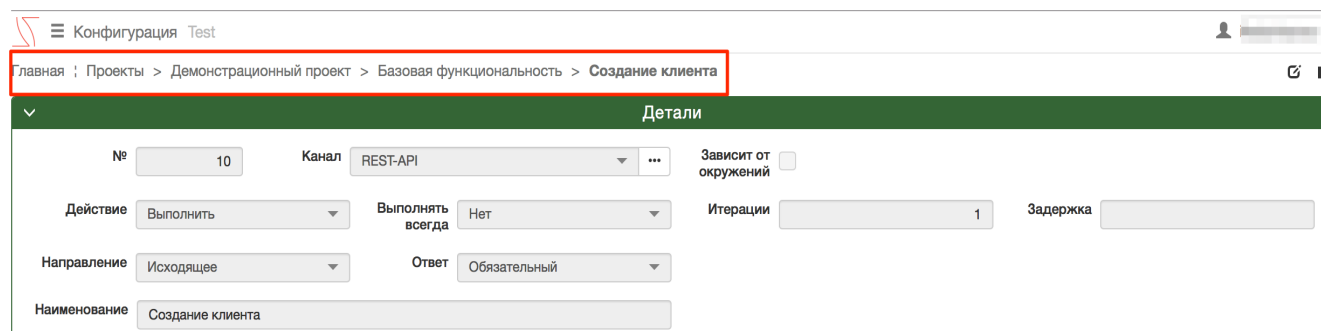


Рисунок 4 Строка навигации

2.4 Меню

Пункты меню сгруппированы в секции в соответствии с их функциональным назначением, которое отображается в заголовке, например, Пользователи.

Меню содержит следующие секции:

- Testing Toolkit,
- Процессы,
- Пользователи.

Меню приложения можно условно разделить на две части — конфигурационную и операционную. К конфигурации относятся все настройки, которые задаются при внедрении приложения в информационную систему и используются в дальнейшей работе. Например, пункты меню **Каналы**, **Диалекты** относятся к конфигурации, а пункт **Проекты** — к операционным данным.

Меню позволяет перейти к следующим страницам:

- Проекты,
- Планы выполнения,
- Шаблоны,
- Наборы параметров,
- Функциональные области,
- Пользовательские настройки,

- Окружения,
- Системы,
- Диалекты,
- Каналы,
- Процессы,
- Пользователи,
- Группы пользователей,
- Роли.

2.5 Табличные формы

Табличные формы или таблицы содержат данные об объектах одного типа, представленные в строках и столбцах. Элементы таблицы называются записями. Примером табличной формы является страница **Тестовые сценарии** (см. рисунок 5).

Код	Наименование	Описание	Действие
BASIC_1	Базовая функциональность	Создание основных объектов: - Клиент - Контракт - Карта - Проведение успешной операции - Проведение операции с отказом (нехватка средств) [раскрыть...]	Выполнить
BASIC_2	Базовая функциональность - тест завершается ошибкой	Создание основных объектов: - Клиент - Контракт - Карта - Проведение успешной операции - Проведение операции с отказом (нехватка средств) [раскрыть...]	Выполнить

Рисунок 5 Табличная форма

При работе с табличной формой доступны следующие действия:

- > — развернуть данные;
- v — свернуть данные;
- ↻ — обновить данные;
- + — добавить запись в таблицу. По кнопке открывается всплывающее окно для ввода данных;
- 🔍 — искать по указанным критериям,
- ✎ — редактировать запись. По кнопке открывается всплывающее окно, в котором можно изменить данные выбранной записи;
- ✖ — удалить строку. По кнопке открывается всплывающее окно, в котором можно подтвердить или отменить удаление строки.
- 👁 — посмотреть детали записи во всплывающем окне.

Подробнее о всплывающих окнах приложения см. в разделе 2.7 «Всплывающие окна приложения».

В некоторых табличных формах доступен поиск элементов по критерию. Для поиска элементов в таблице нажмите кнопку 🔍 (Искать), заполните поля в появившейся форме поиска и снова нажмите 🔍 или <Enter>.

В любом поле формы поиска можно ввести как полную строку, так и подстроку (например, часть кода или наименования). Поиск по подстроке имеет следующий синтаксис: %abc%. Результатом такого поиска являются все элементы, содержащие подстроку abc в указанном поле. Пример показан на рисунке 6.

Код	Наименование	Описание	Действие
BASIC_2	Покупка по чиповой карте. Неуспешная операция. Нет средств	Создание основных объектов: - Клиент - Контракт - Карта - Проведение успешной операции - Проведение операции с отказом (нехватка средств) [раскрыть...]	Выполнить

Рисунок 6 Пример поиска по подстроке

Для получения более подробной информации по объекту используйте следующие способы:

- перейдите по ссылке в колонке **Наименование**, как показано на рисунке 7:

Код	Наименование	Описание
DB	Отправка запроса в БД	Базовая функциональность запросов в БД
INT	Интеграционное взаимодействие	Общее интеграционное взаимодействие
ISO	Работа с ISO-сообщениями	Базовая функциональность отправки/получения ISO сообщений
XML	Работа с Web-Services	Базовая функциональность работы с Web-Services

Рисунок 7 Переход к детальной информации по объекту

- перейдите по ссылке **[раскрыть]**, как показано на рисунке 8:

Код	Наименование	Описание	Действие
BASIC_1	Базовая функциональность	Создание основных объектов: - Клиент - Контракт - Карта - Проведение успешной операции - Проведение операции с отказом (нехватка средств) [раскрыть...]	Выполнить

Рисунок 8 Получение полной информации поля

В случае если таблица состоит из большого числа строк, приложение автоматически разбивает данные по страницам. Для переключения между страницами используйте кнопки в правом нижнем углу (см. рисунок 9).



Рисунок 9 Кнопки навигации по страницам

2.6 Формы смешанного формата

Формы смешанного формата содержат данные разных форматов, например, таблицы, текстовые поля, выпадающие списки и т.д. Примером формы смешанного формата является форма, изображенная на рисунке 10.

Детали

№ Шаблон действия Зависит от окружений

Действие Итерации Задержка

Наименование

Описание

Входящие параметры шаблона

Переменная	Выражение	Тип данных	Описание
pan	\${cardNumber}	Строка	PAN
cardExp	\${expiryDate}	Строка	Срок действия карты

Исходящие параметры шаблона

Переменная	Возвращаемый	Тип данных	Описание	Локал. имя

Рисунок 10 Форма смешанного формата

Для формы смешанного формата доступны следующие действия (см. правый верхний угол формы):

- – редактировать данные,
- – сохранить изменения,
 - В случае успешного сохранения данных появится сообщение «Данные успешно сохранены».
 - В случае ввода некорректных данных появится сообщение «Ошибка сохранения данных».
- – закрыть форму без сохранения,
- — развернуть данные,
- — свернуть данные.

2.7 Всплывающие окна приложения

Редактирование данных в таблице осуществляется при помощи всплывающих окон приложения. При нажатии на кнопки и появляется всплывающее окно, пример которого изображен на рисунке 11, в котором можно ввести или изменить данные записи.

Требование ✕

Детали

Код* Наименование*

Описание

Ответственный*

Сохранить Отмена

Рисунок 11 Всплывающее окно

Используйте следующие кнопки:

- **Сохранить** – сохранить изменения,
- **Отмена** – закрыть форму без сохранения.

После сохранения измененный или добавленный объект выделяется в таблице желтым цветом.

При нажатии на кнопку ✕ появляется всплывающее окно, пример которого изображен на рисунке 12, в котором можно подтвердить или отменить удаление выбранной записи.

Тестовые шаги

№	Наименование	Описание	Окружения	Действие	
10	Создание клиента	Создание клиента по имени, фамилии и номеру документа. Проверяется код ответа. Для дальнейшего использования сохраняется номер клиента из ответного сообщения.		Выполнить	✕ ✖
20	Создание контракта			Выполнить	✕ ✖
30	Создание карты	(подстановка переменных номер клиента и номер контракта). Проверяется код ответа. Для дальнейшего использования сохраняется номер карты и срок действия из ответного сообщения.		Выполнить	✕ ✖

Подтверждение

Удалить Создание контракта?

Удалить Отмена

Рисунок 12 Подтверждение операции удаления

3 Основные объекты

3.1 Взаимодействие объектов SOLAR Testing Toolkit

Основные объекты приложения SOLAR Testing Toolkit делятся на две части: данные конфигурации и данные проектов. К конфигурации относятся сущности, которые используются во всех проектах: диалекты, каналы, функциональные области и шаблоны действия.

Внутри проекта вы можете создавать тестовые сценарии, проверяющие работу процессинговой системы на соответствие Требованиям. Сценарий состоит из одного или нескольких шагов. Можно создать уникальный шаг сценария или же использовать шаблон действия — сохраненную последовательность шагов, выполняющих базовые действия над процессинговой системой. Шаблон действия можно многократно использовать в разных сценариях. Примерами шаблона действия являются закрытие дня, создание финансового документа или отправка авторизационного сообщения.

Шаг шаблона обладает всеми свойствами шага сценария и может вызывать другой шаблон.

Вы можете верифицировать результаты работы сценария после выполнения каждого шага. Для этого выполните проверку данных, полученных из процессинговой системы.

Разметка сценариев с помощью функциональных областей позволит вам повысить эффективность регрессионного тестирования.

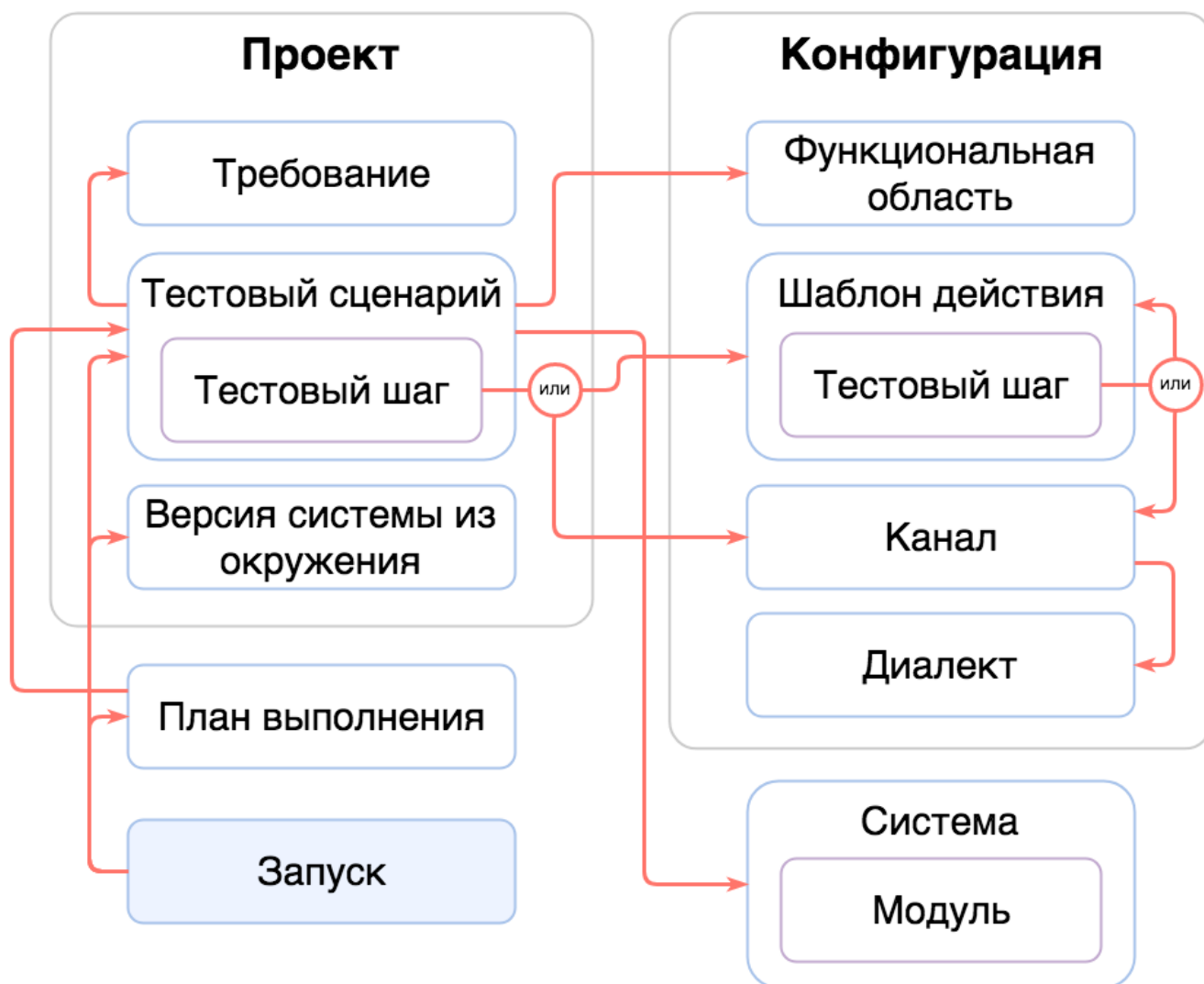


Рисунок 13 Взаимодействие объектов SOLAR Testing Toolkit

3.2 Тестовые шаги

3.2.1 Направление тестового шага

Приложение поддерживает входящие и исходящие тестовые шаги:

- *Исходящие тестовые шаги* позволяют формировать и отправлять сообщения во внешнюю систему и проверять ответные сообщения.
- *Входящие тестовые шаги* позволяют получать и проверять сообщения из внешней системы, формировать и отправлять ответные сообщения.

Как правило, входящий тестовый шаг используется в сочетании с исходящим шагом (см. 3.3 «Основные схемы взаимодействия приложения и внешней системы»).

3.2.2 Порядок выполнения тестовых шагов

Стандартный режим выполнения тестовых шагов предусматривает их последовательное выполнение, когда следующий шаг запускается только после завершения текущего. Если текущий шаг завершается с ошибкой, оставшиеся шаги сценария пропускаются, выполняется финальный шаг, и сценарий завершается с ошибкой.

Режим отложенного выполнения меняет порядок выполнения шагов. При запуске тестового шага с отложенным выполнением он запускается в параллельном «потоке», и

сразу же начинает выполняться следующий тестовый шаг (см. рисунок 14). Шаг с отложенным выполнением можно использовать для тестирования сложного взаимодействия, включающего несколько систем. Если такой шаг завершается с ошибкой, оставшиеся тестовые шаги выполняются до конца сценария (или шаблона — что наступит раньше), после чего сценарий завершается с ошибкой.

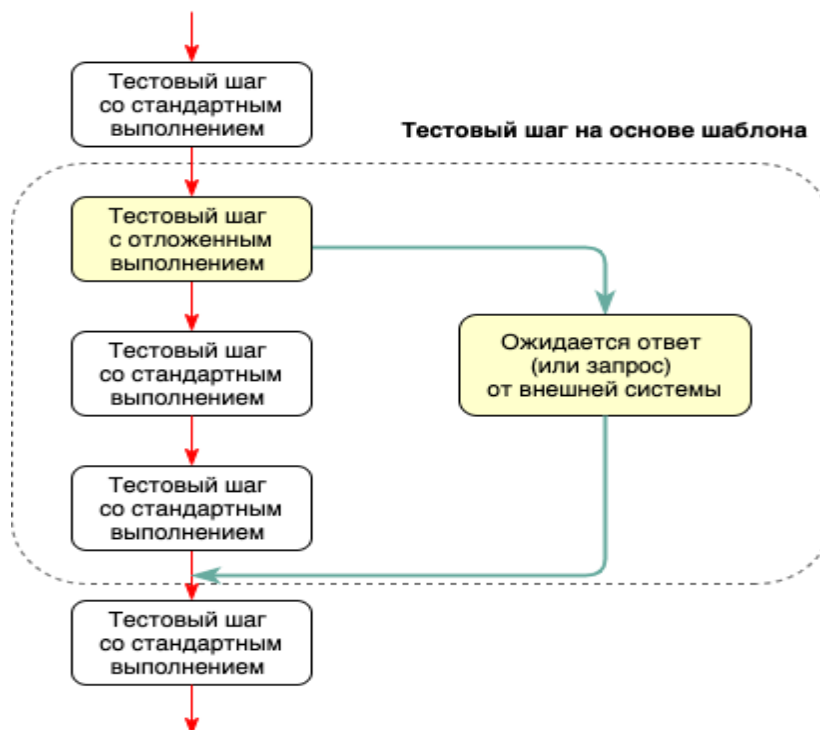


Рисунок 14 Схема выполнения тестовых шагов со стандартным и отложенным выполнением

Обратите внимание: данные, полученные из внешней системы на шаге с отложенным выполнением, не могут быть использованы в других шагах сценария или шаблона. Однако вы можете сохранить такие данные в контекстные переменные и использовать их по завершении шаблона.

3.2.3 Выполнение исходящего тестового шага

Исходящий тестовый шаг выполняется по следующему алгоритму:

1. Если в текущем шаге настроено отложенное выполнение, начинает выполняться следующий шаг, а текущий шаг выполняется параллельно с ним (см. рисунок 14).
2. Ответ по шагу с отложенным выполнением ожидается в течение установленного времени (или до завершения сценария или шаблона — что наступит раньше).
3. В соответствии с данными шага и контекстом приложение формирует сообщение для внешней системы.
4. Сообщение отправляется во внешнюю систему в соответствии с форматом канала, который указан для тестового шага.
5. При получении ответного сообщения полученные данные проверяются на соответствие параметрам, указанным для тестового шага в форме **Проверки**.
6. Если все проверки прошли успешно:
 - a. данные сохраняются в контекстные переменные (если настроена установка переменных);
 - b. шаг успешно завершается.

7. Тестовый шаг завершается с ошибкой в следующих случаях:
 - a. обязательный ответ из внешней системы не получен в течение ожидаемого времени;
 - b. полученные данные не прошли проверки.

3.2.4 Выполнение входящего тестового шага

Входящий тестовый шаг выполняется по следующему алгоритму:

1. Если в текущем шаге настроено отложенное выполнение, начинает выполняться следующий шаг, а текущий шаг выполняется параллельно.
2. Запрос из внешней системы ожидается в течение времени, установленного для тестового шага в поле **Время ожидания**.
3. При получении запроса из внешней системы приложение определяет канал, по которому поступили данные.
4. В соответствии с правилами, указанными для канала в форме **Правила выбора входящего сообщения**, проверяются параметры полученного сообщения и определяется тестовый шаг, в который нужно передать данное сообщение:
 - если найдено несколько подходящих тестовых шагов, все такие шаги завершаются с ошибкой;
 - если найден один подходящий тестовый шаг, сообщение передается в данный шаг.
5. Полученные данные проверяются на соответствие параметрам, установленным для тестового шага в форме **Проверки**.
6. Если все проверки прошли успешно:
 - a. данные сохраняются в контекстные переменные (если настроена установка переменных);
 - b. в соответствии с данными шага и контекстом формируется ответ для внешней системы;
 - c. ответ отправляется во внешнюю систему в соответствии с форматом канала, который указан для тестового шага;
 - d. шаг успешно завершается.
7. Тестовый шаг завершается с ошибкой в следующих случаях:
 - полученные данные не прошли проверки;
 - ответ не может быть отправлен.

Для получения ответа от приложения Testing Toolkit на входящее сообщение как от сторонней системы необходимо обеспечить корректный порядок выполнения шагов:

- (10) входящее сообщение с отложенным выполнением, в котором формируется необходимый ответ,
- (20) исходящее сообщение (запрос от внешней системы), при получении которого подготовленный ответ отправляется во внешнюю систему.

Причем, два этих шага должны находиться в одном шаблоне или быть отдельными шагами одного сценария, выполняемыми в указанном порядке.

3.3 Основные схемы взаимодействия приложения и внешней системы

Взаимодействие приложения и внешней системы (см. рисунок 15) реализуется с помощью сочетания направления (входящее, исходящее) и режима выполнения шагов (стандартное или отложенное выполнение):

- Эмуляция стороны, формирующей поток информации для внешней системы (например, эмуляция сообщений от POS-терминала). В этом случае используется исходящие шаги со стандартным порядком выполнения.
- Эмуляция стороны, принимающей и обрабатывающей поток информации из внешней системы (например, эмуляция ответов платежной системы). В этом случае используются входящие шаги со стандартным порядком выполнения.
- Эмуляция окружения внешней системы с исходящими и входящими потоками информации. В этом случае используется сочетание исходящего шага с отложенным выполнением и входящего шага со стандартным порядком выполнения. На исходящем шаге отправляется сообщение во внешнюю систему для вызова определенной реакции данной системы (например, запроса к другой системе). А на входящем шаге принимается запрос от внешней системы, который послужил реакцией на исходящий шаг, и формируется ответ.



Рисунок 15 Схемы взаимодействия приложения и внешней системы

4 Конфигурация приложения

4.1 Каналы

Список каналов, тестируемых с помощью приложения, доступен в меню **Конфигурация** → **Каналы**. Настройка каналов выполняется исключительно с помощью пользовательского интерфейса, включая создание и редактирование соответствующих адаптеров и настройку их свойств, например, синхронный или асинхронный режим обработки сообщений и др. Правила маршрутизации сообщений в направлении к и от адаптеров создаются приложением автоматически.

Главная : Каналы

Код	Наименование	Диалект	Описание	
h2h	Host-2-Host	Host-2-Host	Онлайн межхостовой интерфейс	
ib	Internet Bank WS	MasterCard	Канал для работы с приложением Internet Bank (на Web Services)	
ivr	Way4Gate XML (IVR_P2P)	MasterCard	Канал для работы с приложением IVR_P2P (UFХ)	
mc	MasterCard	MasterCard	Онлайн интерфейс к МПС MasterCard	
nspkmc	MasterCard NSPK	MasterCard	MasterCard NSPK	
nspkvisa	VISA ISO NSPK	Visa VIP		
ors	ORS ISO H2H	ORS	ORS канал	
procDB	DataBase	JDBC	Прямой доступ в БД	
restOut	REST-API	XML	Онлайн XML WS взаимодействие	
SolarCrmApi	SolarCrmApi	XML		
testChannel	Test Channel	XML		
visa	Visa VIP	Visa VIP	Онлайн интерфейс к МПС VISA	
W4P2H	Test W4P2H	W4 P2H		
way4gate	Way4Gate XML	XML	Канал для работы с приложением way4gate (UFХ)	
way4u	Way4u XML (IFX)	MasterCard	Канал для работы с приложением way4u (IFX)	

Рисунок 16 Каналы

4.1.1 Создание канала

Чтобы создать канал:

1. Добавьте запись в форму **Каналы** и укажите данные, описанные в таблице 4-1.
2. Нажмите **↓** (Сохранить).

Пример канала с заданными параметрами приведен на рисунке 17.

Канал

Детали

Код* mc Наименование* MasterCard

Система* SOLAR Диалект* ISO 8583 Режим отправки* Синхронный Режим ответа

Описание: Онлайн интерфейс к МПС MasterCard

Дополнительные настройки

Диалект* mastercard

Сохранить Отмена

Рисунок 17 Создание канала

Таблица 4-1 Параметры канала

Поле	Описание
Код	Код канала, который может быть использован внешней системой при управлении SOLAR Testing Toolkit через веб-сервис. Код должен быть уникальным в рамках каналов приложения
Наименование	Название канала, отображаемое в формах
Система	Система, с которой взаимодействует канал
Диалект	Диалект, который используется данным каналом. Выбирается из предопределенного списка
Режим отправки	Поле с выбором из двух значений: «Синхронный» и «Асинхронный». Зависит от типа интерфейса. Подробнее см. в разделе 4.1.2
Режим ответа	<p>Параметр актуален для каналов, используемых на входящих шагах. Поле с выбором из двух значений:</p> <ul style="list-style-type: none"> • «Из сценария» (по умолчанию) — ответ формируется в соответствии со сценарием, • «Автоматический» — ответ генерируется компонентой ISO Responder в соответствии с настроенными правилами (опция доступна только для ISO-каналов)
Время ожидания ответа (мс)	<p>Параметр актуален для каналов, используемых на входящих шагах — период времени, по истечении которого отправляется ответ.</p> <p>Значение должно быть больше либо равно максимальной задержке ответа, выбранной для шагов, использующих данный канал.</p> <p>Если не указано, используется значение по умолчанию, равное 15 с</p>
Описание	Описание назначения канала
Дополнительные настройки: Диалект	<p>Диалект ISO (используется только для ISO 8583):</p> <ul style="list-style-type: none"> • visavip, • visa-sms, • mastercard, • mir-online и т.д.
Дополнительные настройки: Тип сообщения	<p>Используется для каналов на базе диалектов «Загрузка файла» и «Выгрузка файла»:</p> <ul style="list-style-type: none"> • формат файла (Clearing, Delimiter-separated value (DSV), HTML, JSON, Plain text, XLSX, XML)

Поле	Описание
Дополнительные настройки: Тип подключения	Используется для каналов на основе диалекта HTTP и ISO 8583. Тип подключения приложения к каналу: <ul style="list-style-type: none"> • Клиент — приложение отправляет сообщения в канал, • Сервер — приложение получает сообщения из канала
Дополнительные настройки: VISA Source Station ID	Используется для каналов на основе диалектов ISO 8583 visa-vip и visa-sms. Значение Source Station ID (поле 6 из заголовка сообщения)
Дополнительные настройки: VISA Destination Station ID	Используется для каналов на основе диалектов ISO 8583 visa-vip и visa-sms. Значение Destination Station ID (поле 5 из заголовка сообщения)
Дополнительные настройки: Автоматически заполнять поля «7», «11», «37»	Используется для каналов на основе диалектов ISO 8583. Флажок включает и отключает автоматическое заполнение полей 7, 11, 37 при отправке запроса
Команда Sign-On	Используется для каналов на основе диалекта ISO 8583. Команда Sign-On является опциональной. Если команда используется каналом, следует установить флажок «Включена». Форма для настройки команды Sign-On, используемой при установлении связи с хостом. Поле/тег заполняется приложением автоматически в соответствии с диалектом ISO, значение выбирается в форме. <ul style="list-style-type: none"> • Время ожидания (сек) — период ожидания установления соединения, • Режим инициации — параметр указывает, как иницируется соединение с хостом: <ul style="list-style-type: none"> ◦ Ожидать команду — хост отправляет команду, ◦ Отправлять команду — приложение отправляет команду
Команда Sign-Off	Используется для каналов на основе диалекта ISO 8583. Форма для настройки команды Sign-Off, используемой при отключении от хоста. Поле/тег заполняется приложением автоматически в соответствии с диалектом ISO, значение выбирается в форме

Поле	Описание
Команда Send-Advices	Используется для каналов на основе диалекта ISO 8583. Форма для настройки команды Send-Advices, которая отправляет advice-сообщения при установке соединения с хостом. Поле/тег заполняется приложением автоматически в соответствии с диалектом ISO, значение выбирается в форме
Команда Ping	Используется для каналов на основе диалекта ISO 8583. Форма для настройки команды Ping, используемой для проверки наличия соединения. Поле/тег заполняется приложением автоматически в соответствии с диалектом ISO, значение выбирается в форме. <ul style="list-style-type: none"> Интервал (сек) — периодичность отправки команды Ping в секундах, Отправлять при наличии трафика — параметр управляет отправкой команды Ping при наличии трафика: <ul style="list-style-type: none"> выбран — команда отправляется при наличии трафика, не выбран — команда не отправляется при наличии трафика
Дополнительные настройки: Тип исходящего сообщения	Только для каналов на основе диалекта HTTP — формат тела сообщения, отправляемого по протоколу HTTP (XML, JSON, Простой текст, Ключ-значение, HTML)
Дополнительные настройки: Тип входящего сообщения	Только для каналов на основе диалекта HTTP — формат тела сообщения, отправляемого по протоколу HTTP (XML, JSON, Простой текст, Ключ-значение, HTML)
Дополнительные настройки канала на основе диалекта HTTP (настраиваются индивидуально для окружения на вкладке Конфигурация для окружений)	
Шаблон URL	Шаблон URL, используемый для отправки сообщения. Для подключения «Клиент» может включать порт, если используется порт, отличный от порта по умолчанию
Порт	Порт сервера. Параметр используется для типа подключения «Сервер» Для обеспечения высокой производительности при работе с нагрузочными проектами настоятельно рекомендуется указать порт, отличный от стандартного. Стандартный порт задается в конфигурационном файле context.xml настройкой slq/http/port и используется для обращения к приложению через браузер

Поле	Описание
Дополнительные настройки канала на основе диалекта ISO 8583 (настраиваются индивидуально для окружения на вкладке Конфигурация для окружений)	
Хост	URL хоста. Приложение пытается подключиться к каждому из заданных хостов
Порт	Порт хоста
Алгоритм распределения нагрузки	Алгоритм выбора канала, в который отправляется запрос: <ul style="list-style-type: none"> Первый активный канал — выбирается первый канал, который готов пропускать сообщения, Циклический — канал выбирается последовательно из имеющихся
Переопределить параметры	Параметр позволяет переопределить параметры команд Sign-On, Sign-Off, Send-Advices и Ping, заданные для канала, и использовать на конкретном окружении другие значения
Дополнительные настройки для работы с клиринговыми файлами	
Формат файла	Диалект, определяющий формат клирингового файла: <ul style="list-style-type: none"> Mastercard IPM, VISA CTF, TSYS Prime Мир ОДПТ
Кодировка	Используется для диалекта Mastercard и Мир. Кодировка файла: <ul style="list-style-type: none"> EBCDIC, ASCII
Базовая директория	Параметр задается индивидуально для окружения на вкладке Конфигурация для окружений . Путь к базовой директории, используемой соответствующим адаптером
Дополнительные настройки для загрузки текстовых файлов	
Кодировка	Кодировка файла. Выбирается из предопределенного списка
Дополнительные настройки для загрузки HTML-файлов	
Кодировка	Кодировка файла. Выбирается из предопределенного списка
Дополнительные настройки для загрузки DSV-файлов	
Кодировка	Кодировка файла. Выбирается из предопределенного списка

Поле	Описание
Тип разделителя	<p>Тип разделителя:</p> <ul style="list-style-type: none"> • Точка, • Точка с запятой, • Таб, • Пробел, • Вертикальный бар, • Пользовательский
Пропустить строки в начале	Количество строк в начале файла, не имеющих отношения к формату, которые отрезаются
Пропустить строки в конце	Количество строк в конце файла, не имеющих отношения к формату, которые отрезаются
Заголовок	<p>Наличие заголовка:</p> <ul style="list-style-type: none"> • Отсутствует • Присутствует <p>Если заголовок присутствует, он должен соответствовать формату и быть первой читаемой строкой</p>
Адресация поля	<p>В отсутствие заголовка параметр определяет метод обращения к элементам и структуру (имена тегов) преобразованного файла.</p> <ul style="list-style-type: none"> • По индексу колонки (0, 1, 2, ...) — файл преобразуется к виду <code><file><row><column></column></row></file></code>, • По заданному имени («Имена полей») — файл преобразуется к виду <code><file><row><fieldName1></fieldName1><fieldName2></fieldName2>...</row></file></code>, где <code>fieldName1</code>, <code>fieldName2</code> — заданные имена полей. <p>Для обеспечения удобства при настройке проверок рекомендуется выбрать параметр «По заданному имени», поскольку в этом случае обращение к элементам становится проще</p>
Дополнительные настройки для загрузки XSLX-файлов	
Пропустить строки в начале	Количество строк в начале файла, не относящихся в формату, которые отрезаются
Пропустить строки в конце	Количество строк в конце файла, не относящихся в формату, которые отрезаются

Поле	Описание
Заголовок	Наличие заголовка: <ul style="list-style-type: none"> Отсутствует, Присутствует <p>Если заголовок присутствует, он должен соответствовать формату и быть первой читаемой строкой</p>
Адресация поля	В отсутствие заголовка параметр определяет метод обращения к элементам и структуру (имена тегов) преобразованного файла. <ul style="list-style-type: none"> По индексу колонки (0, 1, 2, ...) — файл преобразуется к виду <code><file><row><column></column></row></file></code> По имени колонки (A, B, C, ...) — файл преобразуется к виду <code><file><row><A>...</row></file></code> По заданному имени («Имена полей») — файл преобразуется к виду <code><file><row><fieldName1></fieldName1><fieldName2></fieldName2>...</row></file></code>, где <code>fieldName1</code>, <code>fieldName2</code> — заданные имена полей
Дополнительные настройки для загрузки XML-файлов	
Кодировка	Кодировка файла. Выбирается из predetermined списка
Дополнительные настройки канала «Сообщение операционной системе» (настраиваются индивидуально для окружения на вкладке Конфигурация для окружений)	
Переменные окружения	Наименование и значение переменных окружения
Аргументы	Аргументы команды
Дополнительные настройки канала на основе диалекта JMS (настраиваются индивидуально для окружения на вкладке Конфигурация для окружений)	
Тип подключения	Тип подключения приложения к каналу: <ul style="list-style-type: none"> Клиент — приложение отправляет сообщения в очередь, Сервер — приложение получает сообщения из очереди
Тип сообщения	Тип сообщения: <ul style="list-style-type: none"> XML, JSON
Тип очереди	Тип очереди: <ul style="list-style-type: none"> IBM, Oracle AQ, Tibco JMS
Очередь	Имя очереди для сообщений

Поле	Описание
Очередь для ответов	Имя очереди для ответов
Логин	Логин пользователя для подключения к очереди
Пароль	Пароль пользователя
Менеджер очередей	Имя менеджера очередей. Параметр используется для очередей типа IBM MQ
Хост	URL сервера. Параметр используется для очередей типа IBM MQ
Порт	Порт сервера. Параметр используется для очередей типа IBM MQ
Канал	Канал очереди. Параметр используется для очередей типа IBM MQ
URL	URL для соединения с сервером базы данных. Параметр используется для очередей типа Oracle AQ
Максимально е кол-во подключений	Максимальное количество подключений к базе данных. Параметр используется для очередей типа Oracle AQ, Tibco JMS

Для редактирования данных канала, в том числе дополнительных настроек, перейдите на страницу канала по ссылке в колонке **Наименование**. Параметры канала, которые настраиваются индивидуально для окружения, доступны по ссылке с именем окружения, расположенной на вкладке **Конфигурация для окружений**.

Если требуемое окружение отсутствует, это означает, что в настройках окружения данный канал не выбран. Добавьте канал в настройки окружения (см. раздел 4.4.1), после чего отредактируйте конфигурацию канала для окружения.

4.1.2 Режим отправки в настройках канала

В данном разделе описаны схемы обмена сообщениями при использовании синхронного и асинхронного режима на входящем и исходящем шагах.

Каждый адаптер при получении запроса должен обработать его и вернуть синхронный ответ. Ответное сообщение содержит код ответа и непосредственно данные (payload).

Синхронный режим отправки на входящем шаге предполагает следующую схему взаимодействия:

1. Приложение принимает входящий запрос.
2. Приложение обрабатывает запрос.
3. Приложение возвращает синхронный ответ с данными из тестового шага.

Асинхронный режим отправки на входящем шаге предполагает следующую схему взаимодействия:

1. Приложение принимает входящий запрос.
2. Приложение обрабатывает запрос.
3. Приложение возвращает синхронный ответ без данных (payload сообщения пустой).
4. Если ответ требуется, приложение дополнительно отправляет отдельный запрос с данными из тестового шага.

На исходящем шаге приложение сначала должно сформировать запрос.

Синхронный режим отправки на исходящем шаге предполагает следующую схему взаимодействия:

1. Приложение формирует запрос с данными из тестового шага.
2. Приложение отправляет запрос в адаптер.
3. Приложение принимает входящий синхронный ответ от адаптера с данными.
4. Приложение обрабатывает ответ (выполняет проверки, устанавливает переменные).

Асинхронный режим отправки на исходящем шаге предполагает следующую схему взаимодействия:

1. Приложение формирует запрос с данными из тестового шага.
2. Приложение отправляет запрос в адаптер.
3. Приложение принимает входящий синхронный ответ от адаптера и игнорирует его.
4. Приложение ожидает новый входящий запрос, в котором содержится payload ответа на первоначальный запрос.
5. Приложение принимает входящий запрос, используя настроенные правила выбора входящего сообщения.
6. Приложение интерпретирует его как входящий асинхронный ответ на первоначальный запрос.
7. Приложение обрабатывает асинхронный ответ (выполняет проверки, устанавливает переменные).

Выбор режима отправки зависит от диалекта.

- HTTP — синхронный,
- ISO 8583 — асинхронный,
- работа с файлами — синхронный,
- JDBC — синхронный,
- Сообщение операционной системе — возможны оба режима в зависимости от сценария использования. Например, если запрос запускает скрипт, результат которого неинтересен или допускает проверку в конце сценария, можно использовать асинхронный режим.

4.1.3 Настройка правил выбора входящего сообщения

На странице канала в форме **Правила выбора входящего сообщения** (см. рисунок 18) можно определить список проверок для параметров сообщений, поступающих в приложение на входящих тестовых шагах. Сообщения, подходящие под все правила канала, передаются в соответствующий тестовый шаг (см. 3.2.4 «Выполнение входящего тестового шага»).

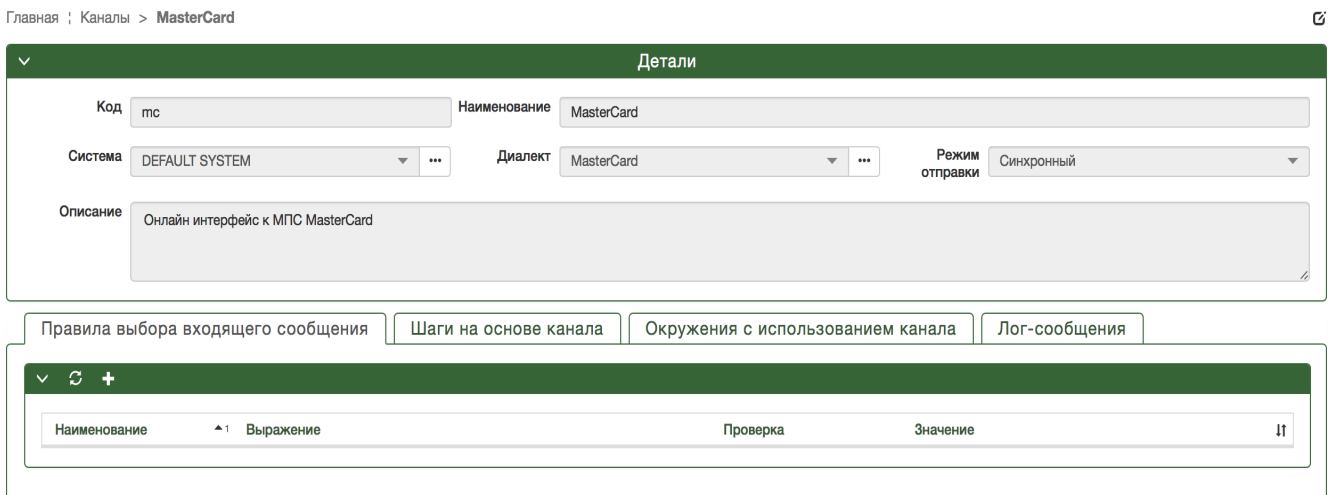


Рисунок 18 Правила выбора входящего сообщения

Чтобы создать правило выбора входящего сообщения, укажите данные, как показано на рисунке 19.

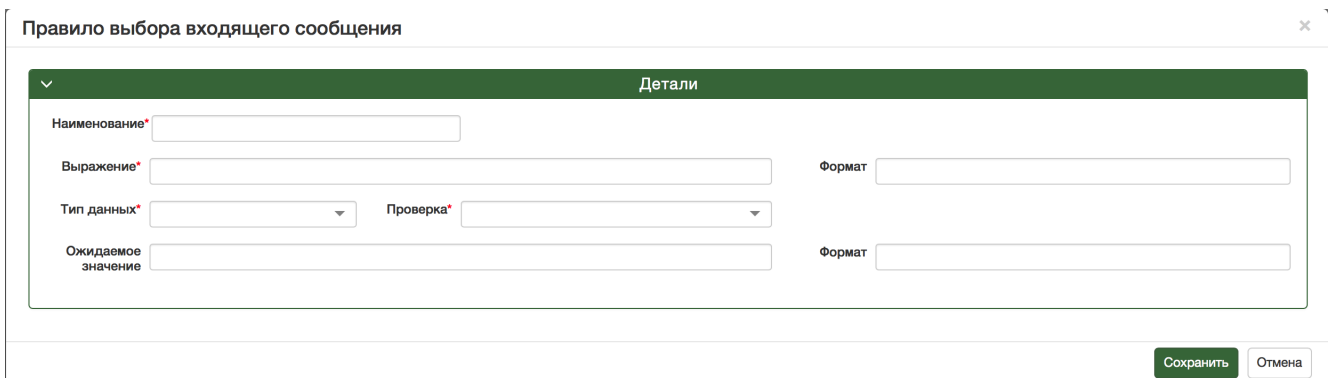


Рисунок 19 Создание правила выбора входящего сообщения

- **Наименование** — название правила, отображаемое в формах,
- **Выражение** — ссылка на поле входящего сообщения или выражение (в зависимости от источника данных). В качестве выражения могут быть использованы контекстные переменные (см. 5.4.6 «Контекстные переменные») и системные функции (см. 5.4.8 «Контекстные функции»). Формат данного поля зависит от диалекта, используемого для канала.
 Подробнее о диалектах см. в разделе 5.4.9 «Типы шагов».
- **Формат** (выражения) — формат данных, который следует применить к выражению (подробнее о синтаксисе данного поля см. в разделе 5.7 «Форматирование»),
- **Тип данных** — тип данных выражения. Поддержаны следующие варианты:
 - Строка,
 - Целое число,
 - Вещественное число,
 - Логическое да/нет,
 - Дата,
 - Время,
 - Дата и время,
 - Таблица.
- **Проверка** — тип сравнения. Поддержаны следующие варианты:
 - Равно,
 - Не равно,

- Меньше,
 - Меньше или равно,
 - Больше,
 - Больше или равно,
 - Входит в список (значения указываются через запятую),
 - Содержит подстроку,
 - Начинается со строки,
 - Заканчивается строкой,
 - Регулярное выражение (подробнее о формате регулярного выражения см. спецификацию <https://docs.oracle.com/javase/8/docs/api/java/util/regex/Pattern.html>),
 - Сравнение таблиц (см. 5.4.7.1 «Проверка переменной типа «Таблица»»),
 - Отсутствует (NULL),
 - Присутствует (NOT NULL).
- **Ожидаемое значение** — значение, которому должно соответствовать выражение. В качестве значения может быть указана контекстная переменная, которая присутствует в контексте входящего шага на момент получения входящего сообщения (см. 5.4.6 «Контекстные переменные»),
 - **Формат** (ожидаемого значения) — формат данных, который следует применить к ожидаемому значению (подробнее о синтаксисе данного поля см. в разделе 5.7 «Форматирование»).

4.1.4 Настройка шаблона Url для канала на основе диалекта HTTP

Одни и те же тестовые сценарии часто используются для тестирования нескольких окружений. В этом случае адрес URL для каждого окружения свой. Чтобы задать Url, который зависит от окружения, рекомендуется настроить для каждого окружения шаблон Url.

Чтобы настроить шаблон Url, откройте страницу канала и выполните следующие действия:

1. Перейдите на вкладку **Конфигурация для окружений**.
2. Выберите окружение и перейдите по ссылке в колонке **Наименование**.
3. Введите в поле **Шаблон Url** шаблон адреса Url, например, `http://abc.com/${url}`, где переменная `${url}` — адрес Url, заданный в свойствах шага (см. раздел 5.4.9.4 «Работа с диалектом HTTP»).
4. Сохраните изменения.

4.2 Функциональные области

Список функциональных областей, тестируемых с помощью приложения, доступен в меню **Конфигурация** → **Функциональные области**. Базовый список может быть расширен пользователем.

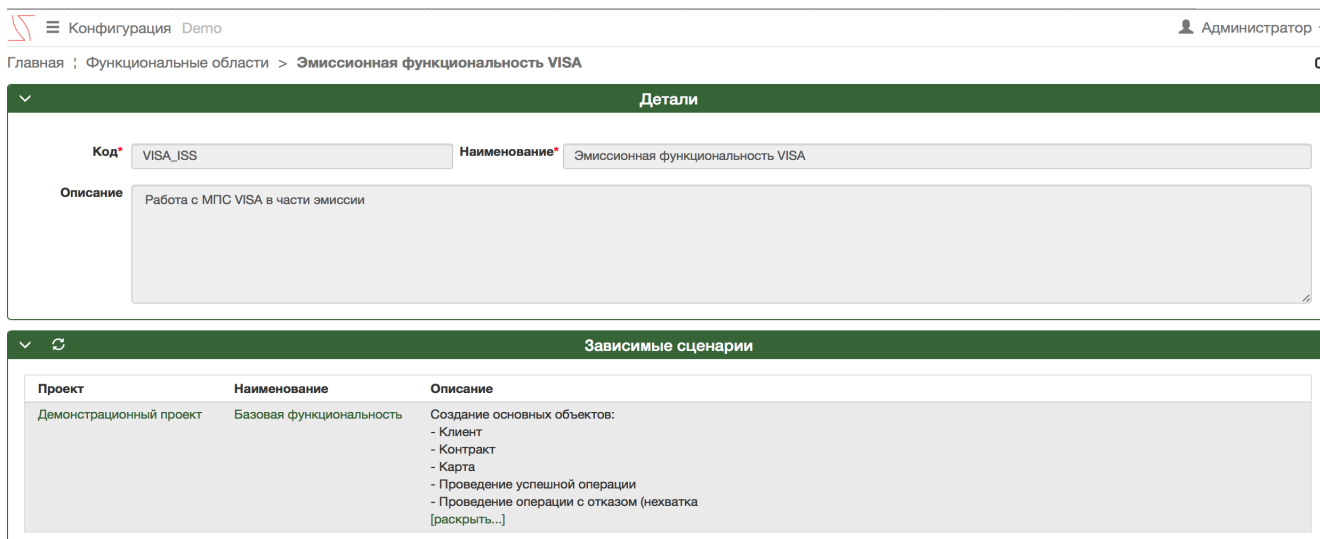


Рисунок 20 Функциональные области

Таблица **Функциональные области** содержит поля, описанные в таблице 4-2.

Таблица 4-2 Состав данных таблицы «Функциональные области»

Поле	Описание
Код	Код области, который может быть использован внешней системой при управлении SOLAR Testing Toolkit через веб-сервис. Код должен быть уникальным в рамках областей приложения
Наименование	Название функциональной области, отображаемое в формах
Описание	Описание назначения области

Чтобы увидеть список сценариев, относящихся к функциональной области, перейдите по ссылке в колонке **Наименование**. Список сценариев отображается в форме **Зависимые сценарии**, как показано на рисунке 21.

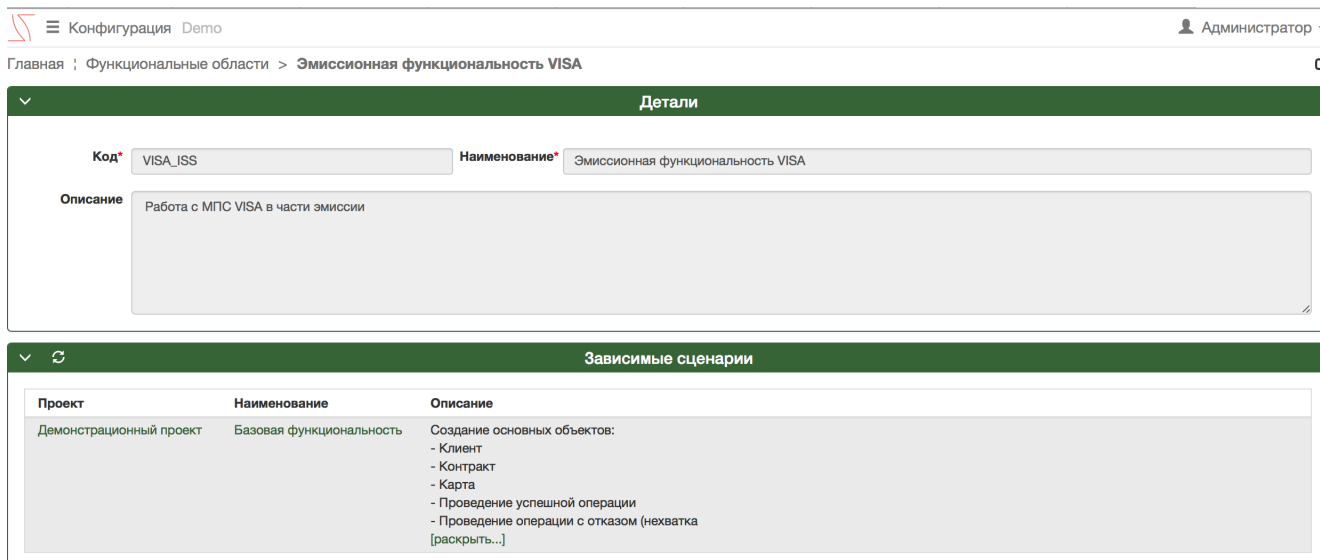


Рисунок 21 Зависимые сценарии функциональной области

4.3 Системы

Системы соответствуют тестируемым приложениям и используются в приложении как один из способов группирования сценариев. Система состоит из одного или нескольких модулей. Модуль системы можно ассоциировать со сценариями.

Системы, заданные в приложении, доступны через меню **Конфигурация** → **Системы**.

Состав данных таблицы **Системы** описан в таблице 4-3.

Таблица 4-3 Состав данных таблицы «Системы»

Поле	Описание
Код	Код системы, который может быть использован внешней системой при управлении SOLAR Testing Toolkit через веб-сервис. Код должен быть уникальным в рамках приложения
Наименование	Название системы, отображаемое в формах
Описание	Описание назначения системы

4.3.1 Создание системы

Чтобы создать систему:


1. Добавьте запись в форму **Системы** и укажите данные, описанные в таблице 4-3.
2. Выберите запись по ссылке в колонке **Наименование**.
3. В форме **Модули** добавьте один или несколько модулей и укажите следующие данные:
 - **Код** — уникальный код,
 - **Наименование** — наименование модуля,
 - **Описание** — описание модуля.
4. В форме **Версии** добавьте версию и укажите следующие данные:
 - **Код** — уникальный код,
 - **Наименование** — наименование версии,
 - **Описание** — описание версии.
5. Нажмите **Сохранить**.

При создании системы флажок **Архивная** недоступен для установки. Флажок используется как атрибут, показывающий текущий статус версии системы, и устанавливается при помощи процедуры, описанной в разделе 4.3.2 «Архивация версии системы».

4.3.2 Архивация версии системы

Если в конфигурации системы присутствует версия, которая больше не используется, рекомендуется установить для нее признак «Архивная». Архивные версии систем не отображаются в выпадающем списке **Версия** при создании и редактировании окружения.

Чтобы архивировать версию системы:

1. Перейдите на страницу системы или версии системы.
2. Нажмите кнопку  (Архивировать).

4.4 Окружения

Окружение в терминах приложения — это набор используемых систем.



Состав данных таблицы **Окружения** описан в таблице 4-4.

Таблица 4-4 Состав данных таблицы «Окружения»

Поле	Описание
Код	Код окружения, который может быть использован внешней системой при управлении SOLAR Testing Toolkit через веб-сервис. Код должен быть уникальным в рамках приложения.
Наименование	Название окружения, отображаемое в формах
Описание	Описание назначения окружения

4.4.1 Создание окружения


Чтобы создать окружение:

1. Добавьте запись в форму **Окружения** и укажите данные, описанные в таблице 4-4.
2. Выберите запись по ссылке в колонке **Наименование**.
3. Нажмите кнопку  (Редактировать) в правом верхнем углу страницы.
4. В форме **Системы** выберите используемые в окружении системы:
 - в колонке **Используется** установите флажок,
 - в колонке **Версия** выберите используемую версию системы.
5. Нажмите кнопку  (Сохранить).
6. В форме **Каналы** установите флажок **Выполнять** для тех каналов, которые доступны на данном окружении.

4.4.2 Экспорт окружения

Экспорт окружения позволяет сохранить окружение в виде файла формата JSON, который в дальнейшем может быть импортирован в приложение, установленное в другой системе. При наличии в конфигурации приложения коннектора для работы с системой контроля версий (VCS) возможен экспорт окружения в репозиторий. В текущей версии поддерживается работа с системой Git.

Чтобы экспортировать окружение:


1. Выберите пункт меню **Конфигурация** → **Окружения**.
2. Откройте страницу окружения, перейдя по ссылке в колонке **Наименование**.
3. Нажмите кнопку  (Экспортировать окружение) в правом верхнем углу.
4. В окне **Экспорт окружения** установите флажок **Включить зависимости**, чтобы добавить информацию о зависимостях окружения (системах, модулях, каналах и конфигураций каналов, заданных для окружения).
5. В списке **VCS** выберите коннектор для работы с VCS, если необходимо выгрузить окружение в VCS.
6. Выгрузите окружение:
 - Нажмите кнопку **Скачать окружение**, чтобы выгрузить шаблон в файловую систему или
 - Нажмите **Загрузить в VCS**, чтобы выгрузить в VCS.

Файл с именем environment_ID.json по умолчанию создается в директории, выбранной для сохранения загружаемых файлов, или выгружается в репозиторий.

4.4.3 Импорт окружения

Импорт окружения позволяет добавить окружение в приложение, используя исходный файл формата JSON, расположенный в файловой системе или репозитории VCS.

Чтобы импортировать окружение:

1. Выберите пункт меню **Конфигурация** → **Окружения**.
2. Нажмите кнопку  (Импортировать окружение) в правом верхнем углу.
3. В окне **Импорт окружения** выберите файл:
 - нажмите кнопку **Выбрать файл** и выберите JSON файл, сохраненный в файловой системе, или
 - в списке **VCS** выберите название коннектора для работы с репозиторием, а в поле **Коммит** — идентификатор коммита.
4. Установите флажок **Заменять существующие**, если в группе уже есть окружения с таким кодом и копии шаблонов не требуются.

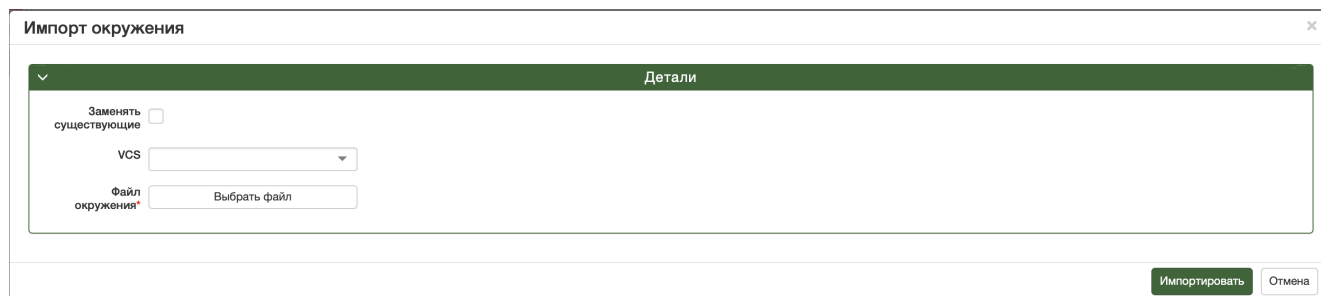


Рисунок 22 Импорт окружения

Чтобы избежать конфликт имен, когда в приложении уже есть окружение с кодом и наименованием импортируемого окружения, к коду и наименованию нового окружения добавляется суффикс.

4.5 Шаблоны действия

Шаблон действия или шаблон — это сохраненный набор тестовых шагов, который может быть вставлен в любой тестовый сценарий. Шаблон не имеет доступа к контекстным переменным сценария, из которого он был вызван, а работает со своим контекстом. Контекст шаблона иницируется переменными, переданными через параметры шаблона на вызывающем шаге (см. раздел 5.4.10 «Использование шаблона действия»). Переменные из контекста шаблона можно передать обратно в сценарий, из которого шаблон был вызван. Для этого надо использовать исходящие параметры, помеченные как возвращаемые (см. 4.5.2 «Создание шаблона»).

Шаблоны можно ассоциировать с группами, чтобы разделять шаблоны, предназначенные для взаимодействия с разными системами.

4.5.1 Создание группы шаблонов

Для создания новой группы шаблонов перейдите в меню **Конфигурация** → **Шаблоны** и добавьте запись в таблицу **Группы шаблонов**. Состав данных описан в таблице 4-5.

Таблица 4-5 Состав данных таблицы «Группы шаблонов»

Поле	Описание
Код	Код группы, который может быть использован внешней системой при управлении SOLAR Testing Toolkit через веб-сервис. Код должен быть уникальным в рамках групп шаблонов
Наименование	Название группы, отображаемое в формах

Поле	Описание
Описание	Описание назначения группы

4.5.2 Создание шаблона

Чтоб создать новый шаблон действия, добавьте запись в таблицу **Все шаблоны**.

Чтобы создать шаблон и одновременно ассоциировать его с группой шаблонов, перейдите на страницу группы шаблонов и добавьте запись в таблицу **Шаблоны действия**.

Таблица **Все шаблоны** содержит поля, описанные в таблице 4-6.

Таблица 4-6 Состав данных таблицы «Шаблоны действия»

Поле	Описание
Группа	Наименование группы шаблонов, к которой относится шаблон. Для шаблонов, которые не ассоциированы с группами, пустое поле
Код	Код шаблона, который может быть использован внешней системой при управлении SOLAR Testing Toolkit через веб-сервис. Код должен быть уникальным в рамках шаблонов приложения
Наименование	Название шаблона, отображаемое в формах
Описание	Описание назначения шаблона

На странице шаблонов действия доступен поиск шаблона по группе шаблонов, коду и полному наименованию (см. рисунок 23).

Рисунок 23 Поиск шаблона действия

При создании шаблона действия (см. рисунок 24) укажите следующие данные:

- **Код**,
- **Наименование**,
- **Описание** (опционально),
- **Входящие параметры** — указанные параметры шаблона действуют для всех шагов, созданных на основе шаблона, и не могут быть изменены на странице шага (кроме поля **Выражение**, см. рисунок 43).
 - **Переменная** — имя переменной. Указывается без специальных символов, например, cardExp.
 - **Значение по умолчанию** (опционально) — значение переменной по умолчанию. Используется на всех шагах на основе данного шаблона, для которых не определено другое значение. Значение для конкретного шага можно изменить на странице шага.
 - **Тип данных** — тип данных переменной. Выбирается из доступных вариантов: Строка, Целое число, Вещественное число, Логическое да/нет, Дата, Время, Дата и время, Таблица.
 - **Описание** (опционально) — описание переменной.

- **Исходящие параметры** — указанные параметры шаблона можно пометить как возвращаемые и сохранить в контексте. Тогда они будут переданы в сценарий, из которого был вызван шаблон.

Рисунок 24 Создание шаблона действия

Для добавления тестовых шагов в шаблон перейдите на его страницу (см. рисунок 25) по ссылке, сохраненной в названии шаблона.

Рисунок 25 Тестовые шаги шаблона действия

Процесс создания тестового шага в шаблоне аналогичен созданию шага в тестовом сценарии (см. раздел 5.4 «Тестовые шаги»).

В форме **Шаги на основе шаблона** на странице шаблона отображается список всех шагов, которые вызывают данный шаблон (см. рисунок 26). В колонках **Проект**, **Тестовый сценарий**, **Родительский шаблон** и **Наименование** отображаются ссылки, по которым можно перейти на страницы соответствующего проекта, сценария, родительского шаблона или шага.

Шаги на основе шаблона				
Проект	Тестовый сценарий	Родительский шаблон	Наименование	Описание
Демонстрационный проект	Базовая функциональность		40 Успешная покупка	Вызов шаблона покупка для карты из предыдущего шага. Проверяется код ответа.
Демонстрационный проект	Базовая функциональность		50 Неуспешная покупка (нехватка средств)	Вызов шаблона покупка для карты из предыдущего шага. Проверяется код ответа.
Демонстрационный проект -1	Базовая функциональность		40 Успешная покупка	Вызов шаблона покупка для карты из предыдущего шага. Проверяется код ответа.
Демонстрационный проект -1	Тестовый сценарий		40 Успешная покупка	Вызов шаблона покупка для карты из предыдущего шага. Проверяется код ответа.
Демонстрационный проект -1	Тестовый сценарий		60 Неуспешная покупка (нехватка средств)	Вызов шаблона покупка для карты из предыдущего шага. Проверяется код ответа.

Рисунок 26 Шаги на основе шаблона

4.5.3 Добавление шаблона к группе


Чтобы добавить шаблон к группе,

1. Найдите шаблон в таблице **Все шаблоны** и нажмите кнопку **Переместить шаблон** в правой части таблицы.
2. В окне **Перемещение** выберите группу шаблонов и нажмите **Переместить**.

4.5.4 Копирование шаблона

Копирование шаблона позволяет быстро создать новый шаблон на основе существующего: создайте копию шаблона, а затем модифицируйте ее.

Чтобы создать копию шаблона:

1. Нажмите кнопку  (Копировать) в таблице **Шаблоны действия** или на странице выбранного шаблона.
2. Во всплывающем окне **Настройки копирования** выберите число копий в поле **Количество** и нажмите **Сохранить**.

Настройки копирования

Детали

Код: Наименование:

Описание:

Retail операция из МПС MasterCard

Количество:

Рисунок 27 Настройки копирования шаблона


Новый шаблон содержит все шаги первоначального шаблона. Во избежание конфликта имен к коду и наименованию нового шаблона добавляется суффикс, например, MC_FLOW (1). Новый шаблон доступен для редактирования.

4.5.5 Экспорт шаблона

Экспорт шаблона позволяет сохранить шаблон в виде файла формата JSON, который в дальнейшем может быть импортирован в приложение, установленное в другой системе.

При наличии в конфигурации приложения коннектора для работы с системой контроля версий (VCS) возможен экспорт шаблона в репозиторий. В текущей версии поддерживается работа с системой Git.

Чтобы экспортировать шаблон:

1. Выберите пункт меню **Конфигурация → Шаблоны**.
2. Найдите шаблон в таблице **Все шаблоны** или в таблице **Шаблоны действия** на странице группы, которой он принадлежит.
3. Откройте страницу шаблона, перейдя по ссылке в колонке **Наименование**.
4. Нажмите кнопку  (Экспортировать шаблон) в правом верхнем углу.
5. В окне **Экспорт шаблона** установите флажок **Включить зависимости**, чтобы добавить информацию о зависимостях шаблона (наборах параметров, каналах, диалектах).
6. В списке **VCS** выберите коннектор для работы с VCS, если необходимо выгрузить шаблон в VCS.
7. Скачайте шаблон:
 - Нажмите кнопку **Скачать шаблон**, чтобы выгрузить шаблон в файловую систему или
 - Нажмите **Загрузить в VCS**, чтобы выгрузить в VCS.
8. Нажмите кнопку **Экспортировать шаблон**.


Файл с именем `template_ID.json` по умолчанию создается в директории, выбранной для сохранения загружаемых файлов, или выгружается в репозиторий.

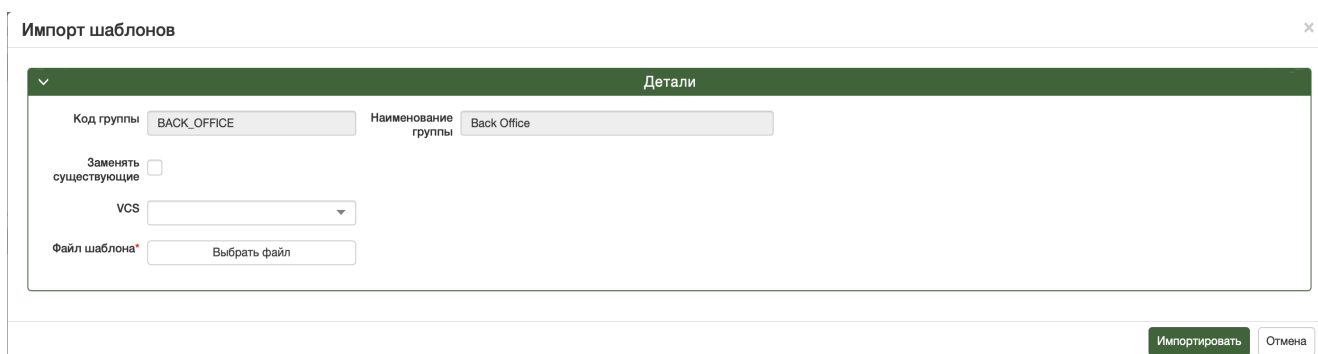
Для экспорта всех шаблонов группы используйте кнопку **Экспортировать все шаблоны**, расположенную в правом верхнем углу страницы группы шаблонов.

4.5.6 Импорт шаблона

Импорт шаблона позволяет добавить шаблон в приложение, используя исходный файл формата JSON, расположенный в файловой системе или репозитории VCS. Поскольку шаблоны объединены в группы, при импорте всегда требуется выбрать группу шаблона.

Чтобы импортировать шаблон:

1. Выберите пункт меню **Конфигурация → Шаблоны**.
2. Перейдите на страницу группы шаблонов, в которую следует добавить шаблон.
3. Нажмите кнопку  (Импортировать шаблоны) в правом верхнем углу.
4. В окне **Импорт шаблонов** выберите файл:
 - нажмите кнопку **Выбрать файл** и выберите JSON файл, сохраненный в файловой системе, или
 - в списке **VCS** выберите название коннектора для работы с репозиторием, а в поле **Коммит** — идентификатор коммита.
5. Установите флажок **Заменять существующие**, если в группе уже есть шаблоны с таким кодом и копии шаблонов не требуются.



Импорт шаблонов

Детали

Код группы: BACK_OFFICE Наименование группы: Back Office

Заменять существующие:

VCS: [dropdown]

Файл шаблона: [Выбрать файл]

Импортировать Отмена

Рисунок 28 Импорт шаблонов

Чтобы избежать конфликт имен, когда в группе уже есть шаблон с кодом и наименованием импортируемого шаблона, к коду и наименованию нового шаблона добавляется суффикс по аналогии с операцией копирования.

4.6 Наборы параметров

Наборы параметров представляют собой подготовленные множества значений, которые можно использовать в тестовых сценариях.

Каждый набор содержит определение переменных и одну или несколько строк с их значениями. Использование наборов параметров избавляет от необходимости явного ввода данных на шагах сценария, что экономит время и снижает вероятность ошибок.

Чтобы обеспечить доступ к переменным из набора параметров, его надо привязать к проекту. Набор параметров, привязанный к проекту, можно использовать во всех сценариях проекта.

Например, требуется эмулировать операцию с использованием банковской карты. Рекомендуется создать набор параметров карты и привязать его к проекту. Каждый раз, когда в тестовом сценарии требуется ввести данные карты, будет достаточно вызвать переменную из набора.

Режим блокировки набора параметров позволяет предотвратить одновременное использование параметров разными сценариями. Эта возможность полезна в тех случаях, когда внешняя система не может одновременно обрабатывать несколько запросов по одной карте.

Наборы параметров можно добавить в приложение любым из способов:

1. Создать вручную с помощью интерфейса приложения (см. раздел 4.6.1)
2. Загрузить из базы данных (см. раздел 4.6.2)
3. Загрузить из XML-файла (см. Приложение В «Интерфейс saveParameterSetRowBatch»)

Обратите внимание, что с помощью загрузки набора параметров из базы данных (2) и экспорта/импорта набора параметров из XML-файла (3) возможно не только создание нового набора параметров, но и обновление существующего.

Подробнее о привязке параметров к проекту см. в разделе 5.2 «Привязка набора параметров к проекту»

Список доступных наборов параметров отображается в форме **Наборы параметров (Общие настройки → Наборы параметров)**.

Таблица 4-7 Состав данных таблицы «Наборы параметров»

Поле	Описание
Код	Код набора параметров, который может быть использован внешней системой при управлении SOLAR Testing Toolkit через веб-сервис. Код должен быть уникальным в рамках наборов параметров
Наименование	Название набора параметров, отображаемое в формах

4.6.1 Создание набора параметров

Чтобы создать набор параметров вручную с помощью интерфейса приложения:

1. Выберите пункт меню **Общие настройки** → **Наборы параметров**.
2. Добавьте новую запись и укажите следующие данные:
 - **Код**,
 - **Наименование**,
 - **Блокировка** — выберите **Присутствует**, если при использовании набора параметров требуется их блокировать;
 - **Описание** (опционально),
 - **Параметры**
 - **Переменная** — имя переменной,
 - **Значение по умолчанию** (опционально) — значение переменной по умолчанию. Используется для тех строк набора параметров, для которых не определено другое значение.
 - **Тип данных** — тип данных переменной. Выбирается из доступных вариантов: Строка, Целое число, Вещественное число, Логическое да/нет, Дата, Время, Дата и время, Таблица.
 - **Описание** (опционально) — описание переменной.
3. Нажмите **Сохранить**.
4. Выберите созданную запись в таблице **Наборы параметров**.
5. Добавьте одну или несколько строк в форме **Строки** и укажите следующие данные.
 - **Код**,
 - **Наименование**,
 - **Описание** (опционально),
 - **Параметры**
 - **Выражение** — значение переменной.
Выражение может содержать вызов системной функции (см. раздел 5.4.8 «Контекстные функции»).



4.6.2 Загрузка набора параметров из базы данных

В данном разделе описан способ создания набора параметров, когда значения параметров получаются с помощью выборки из базы данных. Указанная возможность может быть полезна в том случае, когда значения параметров могут изменяться, а для тестирования важно использовать актуальные значения. Например, при наличии модуля CRM можно создать набор параметров с данными клиента и получать актуальные значения из базы данных CRM непосредственно перед тестированием.

Возможность загрузки набора параметров из базы данных доступна, если в приложении настроен канал для подключения к БД, основанный на диалекте JDBC.

Чтобы загрузить набор параметров из базы данных:

1. Выберите пункт меню **Общие настройки** → **Наборы параметров** (см. рисунок 29).
2. Добавьте новую запись и укажите данные, перечисленные в пункте 2 раздела 4.6.1, а также:
 - **Канал** — выберите канал для подключения к базе данных,
 - **Окружение** — выберите окружение, на котором доступен выбранный канал,
 - **Режим обновления** (опционально) — выберите **Перед тестированием**, чтобы обеспечить обновление параметров перед каждым их использованием в запуске проекта.
 - **Параметр кода строки** (опционально) — наименование колонки, значения которой будут использованы в качестве кода строки набора параметров. Если не указано, будет использоваться колонка «parameter_set_row_code».

- **Параметр наименования строки** (опционально) — наименование колонки, значения которой будут использованы в качестве наименования строки набора параметров. Если не указано, будет использоваться параметр кода строки.
 - **Запрос** — введите выражение SELECT для выборки значений из базы данных. Обратите внимание, что названия колонок, в которых хранятся значения параметров, должны совпадать с названиями параметров.
3. Нажмите **Сохранить**.
 4. Чтобы запустить загрузку набора параметров, перейдите по ссылке в колонке **Наименование** и нажмите  в правом верхнем углу формы.
 5. Чтобы запустить загрузку всех наборов параметров, для которых заданы канал и окружение, нажмите  в правом верхнем углу формы **Наборы параметров**.

Главная | Наборы параметров > Набор параметров

Детали

Код: PARAMETER_SET_ROW Наименование: Набор параметров Блокировка: ▼

Описание:

Переменная	Значение по умолчанию	Тип данных	Описание
param1		Целое число	
param2		Строка	

Канал: DataBase ... Окружение: DEFAULT ENVIRONMENT ...

Параметр кода строки: parameter_set_row_code Параметр наименования строки: parameter_set_row_name

Запрос:

```

1 SELECT 'rowCode1' as "parameter_set_row_code", 'rowName1' as "parameter_set_row_name", 1 as "param1", 'A' as "param2"
2 UNION
3 SELECT 'rowCode2' as "parameter_set_row_code", 'rowName2' as "parameter_set_row_name", 2 as "param1", 'B' as "param2"
4 UNION
5 SELECT 'rowCode3' as "parameter_set_row_code", 'rowName3' as "parameter_set_row_name", 3 as "param1", 'C' as "param2"


```

Рисунок 29 Загрузка набора параметров из базы данных

4.6.3 Экспорт набора параметров

Экспорт набора параметров позволяет сохранить набор в виде файла формата JSON, который в дальнейшем может быть импортирован в приложение, установленное в другой системе.

Чтобы экспортировать набор параметров:

1. Выберите пункт меню **Конфигурация** → **Наборы параметров**.
2. Откройте страницу набора, перейдя по ссылке в колонке **Наименование**.
3. Нажмите кнопку  (Экспортировать набор параметров) в правом верхнем углу. Откроется всплывающее окно **Экспорт набора параметров**.

Детали

Код: atmCards Наименование: ATM Cards

Скачать файл Скачать файл

Отмена


4. Нажмите кнопку **Скачать файл**, чтобы выгрузить набор параметров в файловую систему.

Файл с именем parameterSet_ID.json по умолчанию создается в директории, выбранной для сохранения загружаемых файлов.

4.6.4 Импорт набора параметров

Импорт набора параметров позволяет добавить набор параметров в приложение, используя исходный файл формата JSON, расположенный в файловой системе.

Чтобы импортировать шаблон:

1. Выберите пункт меню **Конфигурация** → **Наборы параметров**.
2. Нажмите кнопку  (Импортировать набор параметров) в правом верхнем углу.
3. В окне **Импорт набора параметров** нажмите кнопку **Выбрать файл** и выберите JSON файл, сохраненный в файловой системе:

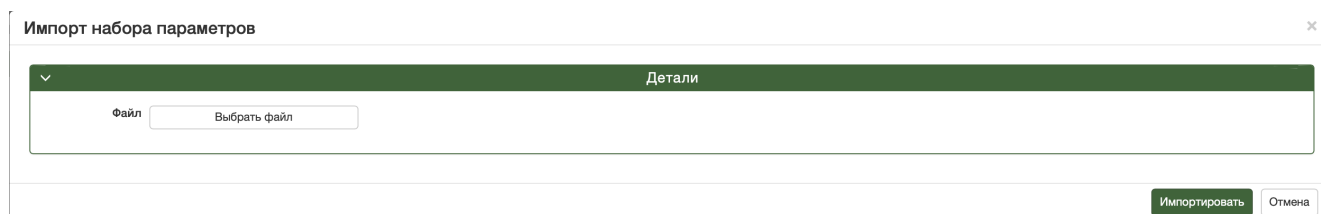


Рисунок 30 Импорт набора параметров

4.7 Пользовательские настройки (окружение)

В текущей версии приложения пользовательские настройки состоят из настройки окружения по умолчанию. Если оно задано, то при запуске сценария, проекта, плана выполнения поле **Окружение** в диалоге запуска автоматически заполняется заданным значением.

Чтобы настроить окружение по умолчанию:

1. Выберите пункт меню **Конфигурация** → **Пользовательские настройки**.
2. В списке **Окружение по умолчанию** выберите значение.

Обратите внимание: при обновлении с версии 2.3.1 или предшествующей ей к версии 2.3.2 и более поздним пользовательские настройки, сохраненные ранее, не могут быть восстановлены. В этом случае рекомендуется настроить их заново.

4.8 Управление пользователями и правами доступа

Учетные записи пользователей создаются и управляются администратором информационной системы при помощи пунктов меню секции **Пользователи**.

Если в приложении настроена аутентификация типа SOLAR DB, при первом входе пользователя в приложение рекомендуется сменить пароль по умолчанию, назначенный администратором (см. раздел 4.8.1).

Подробнее о возможностях настройки учетных записей и прав доступа пользователей см. в документе «SOLAR User Management».

4.8.1 Изменение пароля и данных учетной записи пользователем

Пользователь может изменить пароль и другие данные своей учетной записи в форме **Профиль**. Рекомендуется изменить первичный пароль, назначенный администратором, при первом входе в приложение.

Чтобы изменить пароль своей учетной записи:

1. Нажмите значок учетной записи в правом верхнем углу страницы.
2. В появившемся списке нажмите ссылку **Профиль**. Откроется форма **Профиль** с данными вашей учетной записи.

3. Нажмите кнопку **☰ (Смена пароля)** в правом верхнем углу формы и укажите следующие параметры:
 - **Старый пароль,**
 - **Новый пароль,**
 - **Подтверждение пароля.**
4. Нажмите кнопку **Сохранить**.

Пароль должен соответствовать политике безопасности паролей, заданной администратором. Например, содержать хотя бы одну цифру и отличаться от двух предыдущих паролей. При необходимости обратитесь к администратору и уточните состав требований к паролю учетной записи.

Чтобы изменить данные своей учетной записи:

1. Откройте форму **Профиль**.
2. Нажмите кнопку **✎ (Редактировать)**.
3. Измените нужные данные, описанные в таблице 4-8.
4. Нажмите кнопку **↓ (Сохранить)**.

Таблица 4-8 Данные текущей учетной записи пользователя

Параметр	Значение	Описание
Имя	До 256 символов	Имя пользователя
Имя для входа (login)		Уникальный идентификатор учетной записи пользователя в приложении
E-mail		Адрес электронной почты пользователя
Описание	До 4000 символов	Дополнительные сведения об учетной записи пользователя. Справочная информация

4.8.2 Стандартные группы пользователей

Приложение поддерживает набор стандартных групп с заранее заданной конфигурацией прав доступа, которые рекомендуется использовать при создании новых пользователей. Стандартные группы включают:

1. Гости.
2. Пользователи тестов.
3. Разработчики тестов.
4. Менеджеры тестов.
5. Администраторы.

Права доступа каждой группы описаны в таблице 4-9.

Таблица 4-9 Права доступа стандартных групп

Группа	Права доступа
Гости	Просмотр сценариев

Группа	Права доступа
Пользователи тестов	Полномочия группы «Гости», а также: <ul style="list-style-type: none">• Запуск проектов и сценариев,• Создание, редактирование и удаление проверок на тестовых шагах,• Создание, редактирование и удаление установок переменных на тестовых шагах,• Создание, редактирование и удаление пользовательских настроек
Разработчики тестов	Полномочия группы «Пользователи тестов», а также: <ul style="list-style-type: none">• Создание, редактирование и удаление тестовых шагов, сценариев и проектов,• Создание, редактирование и удаление наборов параметров и строк наборов параметров,• Создание, редактирование и удаление версий, функциональных областей и требований,• Создание, редактирование и удаление планов выполнения и условий планов выполнения
Менеджеры тестов	Полномочия группы «Разработчики тестов», а также: <ul style="list-style-type: none">• Создание, редактирование и удаление шаблонов,• Создание, редактирование и удаление каналов, систем, модулей, диалектов и окружений,• Экспорт конфигурации,• Создание, редактирование и удаление конфигурации отчетов и запуск формирования отчетов,• Создание, редактирование и удаление конфигурации уведомлений
Администраторы	Управление пользователями

Дополнительно можно настроить доступ к проектам и планам выполнения в соответствии с группами.

5 Работа с проектами

Список проектов доступен на главной странице приложения, на которую можно перейти по ссылке **Главная** (см. рисунок 31). Под списком проектов расположена форма **Очередь выполнения**.

Для проекта можно установить режим запуска сценариев с блокировкой.

- Полная блокировка — блокируется окружение целиком.
- Выборочная — блокируются отдельные системы, которые используются каналами, тестируемыми на шагах сценария.

Режим блокировки можно также выбрать в свойствах сценария. При запуске сценария используется режим, заданный в свойствах сценария, а не проекта.

Подробнее о блокировке см. в разделе 5.3 «Тестовые сценарии»

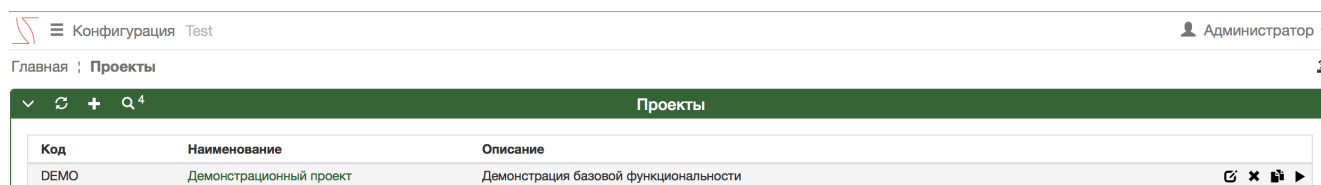


Рисунок 31 Проекты

На странице проектов доступен поиск проекта по коду и полному наименованию (см. рисунок 32).

Подробнее о возможностях поиска см. в разделе 2.5 «Табличные формы».

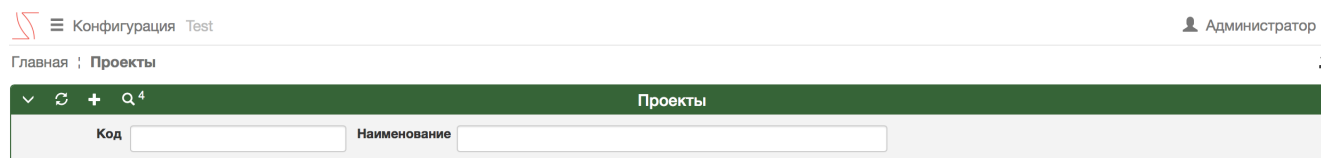


Рисунок 32 Поиск проекта

Для работы с конкретным проектом перейдите по ссылке в колонке **Наименование**.

5.1 Создание проекта

Чтобы создать новый проект, добавьте запись в форму **Проекты** и укажите данные, описанные в таблице 5-1.

Параметры проекта описаны в таблице 5-1.

Таблица 5-1 Параметры проекта

Поле	Описание
Код	Код проекта, который может быть использован внешней системой при управлении SOLAR Testing Toolkit через веб-сервис. Код должен быть уникальным в рамках проектов приложения
Наименование	Название проекта, отображаемое в формах

Поле	Описание
Блокировка сценариев	Атрибут, который указывает на необходимость выполнения сценариев проекта в режиме блокировки. Возможные значения: <ul style="list-style-type: none"> • Полная — блокируется окружение целиком, • Выборочная — блокируются системы, которые используются каналами, тестируемыми на шагах сценария, • Не задано — блокировка не задана на уровне проекта, но может быть задана отдельно для сценария
Окружение по умолчанию	Окружение, которое выбирается по умолчанию в параметрах запуска проекта
Описание	Подробное описание проекта

На странице проектов доступен поиск проекта по коду и наименованию (см. рисунок 32). *Подробнее о возможностях поиска см. в разделе 2.5 «Табличные формы».*

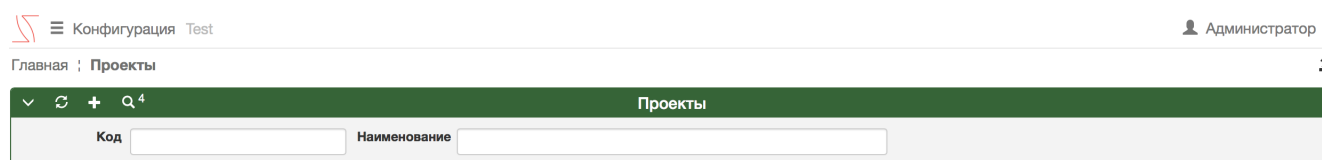



Рисунок 33 Поиск проекта

Для работы с конкретным проектом перейдите по ссылке в колонке **Наименование**.

5.1.1 Копирование проекта

Копирование проекта позволяет быстро создать новый проект на основе существующего. Чтобы создать копию проекта,

1. Нажмите кнопку  (Копировать) в таблице **Проекты** или на странице выбранного проекта.
2. Во всплывающем окне **Настройки копирования** в поле **Количество** выберите число копий и нажмите **Сохранить**.


5.1.2 Экспорт проекта

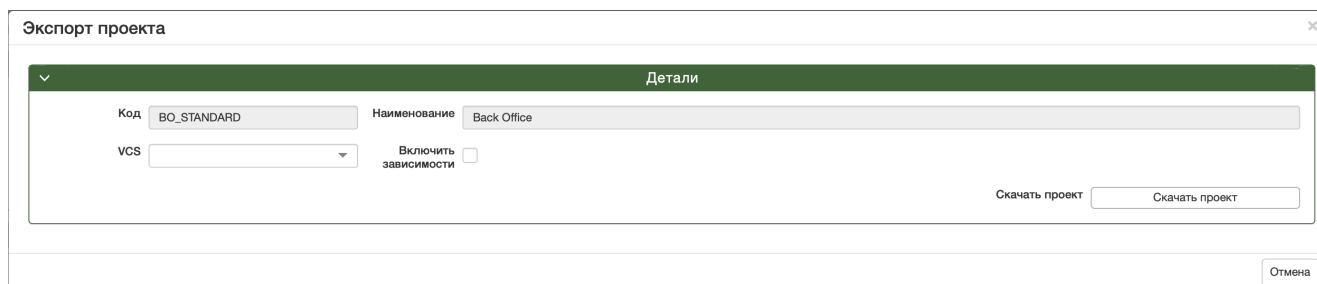
Экспорт проекта позволяет сохранить проект в виде файла формата JSON, который в дальнейшем может быть импортирован в приложение, установленное на другой системе. Формат файла изменился в версии 2.6.0 и не совместим с более ранними версиями.

При наличии в конфигурации приложения коннектора для работы с системой контроля версий (VCS) возможен экспорт проекта в репозиторий. В текущей версии поддерживается работа с системой Git.

Чтобы экспортировать проект,

1. Выберите пункт меню **Конфигурация** → **Проекты**.
2. Откройте страницу проекта, перейдя по ссылке в колонке **Наименование**.

3. Нажмите кнопку  (Экспортировать проект) в правом верхнем углу.




4. В окне **Экспорт проекта** установите флажок **Включить зависимости**, чтобы добавить информацию о зависимостях проекта (шаблонах, наборах параметров, каналах, диалектах).
5. В списке **VCS** выберите коннектор для работы с VCS, если необходимо выгрузить проект в VCS.
6. Скачайте проект:
 - Нажмите кнопку **Скачать проект**, чтобы выгрузить проект в файловую систему или
 - Нажмите **Загрузить в VCS**, чтобы выгрузить проект в VCS.

Файл с именем `project_ID.json` создается в директории, выбранной для сохранения загружаемых файлов, или выгружается в репозиторий.

5.1.3 Импорт проекта

Импорт проекта позволяет добавить проект в приложение, используя исходный файл формата JSON, расположенный в файловой системе или репозитории VCS.

Чтобы импортировать проект,

1. Выберите пункт меню **Конфигурация** → **Проекты**.
2. Нажмите кнопку  (Импортировать проект) в правом верхнем углу.
3. В окне **Импорт проекта** (см. рисунок 34) выберите файл:
 - нажмите кнопку **Выбрать файл** и выберите JSON файл, сохраненный в файловой системе, или
 - в списке **VCS** выберите название коннектора для работы с репозиторием, а в поле **Коммит** — идентификатор коммита.
4. Установите флажок **Заменять существующий**, чтобы заменить проект с таким же кодом, если такой существует, импортируемым проектом.
5. Нажмите кнопку **Импортировать**.

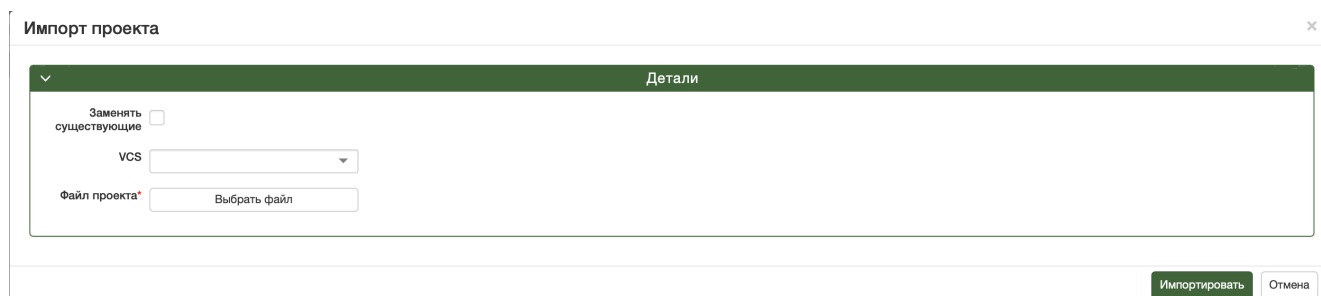


Рисунок 34 Импорт проекта

Чтобы избежать конфликт имен, когда в приложении уже есть проект с кодом и наименованием импортируемого проекта, к коду и наименованию нового проекта добавляется суффикс, например, Code (1) и Project (1).

5.2 Привязка набора параметров к проекту

Для использования в тестовых сценариях проекта заранее подготовленных наборов параметров необходимо привязать требуемые наборы параметров к проекту. При запуске проекта переменные привязанного набора загружаются в контекст и становятся доступны для вызова в следующем формате: $\{[\text{код набора}].[\text{имя переменной}]\}$. При запуске сценариев проекта приложение выбирает строку из набора в соответствии с заданным методом и берет значение переменной из выбранной строки.

Метод выбора строк определяется логикой проекта и позволяет выбрать строку из набора случайным образом, последовательно или же использовать в сценарии конкретную строку.

Подробнее о наборах параметров см. в разделе 4.6 «Наборы параметров».

Чтобы привязать набор параметров к проекту:

1. Перейдите на страницу проекта по ссылке в колонке **Наименование**.
2. Нажмите кнопку **Создать** в форме **Наборы параметров**. Появится окно **Привязанный набор параметров**, изображенное на рисунке 35.

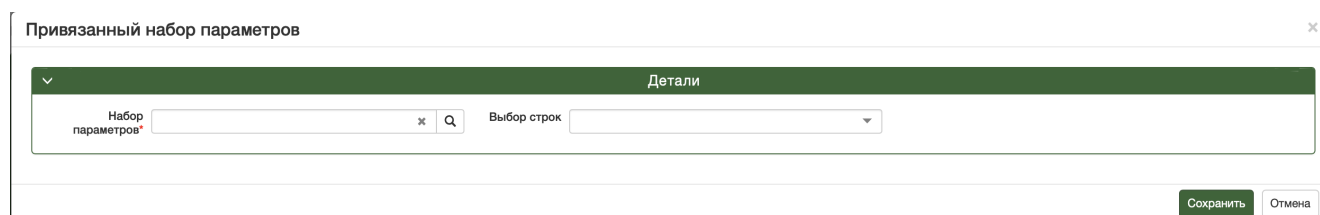


Рисунок 35 Привязка набора параметров

3. Справа от поля **Набор параметров** нажмите кнопку **Искать**, выберите набор параметров и нажмите **Выбрать**.
4. Выберите конкретную строку набора или укажите метод выбора строки при запуске проекта:
 - Справа от поля **Строка набора параметров** нажмите кнопку **Искать**, выберите строку и нажмите **Выбрать** или
 - В списке **Выбор строк** выберите:
 - **Случайный** — строка выбирается случайным образом,
 - **Последовательный** — строка выбирается последовательно,
 - **Из сценария** — используется строка, выбранная на уровне сценария;
5. Нажмите **Сохранить**.
6. Если был указан метод выбора строки **Из сценария**, в каждом сценарии проекта выберите строку набора параметров:
 - a. Перейдите на страницу сценария по ссылке в колонке **Наименование**.
 - b. Нажмите кнопку **Создать** в форме **Наборы параметров**. Появится окно **Привязанный набор параметров**.
 - c. Справа от поля **Набор параметров** нажмите кнопку **Искать**, выберите набор параметров и нажмите **Выбрать**.
 - d. Справа от поля **Строка набора параметров** нажмите кнопку **Искать**, выберите строку и нажмите **Выбрать**.

Если для привязанного набора параметров одновременно выбрать конкретную строку набора и указать метод выбора строк **Из сценария**, то указанная строка действует как строка по умолчанию для тех сценариев, для которых строка набора параметров не задана.

5.3 Тестовые сценарии

Список тестовых сценариев, разработанных для текущего проекта, доступен на странице проекта.

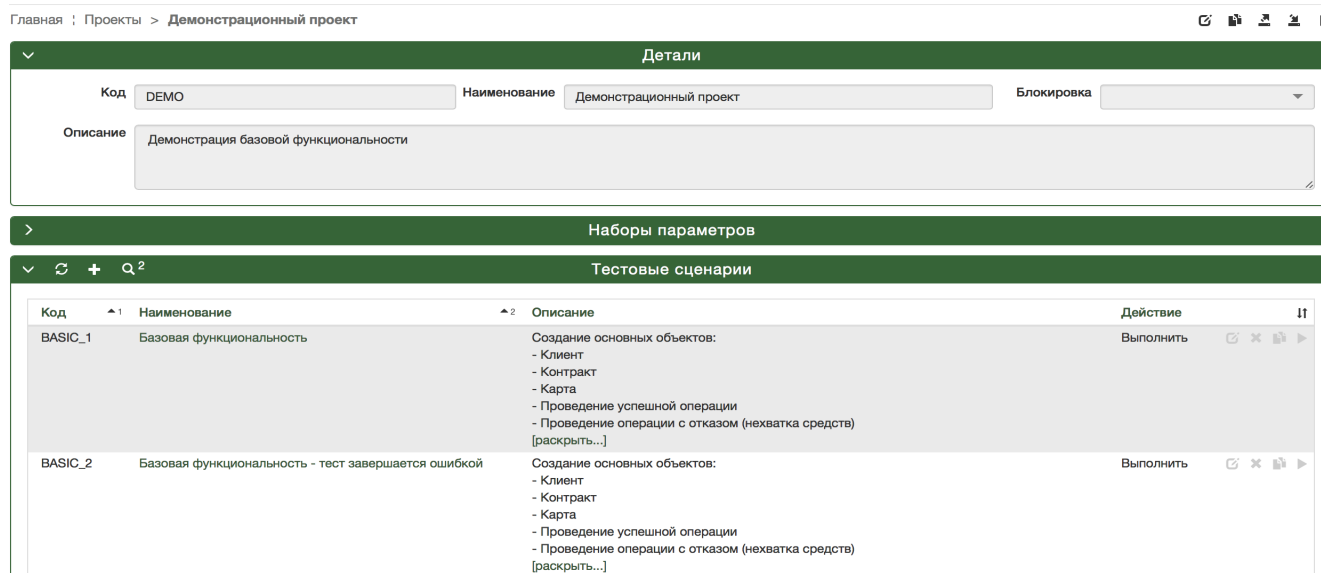


Рисунок 36 Тестовые сценарии проекта

5.3.1 Режим запуска сценариев с блокировкой

Для сценария можно установить режим запуска с блокировкой:

- Полная
 - Запуск сценария блокирует окружение целиком.
 - Остальные сценарии становятся в очередь и будут запущены после снятия блокировки, т.е. завершения текущего сценария с полной блокировкой.
- Выборочная
 - Запуск сценария блокирует отдельные системы, которые используются каналами, тестируемыми на шагах сценария.
 - Сценарии, работающие с другими системами, запускаются параллельно с текущим.
 - Сценарии, которым необходимо работать с заблокированными системами, становятся в очередь.

Выборочную блокировку рекомендуется использовать для тех сценариев, которые из всего окружения используют лишь несколько систем и их модулей. Использование выборочной блокировки позволит ускорить процесс тестирования.

Тип блокировки можно задать на уровне проекта или плана выполнения, тогда все его сценарии, для которых блокировка не задана, унаследуют данный тип блокировки. Если на сценарии выбрана блокировка другого типа, она будет использоваться для данного сценария.

События по установке и снятию блокировки отражаются в журналах событий, например:

- Попытка получить блокировку на окружение «DEFAULT_ENVIRONMENT»,
- Блокировка на окружение «DEFAULT_ENVIRONMENT» получена,
- Блокировка снята с окружения «DEFAULT_ENVIRONMENT».

5.3.2 Приоритеты и порядок выполнения сценариев

Приоритет сценария позволяет сгруппировать сценарии по степени важности. В приложении поддерживаются следующие типы приоритетов:

- Низкий (Minor),
- Нормальный (Normal),
- Высокий (Major),
- Критический (Critical).

При запуске проекта или плана выполнения сценарии выполняются в соответствии с приоритетами в порядке от «критического» до «низкого». Приложение пытается запустить несколько сценариев параллельно. Стратегия запуска определяется путем анализа запускаемого проекта или плана, и информация о ней выводится в журнал сообщений, например: *Стратегия для сценариев с приоритетом «NORMAL»: выполнить последовательно 5 сценариев, затем 24 параллельно блоками по 4 шт.*

Параллельные сценарии запускаются блоками. Разбиение сценариев на блоки происходит с учетом мощности системы и типов блокировок.

5.3.3 Создание тестового сценария

При создании тестового сценария укажите данные, описанные в таблице 5-2.

Таблица 5-2 Состав данных таблицы «Параметры сценария»

Поле	Описание
Код	Код сценария, который может быть использован внешней системой при управлении SOLAR Testing Toolkit через веб-сервис. Код должен быть уникальным в рамках проекта
Наименование	Название сценария, отображаемое в формах
Блокировка	Атрибут, который указывает на необходимость выполнения данного сценария в режиме блокировки. Возможные значения: <ul style="list-style-type: none"> • Полная — блокируется окружение целиком, • Выборочная — блокируются системы, которые используются каналами, тестируемыми на шагах сценария, • Не задано — используется атрибут блокировки плана выполнения или проекта, в рамках которого запускается сценарий. Если на уровне плана или проекта блокировка не задана, сценарий запускается в обычном режиме
Действие	Поле для управления ходом выполнения проекта. Возможные значения: <ul style="list-style-type: none"> • Выполнить, • Пропустить, • Остановить

Поле	Описание
Приоритет	Приоритет сценария. Используется в условиях выбора сценариев в планах выполнения и для определения порядка запуска сценариев плана или проекта. Возможные значения: <ul style="list-style-type: none"> • Низкий, • Нормальный, • Высокий, • Критический
Требование	Бизнес-требование ¹ , которое данный сценарий проверяет
Описание	Подробное описание сценария

Количество пропущенных сценариев отображается на странице **Результаты запуска проекта** (см. раздел 6.2 «Просмотр результатов запуска»).

В списке сценариев проекта доступен поиск сценария по коду и наименованию (см. рисунок 37).

Подробнее о возможностях поиска см. в разделе 2.5 «Табличные формы».

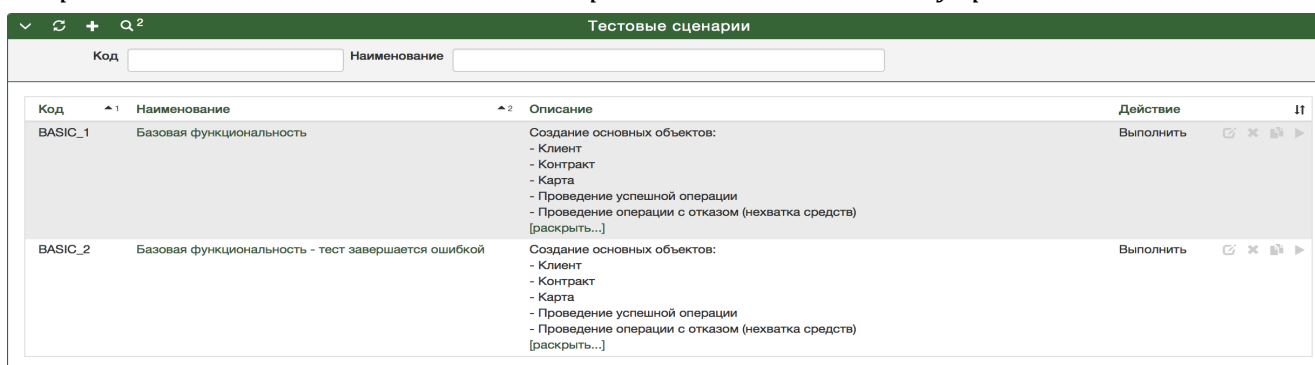


Рисунок 37 Поиск тестового сценария

Для работы с конкретным тестовым сценарием перейдите по ссылке в колонке **Наименование**. Добавьте необходимые тестовые шаги, как описано в разделе 5.4.

Для удобства работы со сценарием в списке шагов отображается тип шага, который составляется из других параметров шага:


- Отправка сообщения — шаг, на котором отправляется сообщение в процессинговую систему;
- Получение сообщения — шаг, на котором принимается сообщение от процессинговой системы;
- Отправка сообщения (Отложено) — шаг с отложенным выполнением, на котором отправляется сообщение в процессинговую систему;
- Получение сообщения (Отложено) — шаг с отложенным выполнением, на котором ожидается сообщение от процессинговой системы;
- Шаблон действия — шаг, вызывающий шаблон действия.

¹В текущей версии возможно указать ссылку на одно требование

5.3.4 Копирование тестового сценария

Копирование тестового сценария позволяет быстро создать новый сценарий на основе сценария из другого проекта.

Чтобы создать копию сценария:

1. Перейдите на страницу сценария или найдите сценарий в таблице **Тестовые сценарии** на странице проекта и нажмите кнопку  (Копировать сценарий).
2. Во всплывающем окне **Настройки копирования** задайте необходимые настройки:
 - В поле **Проект** выберите проект, в который следует скопировать сценарий. Для поиска проекта используйте регулярные выражения вида %SOME STRING%.
 - В поле **Наименование** введите новое имя сценария, если хотите сохранить копию с новым наименованием.
 - В поле **Количество** укажите требуемое количество копий.
 - Установите флажок **Заменить**, если вы хотите скопировать сценарий в уже существующий сценарий целевого проекта и выберите этот сценарий в поле **Сценарий**.
3. Нажмите **Сохранить**.

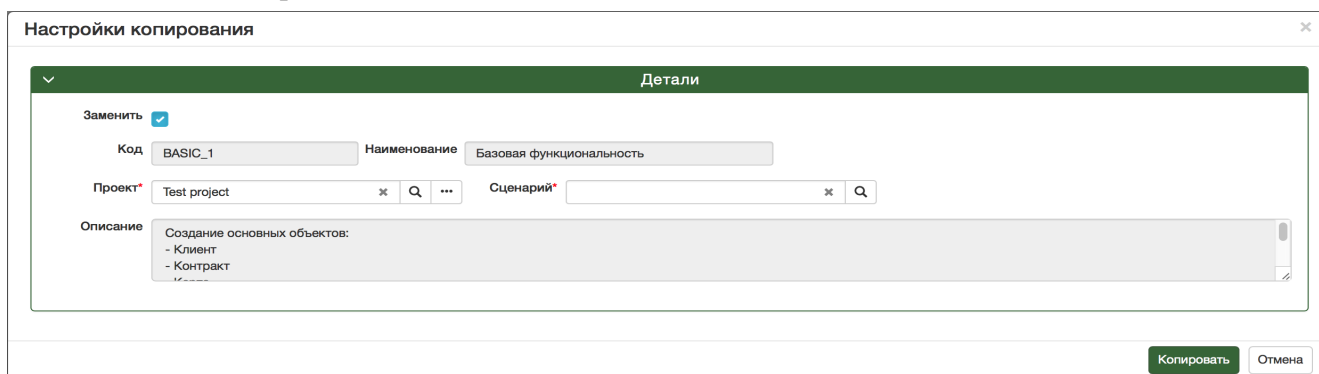


Рисунок 38 Настройки копирования сценария


Новый сценарий содержит все шаги первоначального. Если в проекте существует сценарий с таким же кодом или наименованием и опция **Заменить** не была выбрана, к коду или наименованию нового сценария добавляется суффикс, например, MC_FLOW (1). Новый сценарий доступен для редактирования.

5.3.5 Экспорт тестового сценария

Экспорт тестового сценария позволяет сохранить сценарий в виде файла формата JSON, который в дальнейшем может быть импортирован в приложение, установленное на другой системе. Формат файла изменился в версии 2.6.0 и не совместим с более ранними версиями.

При наличии в конфигурации приложения коннектора для работы с системой контроля версий (VCS) возможен экспорт сценария в репозиторий. В текущей версии поддерживается работа с системой Git.

Чтобы экспортировать тестовый сценарий,

1. Выберите пункт меню **Конфигурация** → **Проекты**.
2. Перейдите на страницу проекта, который содержит сценарий.
3. Откройте страницу сценария, перейдя по ссылке в колонке **Наименование**.
4. Нажмите кнопку  (Экспортировать сценарий) в правом верхнем углу.

5. В окне **Экспорт сценария** установите флажок **Включить зависимости**, чтобы добавить информацию о зависимостях сценария (шаблонах, наборах параметров, каналах, диалектах).
6. В списке **VCS** выберите коннектор для работы с VCS, если необходимо выгрузить сценарий в VCS.
7. Скачайте сценарий:
 - Нажмите кнопку **Скачать сценарий**, чтобы выгрузить сценарий в файловую систему или
 - Нажмите **Загрузить в VCS**, чтобы выгрузить в VCS.


Файл с именем scenario_ID.json по умолчанию создается в директории, выбранной для сохранения загружаемых файлов, или выгружается в репозиторий.

Для экспорта всех сценариев проекта следует экспортировать проект, как описано в разделе 5.1.2 «Экспорт проекта».

5.3.6 Импорт тестового сценария

Импорт тестового сценария позволяет добавить сценарий в приложение, используя исходный файл формата JSON, расположенный в файловой системе или репозитории VCS. Для успешного импортирования сценария в проект убедитесь, что проект содержит требование, проверяемое сценарием.

Чтобы импортировать сценарий,

1. Выберите пункт меню **Конфигурация → Проекты**.
2. Перейдите на страницу проекта, к которому следует добавить сценарий.
3. Нажмите кнопку  (Импортировать сценарий) в правом верхнем углу.
4. В окне **Импорт сценария** выберите файл:
 - нажмите кнопку **Выбрать файл** и выберите JSON файл, сохраненный в файловой системе, или
 - в списке **VCS** выберите название коннектора для работы с репозиторием, а в поле **Коммит** — идентификатор коммита.
5. Установите флажок **Заменять существующий**, если в проекте уже есть сценарий с таким кодом и наименованием и копии сценария не требуются.

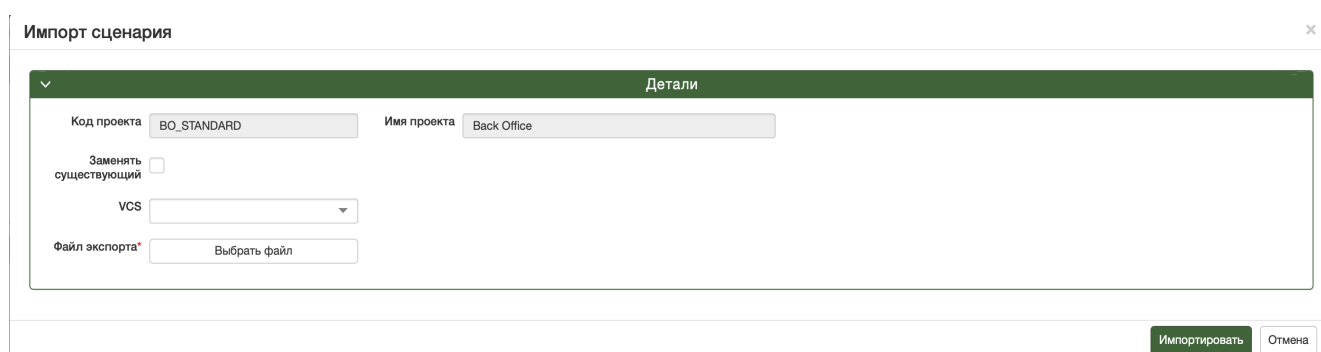


Рисунок 39 Импорт сценария

Чтобы избежать конфликта имен, когда в группе уже есть сценарий с кодом и наименованием импортируемого сценария, а флажок **Заменять существующие** не установлен, к коду и наименованию нового сценария добавляется суффикс по аналогии с операцией копирования.

5.3.7 Разметка сценариев с помощью функциональных областей

Для любого сценария вы можете указать функциональные области, тестируемые данным сценарием.

Чтобы добавить функциональную область к сценарию:

1. Откройте форму выбранного сценария (см. рисунок 40).

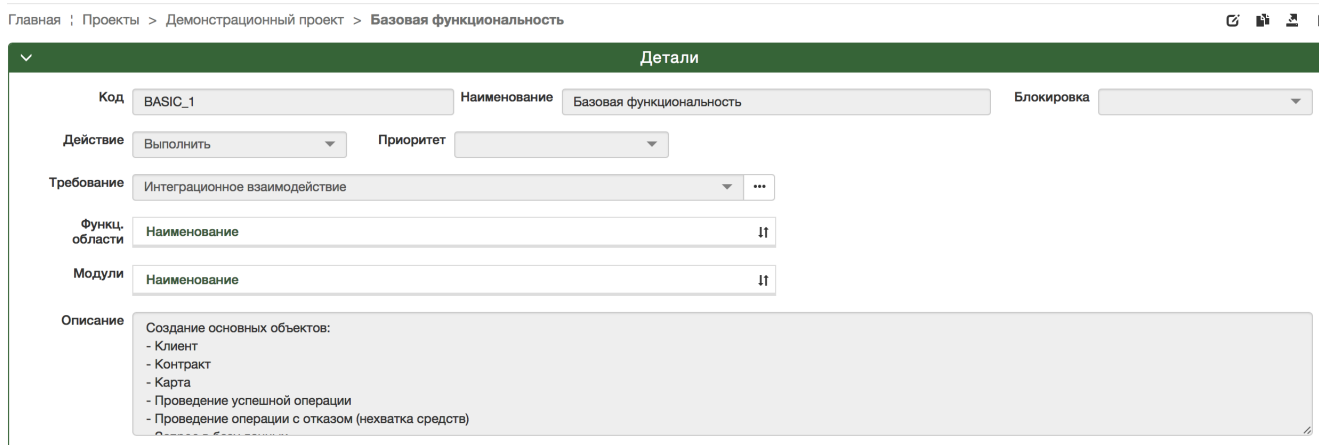


Рисунок 40 Функциональные области сценария

2. В выпадающем списке **Функциональные области** выберите функциональную область.
При отсутствии требуемой функциональной области отредактируйте список функциональных областей (см. 4.2 «Функциональные области») и повторите описанную последовательность действий.

5.4 Тестовые шаги

Список тестовых шагов, разработанных для сценария, доступен на странице сценария в форме **Тестовые шаги**.

5.4.1 Создание тестового шага

Чтобы создать тестовый шаг, добавьте запись в форму **Тестовые шаги** и укажите данные, описанные в таблице 5-3:

Таблица 5-3 Параметры шага

Параметр	Описание
№	Порядковый номер шага ¹
Шаблон действия	Поле заполняется в том случае, если на данном шаге должен быть запущен шаблон действия. <i>Подробнее о создании шаблонов см. в разделе 4.5 «Шаблоны действия».</i>
Канал	Поле заполняется в том случае, если на данном шаге должно быть принято или отправлено сообщение в соответствующий канал

¹При сохранении шага, чей номер не заканчивается на 0, приложение меняет нумерацию всех шагов сценария, чтобы номера заканчивались на 0.

Параметр	Описание
Действие	<p>Поле для управления ходом сценария. Возможные значения:</p> <ul style="list-style-type: none"> • Выполнить, • Пропустить, • Остановить сценарий <p>Для визуального выделения шагов в списке шагов используется подсветка фона: шаги с действием «Пропустить» — серый фон, шаги с действием «Остановить» — красный фон</p>
Фаза выполнения	<p>Параметр указывает, к какой фазе выполнения относится шаг:</p> <ul style="list-style-type: none"> • Начальная — начальный шаг, • Основная — шаг относится к основной фазе выполнения, • Финальная — финальный шаг
Начальная задержка (мс)	Отсрочка выполнения шага в миллисекундах
Итерации	Количество повторных запусков шага
Направление	Поле с выбором из двух значений: «Исходящее», «Входящее». Поле отображается, если на шаге отправляется или принимается сообщение
Запрос	<p>Поле с выбором из двух значений: «Обязательный», «Отсутствует». Поле отображается, если для шага выбрано направление «Входящее».</p> <ul style="list-style-type: none"> • Выберите Отсутствует, если на шаге надо проверить, что нет входящего сообщения; • Выберите Обязательный, если сообщение на шаге обязательное или опциональное
Ответ	<p>Ответ — поле с выбором из двух значений: «Обязательный», «Отсутствует». Поле отображается, если на шаге отправляется или принимается сообщение.</p> <ul style="list-style-type: none"> • Выберите Обязательный для входящего шага с обязательным сообщением, • Выберите Отсутствует для входящего шага с опциональным сообщением
Отложенное выполнение	Флажок управляет порядком выполнения шагов в сценарии (см. раздел 3.2.2 «Порядок выполнения тестовых шагов»)

Параметр	Описание
Задержка ответа	<p>Период времени между получением входящего сообщения и отправкой ответа. Параметр отображается только для входящих шагов.</p> <p>Значение задержки ответа выбирается из заданного интервала с заданной вероятностью.</p> <ul style="list-style-type: none">• Без задержки — ответ будет отправлен без задержки,• Равномерное распределение — задержка ответа задается функцией равномерного распределения, которая определяется параметрами:<ul style="list-style-type: none">◦ Мин. задержка ответа (мс),◦ Макс. задержка ответа (мс),• Нормальное распределение — задержка ответа задается функцией нормального распределения, которая определяется параметрами:<ul style="list-style-type: none">◦ Мин. задержка ответа (мс),◦ Макс. задержка ответа (мс),• Бета-распределение — задержка ответа задается функцией бета-распределения, которая определяется параметрами:<ul style="list-style-type: none">◦ Мин. задержка ответа (мс),◦ Макс. задержка ответа (мс),◦ α(смещение к максимуму),◦ β(смещение к минимуму). <p>В зависимости от параметров α и β графики плотности вероятности имеет различный вид:</p> <ul style="list-style-type: none">• $\alpha < 1, \beta < 1$ – график выпуклый и уходит в бесконечность на границах. С большой вероятностью значение задержки ответа будет близко к максимальной задержке ответа.• $\alpha < 1, \beta \geq 1$ или $\alpha = 1, \beta > 1$ – график строго убывающий. С большой вероятностью значение задержки ответа будет близко к минимальной задержке ответа.• $\alpha = 1, \beta = 1$ – график совпадает с графиком плотности стандартного непрерывного равномерного распределения. Эквивалентно выбору значения Равномерное распределение в выпадающем списке.• $\alpha = 1, \beta < 1$ или $\alpha > 1, \beta \leq 1$ – график строго возрастающий. С большой вероятностью значение задержки ответа будет близко к максимальной задержке ответа.• $\alpha, \beta > 1$. С большой вероятностью время задержки будет близко к значению, находящемуся между минимальным и максимальным. Если $\alpha > \beta$, то значение задержки ответа будет смещается в сторону максимального, но не достигает его. Если $\alpha < \beta$, то данное значение смещается в сторону минимального, не достигая его. Если $\alpha = \beta$, то с большой вероятностью время задержки будет близко к среднему арифметическому максимального и минимального значения. <p>Подробнее о бета-распределении и соответствующих графиках см. на</p>

Параметр	Описание
	<p>странице https://ru.wikipedia.org/wiki/Бета-распределение</p> <p>Для корректного выполнения входящего шага с заданной задержкой ответа убедитесь, что в свойствах канала задано Время ожидания ответа, значение которого больше либо равно максимальной задержке ответа, заданной для шага (см. раздел 4.1.1)</p>
Наименование	Название шага, отображаемое в формах
Время ожидания (мс)	Количество времени, в течение которого ожидается сообщение из внешней системы (в миллисекундах)
Описание	Подробное описание шага
Данные шага	<p>Сообщение, которое должно быть отправлено в процессинговую систему. В сообщении могут быть использованы контекстные переменные (см. 5.4.6 «Контекстные переменные») и системные функции (см. 5.4.8 «Контекстные функции»). Формат сообщения зависит от диалекта канала.</p> <p><i>Подробнее о форматах сообщений см. в разделе 5.4.9 «Типы шагов»</i></p>

На рисунке 41

Рисунок 41 Создание тестового шага

Для работы с конкретным тестовым шагом перейдите по ссылке в колонке **Наименование**.

Поддерживается работа с шагами, на которых не вызывается шаблон и для которых не указан канал. Для таких шагов возможна установка переменных и выполнение проверок с источником «Выражение».

Форма **Правила выбора входящего сообщения** (см. рисунок 42) позволяет задать критерий выбора входящего сообщения для обработки на данном шаге.

Правило выбора входящего сообщения

Детали

Наименование*

Выражение* Формат

Тип данных* Проверка*

Ожидаемое значение Формат

Сохранить Отмена

Рисунок 42 Правило выбора входящего сообщения

Шаги, позволяющие отправлять сообщения в процессинговую систему, также обрабатывают данные, полученные в ответных сообщениях. Для этого предусмотрены следующие разделы шага:

- Установка контекстных переменных,
- Проверки.

Подробнее о контекстных переменных и проверках см. в разделах 5.4.6 «Контекстные переменные» и 5.4.7 «Проверки» соответственно.

5.4.2 Создание тестового шага на основе шаблона

Чтобы создать шаг, вызывающий шаблон действия:

1. Выберите в поле **Шаблон действия** требуемый шаблон.
2. Заполните обязательные поля данных.
3. Сохраните шаг.
4. Выберите шаг по ссылке в колонке **Наименование**.
5. В таблице **Входящие параметры шаблона** отображаются все переменные используемого шаблона.
 - Переменная — имя переменной, указанное в шаблоне.
 - Выражение — ссылка на поле входящего сообщения или выражение (в зависимости от источника данных). По умолчанию значение равно значению, указанному в поле шаблона **Значение по умолчанию**. В качестве выражения могут быть использованы контекстные переменные (см. 5.4.6 «Контекстные переменные») и системные функции (см. 5.4.8 «Контекстные функции»). Формат данного поля зависит от диалекта канала, в который было отправлено сообщение.
Подробнее о диалектах см. в разделе 5.4.9 «Типы шагов».
 - Тип данных — тип данных переменной, указанный в шаблоне.
 - Описание — описание переменной, указанное в шаблоне.
6. Задайте **Выражение** для всех переменных, используемых в шаблоне (см. рисунок 43).

Конфигурация Demo Администратор

Главная ; Проекты > Демонстрационный проект > Базовая функциональность > Успешная покупка

Детали

№* Шаблон действия 🔍 ...

Фаза* Действие*

Задержка Итерации

Наименование*

Описание
 Вызов шаблона покупки для карты из предыдущего шага.
 Проверяется код ответа.

Переменная	Выражение	Тип данных	Описание
pan	\${cardNumber}	Строка	PAN
cardExp	\${expiryDate}	Строка	Срок действия карты

Рисунок 43 Создание шага, вызывающего шаблон действия

- В таблице **Исходящие параметры шаблона** укажите локальные имена возвращаемых переменных, если требуется.

5.4.3 Создание финального тестового шага

Финальный тестовый шаг обязательно выполняется в конце сценария: и в случае успешного выполнения сценария, и в случае возникновения ошибки.


Использование финального тестового шага позволяет восстановить состояние данных, сохраненных перед запуском тестового сценария.

Чтобы создать финальный шаг, добавьте запись в форму **Финализация** и укажите параметры шага (см. раздел 5.4.1 «Создание тестового шага»).

5.4.4 Копирование шагов

Копирование шага позволяет быстро добавить к сценарию или шаблону шаг, существующий в другом сценарии или шаблоне. Таким способом можно создать до 10 копий шага включительно.

Чтобы скопировать тестовый шаг:

- Нажмите кнопку  (Копировать) в таблице **Тестовые шаги** на странице сценария или на странице тестового шага.
- Во всплывающем окне **Настройки копирования** выберите место назначения копируемого шага: проект и сценарий внутри проекта или шаблон; В поле **Количество** укажите количество копий.
Для поиска места назначения используйте регулярные выражения вида %SOME STRING%.
- Нажмите **Сохранить**.

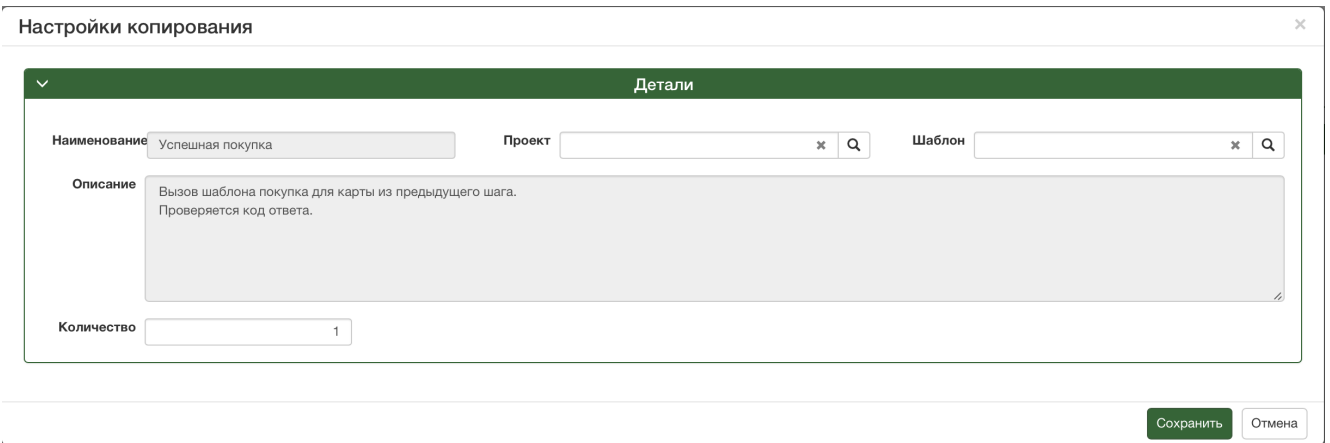


Рисунок 44 Настройка копирования тестового шага

5.4.5 Соответствие тестового шага на основе шаблона и вызываемого шаблона

Будьте внимательны при изменении параметров шаблона или замене шаблона в тестовом шаге. Все параметры шаблона, определенные в тестовом шаге, должны соответствовать параметрам вызываемого шаблона.

Если в вызываемом шаблоне отсутствует параметр, используемый в тестовом шаге, такой параметр выделяется желтым цветом на странице шага (см. рисунок 45). При этом параметр продолжает использоваться при выполнении шага.

Если вызываемый шаблон и тестовый шаг содержат параметры с одинаковым названием, но различными типами данных, такие параметры выделяются красным цветом на странице шага (см. рисунок "Соответствие тестового шага на основе шаблона и вызываемого шаблона" выше). Выполнение шага завершается с ошибкой. Для устранения ошибки скорректируйте тип данных параметра на странице шага или странице шаблона.

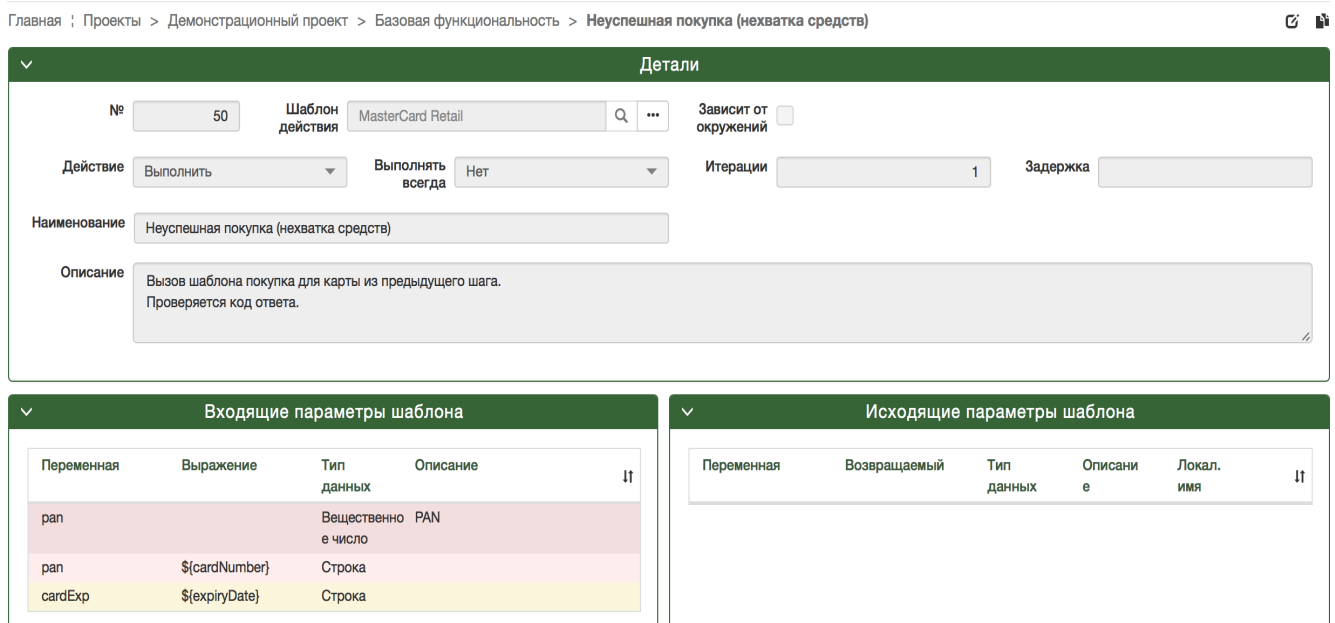


Рисунок 45 Параметры в шаге не соответствуют вызываемому шаблону

5.4.6 Контекстные переменные

Контекстные переменные позволяют использовать информацию, полученную из процессинговой системы на предыдущих тестовых шагах того же сценария.

Для обращения к контекстной переменной используйте специальные символы $\{\}$, например, $\{\text{CardNumber}\}$. Также можно указать формат контекстной переменной, например, $\{\text{CurrentDate:уууу.ММ.дд}\}$. Название переменной должно начинаться с буквы английского алфавита и может состоять из букв английского алфавита, цифр и символа подчеркивания.

Подробнее о поддерживаемых типах форматирования см. в разделе 5.7 «Форматирование».

Установка значения контекстной переменной осуществляется после выполнения тестового шага.

При инициализации контекстной переменной заполните поля данных в окне **Установка переменных**, изображенном на рисунке 46.

Рисунок 46 Установка контекстных переменных

- **Наименование** — название переменной, отображаемое в формах,
- **Переменная** — название переменной, по которому к ней можно обращаться из контекста. Указывается без специальных символов, например, CardNumber;
- **Тип** — тип данных переменной. Поддержаны следующие варианты:
 - Строка,
 - Целое число,
 - Вещественное число,
 - Логическое да/нет,
 - Дата,
 - Время,
 - Дата и время,
 - Таблица,
 - Объект — составной тип данных, в котором хранятся значения нескольких полей (см. раздел 5.4.6.1).
- **Источник** — источник данных. Возможные значения:
 - Входящее сообщение,
 - Исходящее сообщение,
 - Выражение.
- **Выражение** — выражение для выбора полей ответного сообщения, значения которых надо сохранить. Возможные значения:
 - ссылка на конкретное поле в ответном сообщении,
 - “/”, если необходимо сохранить все поля в переменную «таблица»,
 - выражение, формат которого зависит от диалекта используемого канала (Подробнее о диалектах см. в разделе 5.4.9 «Типы шагов».)
Выражение может содержать контекстные переменные (см. 5.4.6 «Контекстные переменные») и системные функции (см. раздел 5.4.8 «Контекстные функции»).

- **Формат** — формат данных, который следует применить к выражению (подробнее о синтаксисе данного поля см. в разделе 5.7 «Форматирование»).

5.4.6.1 Пример сохранения контекстной переменной типа «Объект»

Допустим, на шаге получено сообщение, фрагмент которого приведен в примере 5-1.

Пример 5-1 Фрагмент полученного XML-сообщения

```
<?xml version="1.0" encoding="UTF-8"?>
<file>
  <key>name</key>
  <value>value</value>
  <composite>
    <compositeKey>compositeName</compositeKey>
    <compositeValue>compositeValue</compositeValue>
  </composite>
</file>
```

Чтобы сохранить данные элемента `composite`, состоящего из двух полей, в контекстной переменной, необходимо создать переменную типа «Объект» и указать в поле **Выражение** xPath к элементу (см. рисунок 47).

Рисунок 47 Создание переменной типа «Объект»

В результате будет создана переменная `objVar` со значением

```
{"compositeKey": "compositeName", "compositeValue": "compositeValue"}
```

Для обращения к полям переменной в проверках или при установке других контекстных переменных следует использовать синтаксис `${/objVar/compositeKey}`.

5.4.6.2 Контекстные переменные времени выполнения

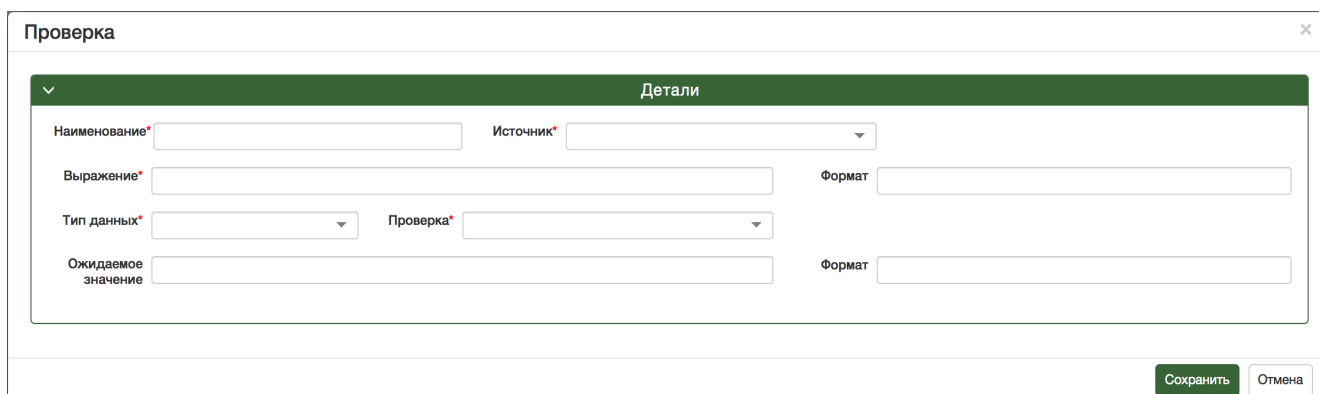
Приложение поддерживает стандартные контекстные переменные времени выполнения:

- `runtime/step/execution/duration` — длительность выполнения шага (в миллисекундах),
- `runtime/scenario/execution/duration` — длительность выполнения сценария (в миллисекундах).

5.4.7 Проверки

После установки контекстных переменных выполняется проверка данных, полученных в ответном сообщении.

Чтобы создать проверку, заполните поля формы, изображенной на рисунке 48

**Рисунок 48 Создание проверки**

- Наименование — название проверки, отображаемое в формах,
- Источник — поле с выбором из списка:
 - Входящее сообщение,
 - Выражение.
- Выражение — ссылка на поле входящего сообщения или выражение (в зависимости от источника данных). В качестве выражения могут быть использованы контекстные переменные (см. 5.4.6 «Контекстные переменные») и системные функции (см. 5.4.8 «Контекстные функции»). Формат данного поля зависит от диалекта канала, в который было отправлено сообщение. Подробнее о диалектах см. в разделе 5.4.9 «Типы шагов».
- Формат (выражения) — формат данных, который следует применить к выражению (подробнее о синтаксисе данного поля см. в разделе 5.7 «Форматирование»),
- Тип данных — тип данных выражения. Поддержаны следующие варианты:
 - Строка,
 - Целое число,
 - Вещественное число,
 - Логическое да/нет,
 - Дата,
 - Время,
 - Дата и время,
 - Таблица.
- Проверка — тип сравнения. Поддержаны следующие варианты:
 - Равно,
 - Не равно,
 - Меньше,
 - Меньше или равно,
 - Больше,
 - Больше или равно,
 - Входит в список (значения указываются через запятую),
 - Содержит подстроку,
 - Начинается со строки,
 - Заканчивается строкой,
 - Регулярное выражение (см. спецификацию <https://docs.oracle.com/javase/8/docs/api/java/util/regex/Pattern.html>),
 - Сравнение таблиц (см. 5.4.7.1 «Проверка переменной типа «Таблица»»),
 - Отсутствует (NULL),
 - Присутствует (NOT NULL).

- Ожидаемое значение — значение, которому должно соответствовать выражение. В качестве значения может быть указана контекстная переменная (см. 5.4.6 «Контекстные переменные»),
- Формат (ожидаемого значения) — формат данных, который следует применить к ожидаемому значению (подробнее о синтаксисе данного поля см. в разделе 5.7 «Форматирование»).

Если полученное значение не соответствует ожидаемому, проверка завершается с ошибкой. На практике в этом случае часто требуется изменить ожидаемое значение на полученное в результате проверки.


Чтобы сохранить полученное значение в качестве ожидаемого, нажмите кнопку  (Задать как ожидаемое), отображаемую в правой части формы проверки. Указанное действие доступно только для проверок, завершившихся с ошибкой, перечисленных в таблице 5-4

Таблица 5-4 Результат операции «Задать как ожидаемое»

Тип проверки	Результат операции
Равно	Полученное значение сохраняется, если ожидаемое не содержало символы-заполнители
Начинается со строки	Ожидаемое значение заменяется на максимальный общий префикс строки
Заканчивается строкой	Ожидаемое значение заменяется на максимальный общий суффикс строкой
<ul style="list-style-type: none"> • Меньше, • Меньше или равно, • Больше, • Больше или равно 	Полученное значение сохраняется

Если алгоритм проверки менялся после получения результатов проверки, корректное сохранение результатов проверки не гарантируется.

Примеры использования см. в разделе 5.4.9 «Типы шагов».

Благодаря тому, что все сообщения в формате JSON конвертируются в XML формат, для ссылки на поле входящего JSON-сообщения при заполнении поля **Выражение** можно использовать синтаксис языка XPath 2.0. При конвертации значений ключей, начинающихся с цифры, добавляется символ подчеркивания. Например, исходное сообщение выглядит, как показано в примере 5-2.

Пример 5-2 Входящее JSON-сообщение

```
{
  "0abc": "value",
  "1": {
    "2": "value"
  }
}
```

Сконvertированное сообщение показано в примере 5-3.

Пример 5-3 Сконvertированное сообщение

```
<?xml version="1.0" encoding="UTF-8"?>
<json>
  <_0abc>value</_0abc>
  <_1>
    <_2>value</_2>
  </_1>
</json>
```

Поле "0abc" конвертируется в <_0abc>. Обращаться к такому полю можно как с символом подчеркивания /json/_0abc, так и без него: /json/0abc.

Подробнее о языке XPath 2.0, см. спецификацию <https://www.w3.org/TR/xpath20/>.

5.4.7.1 Проверка переменной типа «Таблица»

В терминах приложения таблицей является матрица значений: в колонках расположены названия полей объектов, составляющих таблицу, в строках – значения этих полей.

При создании проверки таблицы учтите следующие особенности заполнения полей:

- **Выражение** – используйте “/” в случае источника данных «Входящее сообщение», это позволит выбрать все полученные данные,
- **Тип данных** – «Таблица»;
- **Ожидаемые значения** – структура проверяемого объекта и тип проводимой проверки для каждой составляющей объекта:
 - Сначала добавьте **Переменные**, соответствующие составляющим ожидаемого объекта, укажите их **Тип данных** и **Тип проверки**.
 - Затем добавьте необходимое количество строк, соответствующее размеру таблицы, и введите ожидаемые значения объекта.

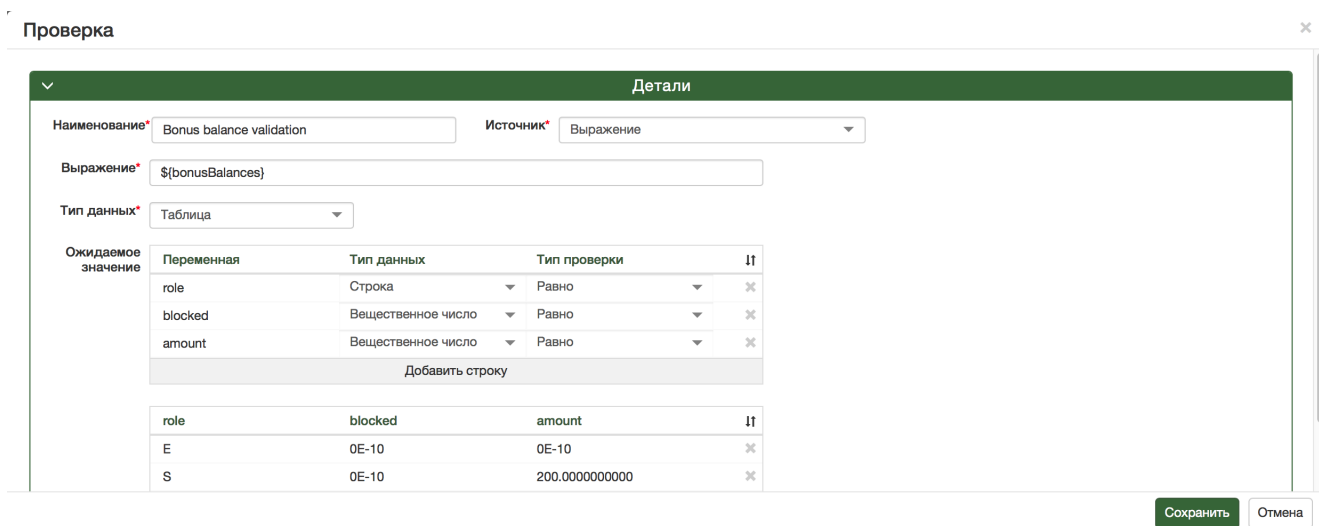


Рисунок 49 Проверка таблицы

5.4.8 Контекстные функции

Приложение поддерживает predetermined list of context functions, which are used along with context variables (see 5.4.6 «Context variables»). Syntax of calling a context function: \${function}.

In the current version of the application, functions described in table 5-5 are available.

Таблица 5-5 Поддерживаемые контекстные функции

Функция	Описание
sysdate()	Текущая дата. Функция может быть использована для генерации уникального в рамках тестового скрипта значения, например, RRN (Retrieval Reference Number) транзакции или номера XML-сообщения. Для этих целей используйте выражение <code>\${sysdate():ddMMYYYY_HHmmsSSS}</code>
sequence()	Следующее значение глобальной (сохраняемой в рамках приложения) последовательности целых чисел из диапазона <code>[0; Long.MAX_VALUE]</code> . Функция <code>sequence</code> используется для генерации уникального значения в рамках приложения. Примеры: <code>\${sequence()}</code> выдает целое значение, например 239. Следующий вызов функции <code>\${sequence()}</code> выдаст 240
sequence(int modulus)	Остаток от деления следующего значения глобальной последовательности на значение, переданное в качестве параметра. Примеры: <code>\${sequence(1000000)}</code> выдает целое значение из диапазона <code>0..999999</code> , например 241. Для получения шестизначного значения можно применить форматирование. <code>\${sequence(1000000):%06d}</code> выдает целое шестизначное значение. Если остаток от деления значения последовательности на 1000000 меньше 6 символов, он дополняется нулями слева, например 000242
random()	Случайное значение из диапазона <code>[0; Long.MAX_VALUE]</code>
random(int maxValue)	Случайное значение из диапазона <code>[0; maxValue]</code>
uuid()	UUID-строка версии 4 (случайная), например 123e4567-e89b-12d3-a456-426655440000
concat(value1, value2)	Конкатенация строковых литералов. Примеры: <code>var1 = null</code> (не задана), <code>var2 = '7'</code> <code>\${concat('abc', 'qwe')}</code> =abcqwe <code>\${concat('%07d:', var2)}</code> = %07d:7 <code>\${concat('abc', var1)}</code> = abc

Функция	Описание
convert(value, 'source_encoding', 'target_encoding')	<p>1. Преобразование шестнадцатеричных значений в строковые и наоборот, где:</p> <ul style="list-style-type: none"> value — значение в исходной кодировке: числовой литерал, строковый литерал или переменная; 'source_encoding' — исходная кодировка, 'target_encoding' — результирующая кодировка. <p>2. Конвертирование из json в xml и наоборот.</p> <p>Примеры:</p> <pre> \${convert(3638, 'hex', 'ascii')} = 68 \${convert('F6F8', 'hex', 'ebcdic')} = 68 \${convert(68, 'ebcdic', 'hex')} = F6F8 \${convert(68, 'ascii', 'hex')} = 3638 \${convert(myValue, 'json', 'xml')} — из json в xml \${convert(myValue, 'xml', 'json')} — из xml в json </pre>
unicodeDecode	<p>Принимает символ, закодированный как hex, decimal, html и возвращает раскодированный символ.</p> <p>Примеры:</p> <pre> Hello\${unicodeDecode('&nbsp;')}World! = Hello World! Hello\${unicodeDecode('&#xa0;')}World! = Hello World! Hello\${unicodeDecode('&#160;')}World! = Hello World! </pre>
deflateEncode (value)	<p>Кодирование шестнадцатеричного значения по алгоритму Deflate. Возвращает закодированное шестнадцатеричное значение.</p> <pre> \${deflateEncode('4F726967696E616C204D657373616765')} = 789CF32FCA4CCFCC4BCC51F04D2D2E4E4C4F050033BD061B </pre>
deflateDecode (value)	<p>Декодирование шестнадцатеричного значения по алгоритму Deflate. Возвращает декодированное шестнадцатеричное значение.</p> <pre> \${deflateDecode ('789CF32FCA4CCFCC4BCC51F04D2D2E4E4C4F050033BD061B')} = 4F726967696E616C204D657373616765 </pre>
base64Encode	<p>Кодирование шестнадцатеричного значения по алгоритму Base64. Возвращает закодированное шестнадцатеричное значение.</p> <pre> \${base64Encode('4F726967696E616C204D657373616765')} = 54334A705A326C75595777675457567A6332466E5A513D3D </pre>
base64Decode	<p>Декодирование шестнадцатеричного значения по алгоритму Base64. Возвращает декодированное шестнадцатеричное значение.</p> <pre> \${base64Decode ('54334A705A326C75595777675457567A6332466E5A513D3D')} = 4F726967696E616C204D657373616765 </pre>

Функция	Описание
<p>paRes(pan, encodedPaReq, status, eci, cavvAlgorithm, cavvValue)</p>	<p>Генерация XML-представления PARes (Payer Authentication Response).</p> <ul style="list-style-type: none"> • pan — Primary Account Number, • encodedPaReq — encoded Payer Authentication Request, • status — статус транзакции, • eci — Electronic Commerce Indicator, • cavvAlgorithm — алгоритм CAVV, • cavvValue — значение CAVV. <p>См. пример 5-5, содержащий результат вызова функции</p>
<p>paReqEncode</p>	<p>Кодирование сообщения PAReq (Payer Authentication Request).</p> <p><code>\${paReqEncode('<xml>text</xml>')} = eJyzqjNsStJrSix0QexACz3BYs=</code></p> <p>Вызов <code>\${paReqEncode('...')}</code> эквивалентен <code>\${convert(base64Encode(deflateEncode(convert('...', 'ascii', 'hex'))), 'hex', 'ascii')}</code></p>
<p>paResDecode</p>	<p>Декодирование закодированного значения PARes (Payer Authentication Response).</p> <p><code>\${paResDecode('eJyzqjNsStJrSix0QexACz3BYs=')} = <xml>text</xml></code></p> <p>Вызов <code>\${paResDecode('...')}</code> эквивалентен <code>\${convert(deflateDecode(base64Decode(convert('...', 'ascii', 'hex'))), 'hex', 'ascii')}</code></p>
<p>xPath (xmlString, xpathExpression)</p>	<p>Применяет выражение xpath к строке XML.</p> <ul style="list-style-type: none"> • xmlString — строка XML, • xpathExpression — выражение xpath. <p><code>\${xpath('<tag>text</tag>', '/tag')} = text</code></p> <p>Может использоваться совместно с <code>convert (myValue, 'json', 'xml')</code> для разбора/анализа json-столбцов выборки</p>
<p>isoGenerateField</p>	<p>Генерация данных для полей ISO 8583 в виде случайной последовательности заглавных латинских букв и цифр длиной 6 символов.</p> <p>Пример использования:</p> <p><code>\${isoGenerateField(38)}</code> или <code>\${isoGenerateField('38')}</code></p>

Пример 5-5 содержит результат вызова функции paRes с параметрами pan=1234567890123456, encodedPaReq из примера 5-4, status='Y', eci='02', cavvAlgorithm=1, cavvValue='AAABBEg0VhIOVniQEjRWAAAAAAAAA='.

Пример 5-4 Значение параметра encodedPaReq

```
<ThreeDSecure>
  <Message id="id_23002931_2334_4d13_bef2_001a8e56d59b">
    <PAReq>
      <version>1.0.2</version>
      <Merchant>
        <acqBIN>014565</acqBIN>
        <merID>123321</merID>
        <name>Test Merchant</name>
        <country>840</country>
```

```

    <url>https://example.com</url>
  </Merchant>
  <Purchase>
    <xid>DCpfekSLRT6JV0MOVmWtxQAAAAA=</xid>
    <date>20190219 14:12:07</date>
    <amount>6.0000000000</amount>
    <purchAmount>6.0000000000</purchAmount>
    <currency>840</currency>
    <exponent>0</exponent>
  </Purchase>
  <CH>
    <acctID>A0fTY+pKUTu/6hcZWZJiAA==</acctID>
    <expiry>****</expiry>
  </CH>
</PAREq>
</Message>
</ThreeDSecure>

```

Пример 5-5 Результат вызова функции pages

```

<?xml version="1.0" encoding="UTF-8"?>
<ThreeDSecure>
  <Message id="id_23002931_2334_4d13_bef2_001a8e56d59b">
    <PAREs id="pares_id_23002931_2334_4d13_bef2_001a8e56d59b">
      <version>1.0.2</version>
      <Merchant>
        <acqBIN>014565</acqBIN>
        <merID>123321</merID>
      </Merchant>
      <Purchase>
        <xid>DCpfekSLRT6JV0MOVmWtxQAAAAA=</xid>
        <date>20190219 14:12:07</date>
        <purchAmount>6.0000000000</purchAmount>
        <currency>840</currency>
        <exponent>0</exponent>
      </Purchase>
      <pan>0000000000003456</pan>
      <TX>
        <time>20200903 09:53:56</time>
        <status>Y</status>
        <eci>02</eci>
        <cavvAlgorithm>1</cavvAlgorithm>
        <cavv>AAABBEg0VhI0VniQEjRWAAAAAAA=</cavv>
      </TX>
    </PAREs>
    <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
      <SignedInfo>
        <CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315" />
        <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
        <Reference URI="#pares_id_23002931_2334_4d13_bef2_001a8e56d59b">
          <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
          <DigestValue>wvqIq5RNm/r3kVUBPt6aQw7jG2g=</DigestValue>
        </Reference>
      </SignedInfo>

```

```
<SignatureValue>qvRW902QSOscvdvBE/t/peF2aRgFophVXrGz5mbnJXajGW+rklM7/nqz3J+2CF
8Bjb3weAM5X4
pUN4rtTv2qBojqPp2ONBdnmlKdgB2sBMm6vDdCTMlWvmztGVjRCkfgH6A4kE10W8ZXJ6geZoofCX
58+jR2m+Cw1fsfnT8+SN4P+mr88CsFe4NYETK+eYBhOeOfn0Rl6txdv7UsW+zrYPdEEkXnhXneki
VuMEQ1H0hTFzklHG8/2Hrm6P0iijoiu16uFOmEM+WvuVzVzuPwuHPS+LbhyMM5NAa1D3ouIwNaXS
6oswqAW9XdMSGPubJJ8QA9XNjOtsDREQfffsEg==</SignatureValue>
  <KeyInfo>
    <X509Data>

<X509Certificate>MIICxzCCAa+gAwIBAgIETvkROzANBgkqhkiG9w0BAQsFADAUMRIWEAYDVQQDEw1
sb2NhbGhvc3Qw
HhcNMTgxMDAzMTYwMzUzWhcNNDYwMjE4MTYwMzUzWjAUMRIWEAYDVQQDEw1sb2NhbGhvc3QwggEi
MA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQL7Ch020s8GIWYGVIT2Ho66QC6803pIz8qFNPM
DHq6zvYuFX5/H/+Cygelxy9BluEViNaIpr6yTcVavJWVvnp2EgoNNv52IUBOCLQyenOGHAF4p7Rc
nROKqX6jj1hufIKAWi/gDxaZ/V0wrwzZuqrrhGV2+zOPpar4iYphvEcMQzvgufAhpiF7LFDjAHjr
e2XdnWF5H6Vs1ceNkUAbYcTSKJS81/qYF429YG3yFLbCWzDIGGEaLj1AfynuQZ4dIyBbDeRLZmKw
wylszKx9GsAOI+EX51p0gNk/Cv6y/oIjWWBN3Btf4kZlahepQyrn1jgZ7UF77vEdz3mYl8/1iEL5
AgMBAAGjITAfMB0GA1UdDgQWBBDtu84uw2gRQxKAXYV6uvoPHGOzrjANBgkqhkiG9w0BAQsFAAOC
AQEAWi4tDk+nTQVXCZuSkX4pKW6mhDjwwB0eKaJ81XI/7k+POjrrtEzSJKF3jsEZHmRnGwUMV5Un
dwCF7GpJH3q6jyR3zucFayhbz4YdaLWT6+I9lJobURbezQAB+1TFFHR0fqZZZRBuNv7pEEzWxpA6
HgF3wHtpCUt0UTlXE7Xt2E3QPzW144kbgfW09NrVlJO1UF2Hbra88CcgjCe65oG8sTOe8uyGn44
tIDVRgK0Qc1fRrWA4ZoL7XCkqBaQzr331s9OG3NjL2dHx9E/cFWOGjqI6wYiiaE6rTH8tdT+FrWu
U8F0VnZLyvEMwp8F7vVx+cIxTnBJOvGVAMQNLuudWw==</X509Certificate>
  </X509Data>
</KeyInfo>
</Signature>
</Message>
</ThreeDSecure>
```

5.4.9 Типы шагов

5.4.9.1 Работа с диалектом ISO

Для создания ISO-сообщения укажите в ячейке **Данные шага** значения всех значимых полей (см. рисунок 50, а при наличии бинарного дампа данных воспользуйтесь функцией автоматического разбора (см. раздел 5.4.9.1.1).

Установите флажок **Разрешить ошибки**, чтобы ошибки протокола не приводили к ошибочному завершению тестового сценария.

Детали

№* 10 Канал MasterCard Зависит от окружений

Действие* Выполнить Итерации 1 Задержка

Направление* Исходящее Ответ* Обязательный

Наименование* Операция покупки

Описание Шаблонная операции покупки. Параметры: - pan

Поле	Значение
000 - MTID	0100
002 - Primary Account Number (PAN)	\$(pan)
003 - Processing Code	000000
004 - Amount, Transaction	000000090000
006 - Amount, Cardholder Billing	000000090000
007 - Transmission Date and Time	0000000000
010 - Conversion Rate, Cardholder Billi	1000000
011 - Systems Trace Audit Number	000000

Рисунок 50 Создание ISO-сообщения

Поле 55 состоит из нескольких тегов. Его можно заполнить одним из следующих способов:

- выбрать в списке **Поле/Тег** элемент «055 — Integrated Circuit Card (ICC) System-Related Data» и задать строку, содержащую значения всех тегов;
- выбрать последовательно элементы, соответствующие одному тегу, и задать значение каждого тега отдельно.

Обратите внимание: если поля 7, 11, 37 заполнены нулями или их значения не указаны при создании шага или не получены на входящем шаге, то данные поля автоматически заполняются ISO-адаптером.

При установке контекстных переменных и создании проверок можно обращаться к полям ответного ISO-сообщения. Например, для того чтобы проверить код ответа, настройте следующую проверку (см. рисунок 51):

Проверка

Детали

Наименование* Код ответа Источник* Входящее сообщение

Выражение* {\$DE39} Формат

Тип данных* Строка Проверка* Равно

Ожидаемое значение 00 Формат

Сохранить Отмена

Рисунок 51 Создание проверки

5.4.9.1.1 Автоматическое заполнение полей шага данными бинарного дампа

Чтобы заполнить поля секции **Данные шага** данными бинарного дампа:

1. Перейдите на страницу шага по ссылке в колонке **Наименование**.
2. Нажмите кнопку (**Загрузить ISO сообщение**) в верхнем правом углу: появится всплывающее окно **Загрузка ISO**.
3. Скопируйте и вставьте данные бинарного дампа в секцию **Данные** и нажмите **Загрузить**.

4. В случае успешного разбора данных закройте окно с сообщением об успешной операции и обновите страницу шага.

5.4.9.2 Работа с диалектом XML

Для создания XML-сообщения необходимо поместить данные в ячейку **Данные шага**, как показано на рисунке 52.

Главная | Проекты > Демонстрационный проект > Базовая функциональность > Создание клиента

Детали

№: 10 Канал: REST-API Зависит от окружений:

Действие: Выполнить Выполнять всегда: Нет Итерации: 1 Задержка:

Направление: Исходящее Ответ: Обязательный

Наименование: Создание клиента

Описание: Создание клиента по имени, фамилии и номеру документа. Проверяется код ответа. Для дальнейшего использования сохраняется номер клиента из ответного сообщения.

Данные шага

```

1 <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:soap="http://www.solanteq.com/soap-adapter">
2 <soapenv:Header/>
3 <soapenv:Body>
4 <ClientCreateRequest>
5 <FirstName>First</FirstName>
6 <LastName>Last</LastName>
7 <DocumentNumber>12345</DocumentNumber>
8 </ClientCreateRequest>
9 </soapenv:Body>
10 </soapenv:Envelope>

```

Рисунок 52 Создание XML-сообщения

При создании проверок и установке контекстных переменных можно обращаться к полям ответного XML-сообщения, как показано на рисунке 53.

Проверка

Детали

Наименование: Код ответа Источник: Входящее сообщение

Выражение: /Envelope/Body/ClientCreateResponse/body/response/code Формат:

Тип данных: Строка Проверка: Равно

Ожидаемое значение: SLR-0001 Формат:

Сохранить Отмена

Рисунок 53 Создание проверки

При заполнении поля **Выражение** используйте синтаксис языка XPath.

Подробнее о синтаксисе языка XPath см. спецификацию <https://www.w3.org/TR/xpath/>.

XPath позволяет выбрать все вхождения элементов независимо от глубины их расположения. Для сохранения и проверок переменных, полученных в XML-сообщениях со вложенной структурой, используйте переменную типа «Таблица».

Например, фрагмент полученного XML-сообщения имеет следующий вид (см. пример 5-6).

Пример 5-6 Фрагмент полученного XML-сообщения

```

<object>
  <property name="someProperty">123</property>
  <complexProperty>
    <property name="someOtherProperty">345</property>
  </complexProperty>
</object>

```

```

    </complexProperty>
    ...
</object>

```

Чтобы сохранить переменную со значениями всех элементов `property`, создайте переменную типа «Таблица» и укажите в поле **Выражение** путь `/object//property`. В этом случае сохранится переменная со значениями 123, 345.

Чтобы сохранить в переменной значение XML-атрибута `someOtherProperty`, создайте другую переменную типа «Таблица» и укажите в поле **Выражение** путь `//property [@name='someOtherProperty']`. В этом случае сохранится переменная со значением 345.

5.4.9.3 Работа с диалектом JDBC

Для создания запроса к базе данных необходимо заполнить следующую информацию (см. рисунок 54):

Детали

Номер шага* 70 Канал DataBase Действие* Выполнить шаг

Направление* Исходящее Ответ* Обязательный Задержка

Наименование* Проверка остатка по контракту прямым запросом

Описание Проверка остатка по контракту прямым запросом в базу данных по номеру контракта. Проверяются кредитный лимит и баланс собственных средств.

Тип запроса* Выборка

Данные шага 1 `select own_balance, credit_limit from contract where contr_number = :contractNumber`

Переменная*	Тип данных	Значение	Направление	Описание
contractNumber	Строка	\$(contractNumber)	IN	

Добавить строку

Максимальный размер выборки

Рисунок 54 Создание запроса к базе данных

- **Тип запроса** – обязательный параметр. Возможные значения:
 - Выборка — запрос данных,
 - Обновление — добавление, изменение и удаление данных,
 - Вызов функции — вызов хранимой процедуры, функции и запуск SQL выражения.
- **Данные шага** — Параметр, передаваемый в составе запроса. Должен начинаться со специального символа «:», например `select first_name, last_name from client where id = :id`
- **Параметры запроса** — инициализация параметров запроса:
 - **Переменная** — имя переменной для подстановки в запрос без специального символа, например, `id`.
 - **Тип данных** — тип данных параметра. Обязательно указывается во всех типах запросов в базу данных. Значение выбирается из доступных вариантов: Строка, Целое число, Вещественное число, Логическое да/нет, Дата, Время, Дата и время.

- **Значение** — значение параметра. В качестве значения параметра можно использовать контекстные переменные (см. 5.4.6 «Контекстные переменные»). Указывается только для входящих переменных.
- **Направление** — значение выбирается из доступных вариантов: IN, OUT, IN OUT, RETURN.

Каждый параметр упоминается в запросе только один раз, поэтому во избежание ошибок при создании запроса для выполнения SQL-кода рекомендуется заводить дополнительную переменную и присваивать ей значение параметра.

Для запроса типа «Вызов функции» важен порядок параметров запроса (см. раздел 5.4.9.3.3 «Работа с типом запроса «Вызов функции»»).

- **Максимальный размер выборки** — применяется только к запросам типа «Выборка». Задаёт ограничение на число строк в ответном сообщении из базы данных. Значение по умолчанию — 100, максимум — 1000.

При создании проверок и установке контекстных переменных можно обращаться к полям ответного сообщения из базы данных.

5.4.9.3.1 Работа с типом запроса «Выборка»

Тип запроса «Выборка» используется для получения информации из базы данных. Для запроса данного типа применимы только входящие параметры (см. рисунок 54). Данные, полученные из базы, могут быть сохранены в контекстные переменные или использованы в проверке.

Рисунок 55 Установка контекстных переменных запроса типа «Выборка»

5.4.9.3.2 Работа с типом запроса «Обновление»

Тип запроса «Обновление» используется для добавления, обновления и удаления информации базы данных. Для запроса данного типа применимы только входящие параметры. Пример запроса на добавление информации в базу данных приведен на рисунке 56.

Детали

Номер шага* Канал Действие*

Направление* Ответ* Задержка

Наименование*

Описание

Тип запроса*

Данные шага **1** `insert into client (first_name, last_name) (:first, :last)`

Переменная*	Тип данных	Значение	Направление	Описание
first	Строка	John	IN	✕
last	Строка	Smith	IN	✕

Добавить строку

Рисунок 56 Запрос на добавление информации в базу данных

Пример запроса на изменение информации базы данных приведен на рисунке 57.

Детали

Номер шага* Канал Действие*

Направление* Ответ* Задержка

Наименование*

Описание

Тип запроса*

Данные шага **1** `update client set first_name=:first where id=:id`

Переменная	Тип данных	Значение	Направление	Описание
first	Строка	\$(FirstName)	IN	
id	Целое число	10	IN	

Рисунок 57 Запрос на изменение информации базы данных

Пример запроса на удаление информации из базы данных приведен на рисунке 58.

Детали

Номер шага* 100 Канал DataBase Действие* Выполнить шаг

Направление* Исходящее Ответ* Опциональный Задержка

Наименование* Delete

Описание
Удаление данных с подстановкой переменной

Тип запроса* Обновление

Данные шага
1 delete from client | where id =:id

Переменная	Тип данных	Значение	Направление	Описание
id	Целое число	10	IN	

Рисунок 58 Запрос на удаление информации из базы данных

5.4.9.3.3 Работа с типом запроса «Вызов функции»

Тип запроса «Вызов функции» используется для вызова хранимой процедуры базы данных, функции и запуска SQL выражения. Для данного типа запроса применимы все типы параметров.

В поле **Данные шага** укажите полное название процедуры. Названия параметров должны совпадать с названиями переменных для данной процедуры.

Параметры необходимо указать в том порядке, в котором они идут в объявлении процедуры.

Детали

Номер шага* 110 Канал DataBase Действие* Выполнить шаг

Направление* Исходящее Ответ* Опциональный Задержка

Наименование* Procedure

Описание
Вызов процедуры из пакета Solar с сигнатурой
PROCEDURE NEW_CLIENT(
inValue in VARCHAR(255)

Тип запроса* Вызов функции

Данные шага
1 solar.NEW_CLIENT

Переменная	Тип данных	Значение	Направление	Описание
inValue	Строка	inValue	IN	

Рисунок 59 Запрос на запуск процедуры

Данные, полученные из базы, могут быть сохранены в контекстные переменные или использованы в проверке.

Рисунок 60 Установка контекстных переменных запроса типа «Вызов функции»

Пример запроса для запуска функции приведен на рисунке 61. В поле **Данные шага** укажите полное название функции. Названия параметров должны совпадать с названиями переменных функции.

В поле **Параметры запроса** сначала укажите returns, далее — все параметры в том порядке, как они идут в объявлении функции.

Переменная	Значение	Тип данных	Направление	Описание
returns		Строка	OUT	
inValue		Целое число	IN	

Рисунок 61 Запрос на запуск хранимой функции

Пример запроса для запуска анонимного SQL-кода приведен на рисунке 62.

В поле **Данные шага** необходимо указать SQL-код, который должен быть выполнен.

Детали

№ Канал Зависит от окружений

Действие Выполнять всегда Итерации Задержка

Направление Ответ

Наименование

Описание

Тип запроса

Данные шага

1	<code>select own_balance, credit_limit from contract where contr_number = :contractNumber</code>
---	--

Параметры запроса

Переменная	Значение	Тип данных	Направление	Описание
contractNumber	\$(contractNumber)	Строка	IN	

Максимальный размер выборки

Рисунок 62 Запрос на запуск SQL-кода

5.4.9.4 Работа с диалектом HTTP

Для создания HTTP-сообщения заполните форму **Данные шага**.

- Поместите данные тела сообщения в секцию **Сообщение** в формате, поддерживаемом данным каналом.
- (Для исходящего шага) в поле **Url** введите значение, которое используется для подстановки в шаблон Url, заданный в свойствах канала (см. раздел 4.1.4).
- (Для исходящего шага) в списке **Метод** выберите метод HTTP: GET, POST, PUT, DELETE, PATCH, HEAD, OPTIONS или TRACE. Если метод не указан, то он берется из конфигурации адаптера.
- (Для исходящего шага) используйте флажок **Разрешить ошибки клиента и сервера**, чтобы игнорировать ошибки в ответном сообщении:
 - флажок установлен — независимо от кода ответа HTTP в ответном сообщении, включая ошибки 4xx и 5xx, шаг завершается успешно, то есть без ошибок выполнения;
 - флажок не отмечен — статусы HTTP 4xx и 5xx в ответном сообщении приводят к ошибке выполнения шага.
- (Для входящего шага) в списке **Статус** выберите код ответа, который следует передавать в ответном сообщении. По умолчанию используется 200 (OK).
- В секции **Заголовки** добавьте нужное количество строк и введите заголовки HTTP.

При создании проверок для обращения к элементам HTTP-сообщения используйте следующие префиксы:

- #url — входящий адрес URL,
- #headers — заголовки HTTP; токен проверяется как #headers/Authorization;
- #body — тело сообщения,
- #method — метод HTTP,
- #status — код ответа.

Сообщение в оригинальном формате преобразуется в XML. На форме с результатами выполнения шага для исходящих и входящих сообщений отображается как сообщение в оригинальном формате, так и его XML-представление.

5.4.9.5 Работа с диалектом gRPC

Для создания gRPC-сообщения заполните форму **Данные шага**.

- В поле **Имя метода** введите имя метода в формате `<имя_пакета>.<имя_сервиса>/<имя_метода>`, где `<имя_пакета>` задано в .proto файле.
- В секции **Сообщение** введите сообщение в формате JSON.
- В секции **Метаданные** добавьте нужное количество строк и введите метаданные в виде пар название/значение.

Пример заполненной формы **Данные шага** приведен на рисунке 63.

The screenshot shows a web form titled "Данные шага". At the top, there is a field "Имя метода" containing the text "TransactionServer.DispatcherRPC/ProtoTransaction". Below this is a section "Сообщение" with a text area containing a JSON object. The JSON object has a "Header" field with "AppID", "AppVer", "User", and "FirmwareVer" properties, and a "Request" field with "SourceID", "MethodName", "TransactionType", "Amount", "Currency", "StartTime", "Details", "OriginalTransactionType", "TipAmount", and "EposSourceID" properties. At the bottom of the form is a "Метаданные" section with a table:

Наименование	Значение	И1
tid	88880101	

Рисунок 63 Форма «Данные шага» с данными gRPC-сообщения

Сообщение в оригинальном формате преобразуется в XML. На форме с результатами выполнения шага для исходящих и входящих сообщений отображается как сообщение в оригинальном формате, так и его XML-представление.

5.4.9.6 Работа с диалектом RabbitMQ

Для создания сообщения, отправляемого в очередь RabbitMQ, заполните форму **Данные шага**.

- В секции **Сообщение** введите сообщение в формате JSON.
- В секции **Заголовки** добавьте нужное количество строк и введите заголовки в виде пар наименование-значение.

Заголовки `request`, `group` используются для маршрутизации сообщений. Адаптер RabbitMQ передает эти заголовки вместе с сообщением в роутер системы SOLAR, который использует их для определения получателя.

Например, для отправки запроса веб-сервиса в приложение Back-Office надо указать наименование: `request` и значение: `bo-api.<service name>.execute`, т.е. наименование службы, которая обработает запрос.

Если указать group и значение myGroup, то получатель сообщения в файле routing.json должен создать правило для группы myGroup, источника rabbitAdapter и указать назначение (destination), по которому роутер определит маршрут.

Пример заполненной формы **Данные шага** приведен на рисунке 64.

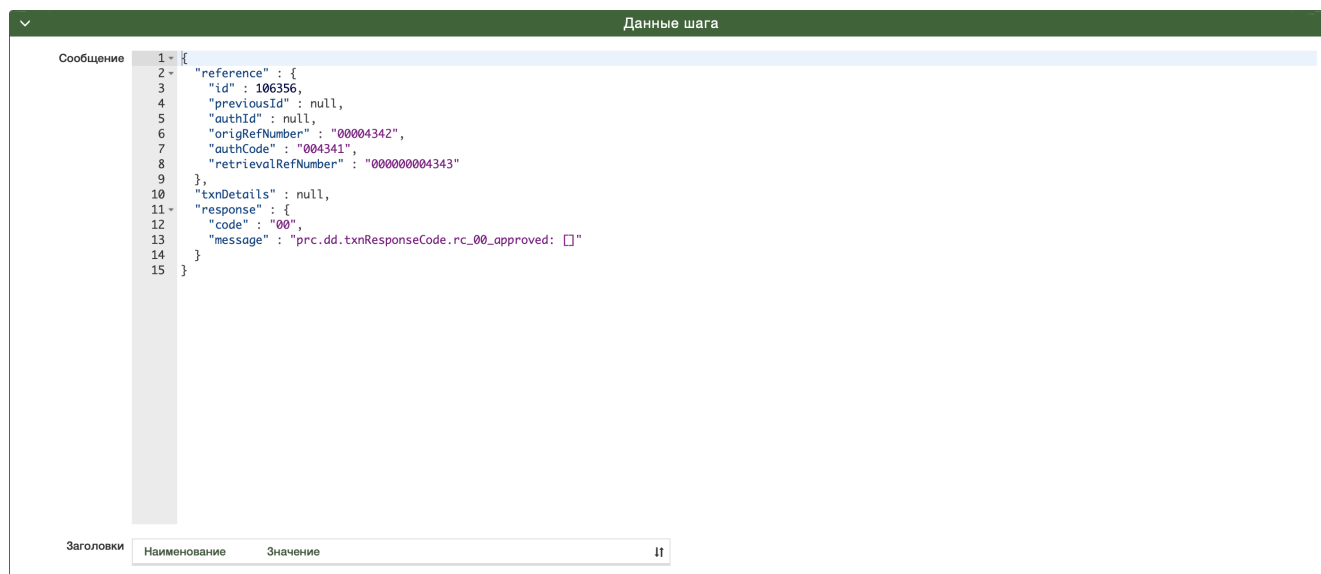


Рисунок 64 Форма «Данные шага» с данными RabbitMQ-сообщения

Сообщение в оригинальном формате преобразуется в XML. На форме с результатами выполнения шага для исходящих и входящих сообщений отображается как сообщение в оригинальном формате, так и его XML-представление.

Для обращения к заголовкам при создании проверок и установке переменных используйте префикс #headers. Например, для доступа к заголовку request используйте выражение #headers.request.

5.4.9.7 Работа с диалектом «Сообщение операционной системе»

Диалект «Сообщение операционной системе» позволяет создавать шаги, на которых выполняются команды операционной системы.

Для создания сообщения, отправляемого операционной системе, заполните форму **Данные шага**.

- В поле **Команда** введите путь к выполняемому скрипту относительно директории SOLAR_DATA_DIR.
 - В директории SOLAR_DATA_DIR рекомендуется создать вложенную директорию и расположить скрипт в ней.
 - Предоставить право доступа к указанной вложенной директории текущему пользователю, запускающему приложение Testing Toolkit.
- В поле **Аргументы** введите аргументы команды. Аргументы, заданные на шаге, передаются в операционную систему вместе с переменными окружения и аргументами команды, заданными на форме канала для конкретного окружения.
- В поле **Сообщение** введите текст, который выводится в стандартный поток ввода stdin.

Для обращения к элементам сообщений при создании проверок и установке переменных используйте следующие префиксы:

- #pid —id процесса,
- #exitCode — код выхода,
- #command — команда,
- #inputData/text/row/data — элемент данных исходящего сообщения, которые пишутся в stdin;
- #outputData/text/row/data — элемент данных входящего сообщения, которые читаются из stdout;
- #errorData/text/row/data — элемент данных об ошибке, которые читаются из потока stderr.

На форме результатов шага отображается идентификатор процесса, код выхода и аргументы. Для входящего шага отображается преобразованный в XML вывод из stdout, а для исходящего — вывод в stdin.

5.4.9.8 Работа с диалектом JMS

Для отправки сообщения в очередь JMS и получения сообщений из очереди необходимо создать отдельные каналы с соответствующим типом подключения. Если используются разные очереди, то для каждой очереди требуются отдельные каналы.

Для корректной работы диалекта JMS убедитесь, что параметр `payloadAsJmsMessage` в конфигурации адаптера имеет значение `true`. Тип сообщения в `Payload` в фильтрах для данного диалекта — `JmsMessage`.

Для создания сообщения, отправляемого в очередь JMS или получаемого из очереди, заполните форму **Данные шага**.

- В поле **Сообщение** введите тело сообщения в соответствующем формате (XML или JSON).

Пример заполненной формы **Данные шага** приведен на рисунке 65.

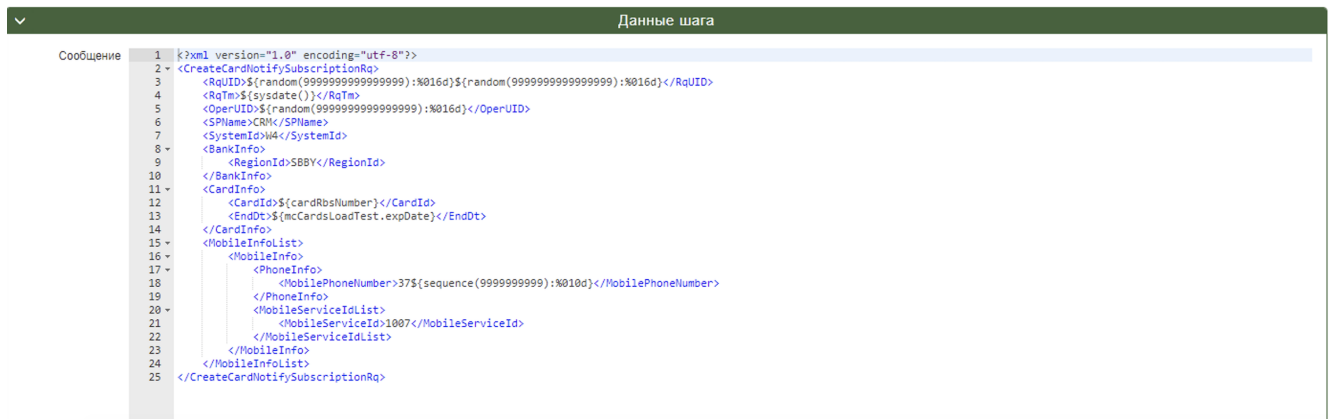


Рисунок 65 Форма «Данные шага» с данными JMS-сообщения

Для обращения к элементам сообщений при создании проверок и установке переменных используйте синтаксис языка XPath.

5.4.10 Использование шаблона действия

На шаге вызова шаблона выполняется следующая последовательность действий:

1. Установка параметров шаблона.
2. Выполнение шаблона действия (включая установку контекстных переменных и проверки, которые в нем определены).

3. Установка контекстных переменных, заданных на шаге вызова шаблона (могут быть использованы переменные из контекста шаблона).
4. Выполнение проверок, заданных на шаге вызова шаблона (могут быть использованы переменные из контекста шаблона).

Чтобы создать шаг, вызывающий шаблон действия, заполните поле **Шаблон действия** и задайте **Выражение** для всех переменных, используемых в шаблоне (например, как на рисунке 66).

Главная > Проекты > Демонстрационный проект > Базовая функциональность > Успешная покупка

Детали

№* Шаблон действия Зависит от окружений

Действие* Выполнять всегда* Итерации Задержка

Наименование*

Описание

Входящие параметры шаблона

Переменная	Выражение	Тип данных	Описание	
pan	\${cardNumber}	Строка	PAN	<input type="button" value="x"/>
cardExp	\${expiryDate}	Строка	Срок действия карты	<input type="button" value="x"/>

Исходящие параметры шаблона

Переменная	Возвращаемый	Тип данных	Описание	Локал. имя	

Рисунок 66 Создание шага, вызывающего шаблон действия

5.5 Планы выполнения

План выполнения — это набор тестовых сценариев, выбранных в соответствии с заданными условиями. В любом условии задается один из следующих объектов: проект, функциональная область, приоритет сценария, система или модуль. Кроме того, можно указать набор параметров, с которым следует запускать выбранные сценарии. Допустим, к проекту привязан набор параметров, строки которого последовательно выбираются при запуске проекта. А в плане выполнения требуется запустить сценарии проекта не со всеми строками набора, а только с конкретной, выбранной по некоторому правилу. С этой целью надо переопределить наборы параметров на уровне плана (см. раздел 5.5.2).

Список планов выполнения доступен через меню **Конфигурация → Планы выполнения**.

Под списком планов выполнения расположена форма **Очередь выполнения**, в которой отображаются запущенные планы и их текущий статус.

Состав данных таблицы **Планы выполнения** описан в таблице 5-6.

Таблица 5-6 Состав данных таблицы «Планы выполнения»

Поле	Описание
Код	Код плана выполнения, который может быть использован внешней системой при управлении SOLAR Testing Toolkit через веб-сервис. Код должен быть уникальным в рамках приложения
Наименование	Название плана выполнения, отображаемое в формах
Блокировка сценариев	Атрибут, который указывает на необходимость выполнения сценариев в режиме блокировки. Возможные значения: <ul style="list-style-type: none"> • Полная — блокируется окружение целиком, • Выборочная — блокируются системы, которые используются каналами, тестируемыми на шагах сценария, • Не задано — блокировка не задана на уровне плана, но может быть задана отдельно для сценария
Описание	Описание назначения плана выполнения

5.5.1 Создание плана выполнения

Чтобы создать план выполнения:

1. Добавьте запись в форму **Планы выполнения** и укажите данные, описанные в таблице 5-6.
2. В форме **Настройки прав редактирования** можно выбрать группу пользователей, чтобы разрешить редактирование плана только пользователям выбранной группы.
3. В списках **Группы пользователей** и **Пользователи** формы **Нотификации** настройте получателей сообщения с отчетом о результатах выполнения плана:
 - а. Нажмите кнопку **Добавить строку** и выберите группу или отдельного пользователя.
4. Нажмите **Сохранить**.
5. Выберите запись по ссылке в колонке **Наименование**.
6. В форме **Условия** добавьте одно или несколько условий выбора сценариев и укажите следующие данные:
 - **№** — порядковый номер,
 - **Наименование** — наименование условия,
 - **Объект** — выберите один или несколько объектов списка и укажите их значения.
 - Проект,
 - Функциональная область,
 - Приоритет,
 - Система,
 - Модуль.
7. (Опционально) В форме **Наборы параметров** выберите наборы параметров, которые следует использовать при запуске плана (см. раздел 5.5.2 «Переопределение набора параметров»).
8. Нажмите **Сохранить**.
9. Перезагрузите страницу, чтобы применить условия.

Тестовые сценарии, которые соответствуют заданным условиям, отображаются в форме **Выбранные сценарии**.

5.5.2 Переопределение набора параметров

Форма **Наборы параметров** на странице плана выполнения предназначена для переопределения набора параметров, которые следует использовать при запуске плана выполнения.

Чтобы переопределить набор параметров:

1. Перейдите на страницу плана по ссылке в колонке **Наименование**.
2. Нажмите кнопку **Создать** в форме **Наборы параметров**. Появится окно **Привязанный набор параметров**, изображенное на рисунке 67.

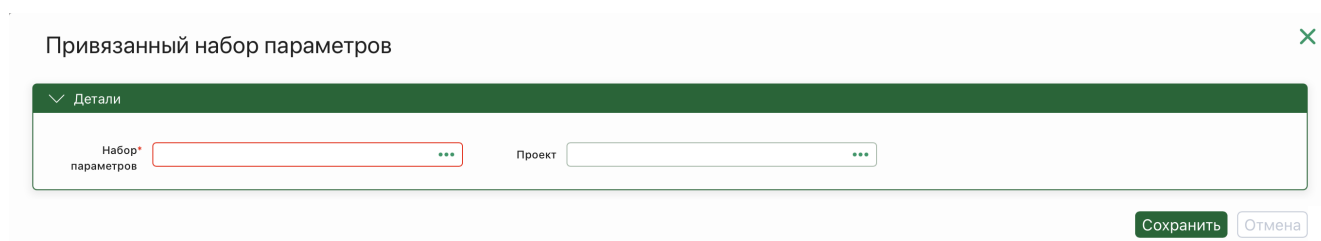


Рисунок 67 Привязанный набор параметров

3. Справа от поля **Набор параметров** нажмите кнопку **Искать**, выберите набор параметров и нажмите **Выбрать**.
4. Справа от поля **Проект** нажмите кнопку **Искать**, выберите проект, в котором используется выбранный набор параметров, и нажмите **Выбрать**.
5. Нажмите **Сохранить**.
6. Перейдите по ссылке в колонке **Правила выбора строк**.
7. Нажмите кнопку **Создать** в форме **Правила выбора строк набора параметров**. Появится окно **Правило выбора строк набора параметров**, изображенное на рисунке 68.

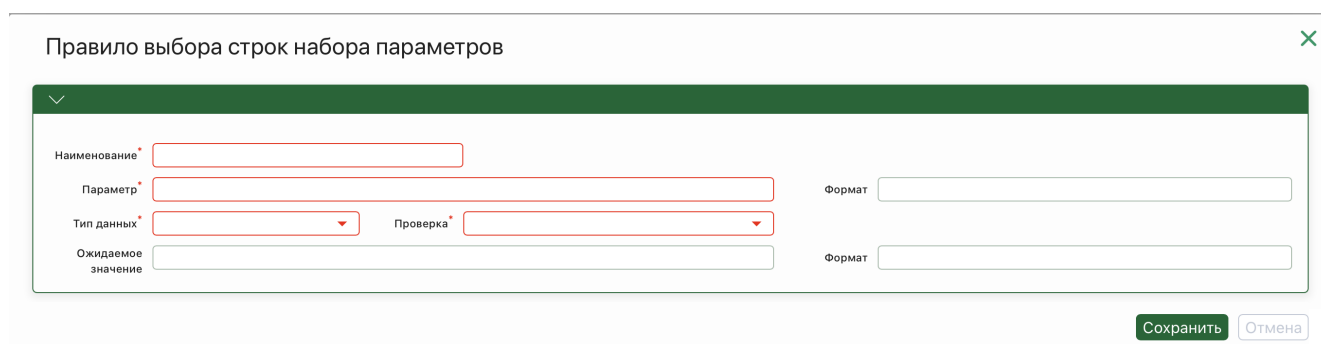


Рисунок 68 Правило выбора строки набора параметров

8. Укажите следующие параметры:
 - **Наименование** — наименование правила,
 - **Параметр** — имя параметра,
 - **Формат** — формат данных, который следует применить к параметру (подробнее о синтаксисе данного поля см. в разделе 5.7 «Форматирование»),
 - **Тип данных** — тип данных параметра,
 - **Проверка** — тип сравнения:
 - Равно,
 - Не равно,

- Меньше,
- Больше,
- Меньше или равно,
- Больше или равно,
- **Ожидаемое значение** — значение, с которым сравнивается значение параметра.
- **Формат** (ожидаемого значения) — формат данных, который следует применить к ожидаемому значению (подробнее о синтаксисе данного поля см. в разделе 5.7 «Форматирование»).

9. Нажмите **Сохранить**.

Если набор параметров переопределен, то для данного плана выполнения выбирается строка набора параметров, для которой выполнены правила, заданные в форме **Правила выбора строк набора параметров**.

5.5.3 Запуск плана выполнения

Чтобы запустить план выполнения:

1. Откройте форму **Планы выполнения** и нажмите кнопку ► (Запустить план) в правой части строки выбранного плана или перейдите на страницу плана и нажмите аналогичную кнопку в правом верхнем углу страницы.
2. В форме **Настройки выполнения** установите необходимые параметры:
 - a. Измените описание запуска, если это необходимо.
 - b. Установите флажок **Прерываться на ошибке**, если в случае возникновения ошибки требуется прерывать выполнение плана.
 - c. В поле **Окружение** выберите окружение, если оно еще не выбрано.
 - d. Установите флажок **Отправить отчет на почту**, чтобы по окончании выполнения плана отправить стандартный отчет о выполнении на адрес, заданный в профиле пользователя.

Информация о пользователях, запускавших план выполнения, отображается на странице плана выполнения в форме **Запуска плана** в поле **Пользователь**.

Сценарии плана, к которым привязаны наборы параметров без указания конкретной строки, запускаются несколько раз с разными строками параметров, выбираемыми в соответствии с заданным методом (случайный или последовательный).

5.6 Требования

Список требований, тестируемых в рамках проекта, доступен на странице проекта. Требования можно добавить вручную (см. раздел 5.6.1) или импортировать из CSV-файла (см. раздел 5.6.2).

Чтобы получить информацию о матрице покрытия тестами (количестве тестовых сценариев, разработанных для проверки каждого требования), сформируйте отчет о покрытии, как описано в разделе 5.6.3.

▼
Детали

Код

Наименование

Блокировка

Описание

>
Наборы параметров

>
Q² Тестовые сценарии

>
Версии

>
Запуски проекта

▼
↻ +
Требования

Код	↕ 1 Наименование	↕ 2 Описание	
DB	Отправка запроса в БД	Базовая функциональность запросов в БД	🔗 ✕
INT	Интеграционное взаимодействие	Общее интеграционное взаимодействие	🔗 ✕
ISO	Работа с ISO-сообщениями	Базовая функциональность отправки/получения ISO сообщений	🔗 ✕
XML	Работа с Web-Services	Базовая функциональность работы с Web-Services	🔗 ✕

Рисунок 69 Требования проекта

Таблица **Требования** содержит поля, описанные в таблице 5-7.

Таблица 5-7 Состав данных таблицы «Требования»

Поле	Описание
Код	Код требования, который может быть использован внешней системой при управлении SOLAR Testing Toolkit через веб-сервис. Код должен быть уникальным в рамках проекта
Наименование	Название требования, отображаемое в формах
Описание	Описание требования
Ответственный	Пользователь, ответственный за требование

Чтобы увидеть список сценариев, проверяющих требование, перейдите по ссылке в колонке **Наименование**. Список сценариев из разных проектов отображается в форме **Тестовые сценарии**, как показано на рисунке 70.

☰ Конфигурация Demo 👤 Администратор

Главная ; Проекты > Демонстрационный проект > Интеграционное взаимодействие

Детали

Код* INT Наименование* Интеграционное взаимодействие

Описание
Общее интеграционное взаимодействие

Ответственный* Администратор

Тестовые сценарии

Код	Наименование	Описание
BASIC	Базовая функциональность	Создание основных объектов: - Клиент - Контракт - Карта - Проведение успешной операции - Проведение операции с отказом (нехватка средств) [раскрыть...]

Рисунок 70 Тестовые сценарии, проверяющие требование

5.6.1 Создание требования

Чтобы создать требование:

1. Добавьте запись в форму **Требования** и укажите данные, описанные в таблице 5-7.
2. Нажмите **Сохранить**.

5.6.2 Импорт требований

Требования можно импортировать из CSV-файла. В первой строке файла указывается версия схемы. Каждая следующая строка файла соответствует одному требованию. В качестве разделителя используется запятая.


Для подготовки файла можно использовать Excel и выбрать при сохранении формат CSV. Вариант CSV-файла приведен в примере 5-7.

Пример 5-7 Вид CSV-файла для импорта требований

```
1.0.0,,
code1,Name1,description1
code2,name2,description2
code3,name3,description3
code4,name4,description4
code5,name5,description5
```

В результате импорта указанного файла в проекте будут созданы пять требований с указанными в файле кодами, наименованиями и описаниями.

Чтобы импортировать требования в проект:

1. Откройте страницу проекта и в правом верхнем углу нажмите кнопку  (Импортировать требования).
2. В окне **Импортирование требований** нажмите кнопку **Выбрать файл** и выберите в файловой системе подготовленный файл.
3. Установите флажок **Создать сценарии**, чтобы для каждого требования автоматически создать сценарий с таким же кодом, наименованием и описанием.
4. Нажмите **Импортировать**.

5.6.3 Формирование отчета о покрытии требований

Отчет о покрытии требований тестовыми сценариями содержит список требований и количество тестовых сценариев, разработанных для каждого требования.

Чтобы сформировать отчет о покрытии:

1. Выберите пункт меню **Отчеты** → **Отчеты**
2. Найдите в форме **Типы отчетов** отчет **Report on the Requirements Coverage in the Project** и нажмите кнопку ► (Сформировать отчет).
3. В окне **Отчет о покрытии требований проекта** в поле **Проект** выберите проект и нажмите кнопку **Сформировать отчет**.
4. В случае успешного формирования отчета файл отчета доступен для просмотра по ссылке в колонке **Файл** формы **Сформированные отчеты**.

5.7 Форматирование

5.7.1 Форматирование даты и времени

По умолчанию дата и время отображаются в формате уууу-ММ-ддТНН:мм. Вы можете изменить формат представления даты и времени в формах, используя обозначения в таблице 5-8.

Таблица 5-8 Форматирование даты и времени

Параметр	Описание
у	год
М	месяц
д	день месяца
Н	часы в 24-часовом формате
h	часы в 12-часовом формате
m	минуты
s	секунды
a	значение AM/PM

Примеры форматов:

- dd.ММ.уууу – только дата (20.04.1889),
- НН-мм-сс – только время (14-08-00),
- hh.mm.ss а – только время в 12-часовом формате (02.08.00 PM).

Подробнее о форматировании даты и времени см. на странице

<https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/time/format/DateTimeFormatter.html>

5.7.2 Форматирование чисел

По умолчанию приложение выводит минимальное количество знаков после разделителя, необходимое для отображения числа. Разделителем выступает «.». Вы можете изменить формат представления чисел, используя обозначения в таблице 5-9.

Таблица 5-9 Форматирование чисел

Параметр	Описание	Пример
<code>\${var:%d}</code>	Целое число	«5»
<code>\${var:+%d}</code>	Целое число с указанием знака	«+5»
<code>\${var:%3d}</code>	Целое число шириной 3 символа	« 5»
<code>\${var:%03d}</code>	Целое число шириной 3 символа с заполнением пустых мест нулями	«005»
<code>\${var:%f}</code>	Вещественное число с количеством знаков после разделителя по умолчанию – 6	«3.500000»
<code>\${var:%.1f}</code>	Вещественное число с одним знаком после разделителя	«3.5»
<code>\${var:%5.1f}</code>	Вещественное число шириной 5 символов с одним знаком после разделителя	« 3.5»

Подробнее о форматировании чисел см. на странице

<https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/util/Formatter.html>

5.8 Запуск проекта

Чтобы запустить проект:

1. Откройте форму **Проекты** и нажмите кнопку ► (Запустить проект) в правой части строки выбранного проекта или перейдите на страницу проекта и нажмите аналогичную кнопку в правом верхнем углу страницы.
2. В форме **Настройки выполнения** установите необходимые параметры:
 - a. Измените описание запуска, если это необходимо.
 - b. Установите флажок **Прерываться на ошибке**, если в случае возникновения ошибки требуется прерывать выполнение проекта.
 - c. В поле **Окружение** выберите окружение, если оно еще не выбрано.
 - d. В поле **Приоритеты** добавьте необходимое количество строк и укажите приоритеты, чтобы выбрать для запуска только сценарии с заданными приоритетами.
 - e. Установите флажок **Отправить отчет на почту**, чтобы по окончании выполнения проекта отправить стандартный отчет о выполнении на адрес, заданный в профиле пользователя.

Информация о пользователях, запускавших план выполнения, отображается на странице проекта в форме **Запуски проекта** в поле **Пользователь**.

5.9 Отчеты о выполнении

Приложение позволяет формировать стандартные отчеты в формате PDF о результатах запуска отдельного сценария, проекта или плана выполнения.

Отчет можно отправить на почту, что настраивается в форме запуска сценария, проекта или плана (см. разделы 6.1 «Запуск сценария» и 5.5.3 «Запуск плана выполнения»).

Отчет о результатах любого выполненного сценария, проекта или плана выполнения можно сформировать, как описано в разделе 5.9.1 «Формирование отчета о выполнении».

5.9.1 Формирование отчета о выполнении

Чтобы сформировать отчет о результатах запуска сценария, проекта или плана выполнения:


1. Перейдите на страницу с результатами запуска сценария, проекта или плана выполнения.
2. В форме **Действия** нажмите кнопку **Отчет о выполнении**.
3. В окне **Результат выполнения** нажмите **Заккрыть**.

Отчет сохраняется в файле с именем scenario_result.pdf (project_result.pdf, plan_result.pdf) в папке, выбранной для загрузки файлов, а также отображается на экране.

5.10 Просмотр истории изменений

Приложение сохраняет историю изменений основных объектов: сценариев, шагов, шаблонов и наборов параметров. Для каждого изменения сохраняется имя пользователя, который внес изменение, дата и состояние объекта после внесения изменений. История изменений доступна для просмотра на странице объекта. Если в процессе работы окажется, что изменения были некорректны и привели к ошибкам, существует возможность восстановить состояние объекта до внесения нежелательных изменений.

Чтобы просмотреть историю изменений объекта:

1. Откройте страницу объекта (сценария, шаблона, шага или набора параметров).
2. Для проекта, сценария, шаблона или шага перейдите на вкладку **История изменений** (см. рисунок 71).
 - a. Для сценария доступны формы **История изменений сценария** и **История изменений шагов**.
 - b. Для шаблона доступны формы **История изменений шаблона** и **История изменений шагов**.
 - c. Для шага доступны формы **История изменений шага**, **История изменений установок переменных**, **История изменений проверок** и **История изменений правил выбора входящих сообщений**. Для индикации удаленных шагов используется красный фон.
3. В форме **История изменений <сценария, шаблона ...>** перейдите на страницу с деталями изменения по ссылке в колонке **Наименование**.
4. Чтобы восстановить выбранную ревизию, нажмите кнопку  в правом верхнем углу.
5. Проверьте данные в форме **Детали ревизии** и нажмите **Восстановить**.

История изменений шаблона							
Код	Наименование	Описание	Изменен	Пользователь			
История изменений тестовых шагов							
№	Наименование	Описание	Окружения	Действие	Изменен	Пользователь	ИИ
10	Select max(id) as from acc_account_entry			Выполнить	2019-06-20 15:17:28	sk	
20	acc.accountEntryJobService.prepareAll		Reference Processing (Loyalty): Release, Reference Processing (Loyalty): Test	Выполнить	2020-11-21 23:55:45	Ks	
30	acc.accountEntryJobService.closeEntriesInteval			Выполнить	2020-11-21 23:56:01	Ks	
40	Request entries from acc_account_entry			Выполнить	2019-06-20 15:17:18	sk	

Рисунок 71 Просмотр истории изменений шаблона

6 Работа со сценарием

6.1 Запуск сценария

Существует несколько способов запуска тестового сценария:

1. Запуск сценария по кнопке ► (Запустить сценарий) в правом верхнем углу страницы сценария (см. рисунок 72).

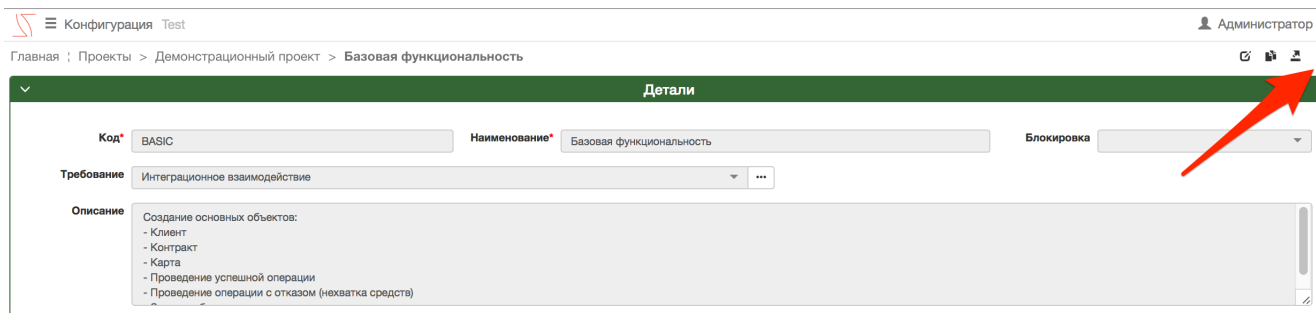


Рисунок 72 Запуск сценария со страницы сценария

2. Запуск сценария со страницы проекта (см. рисунок 73).

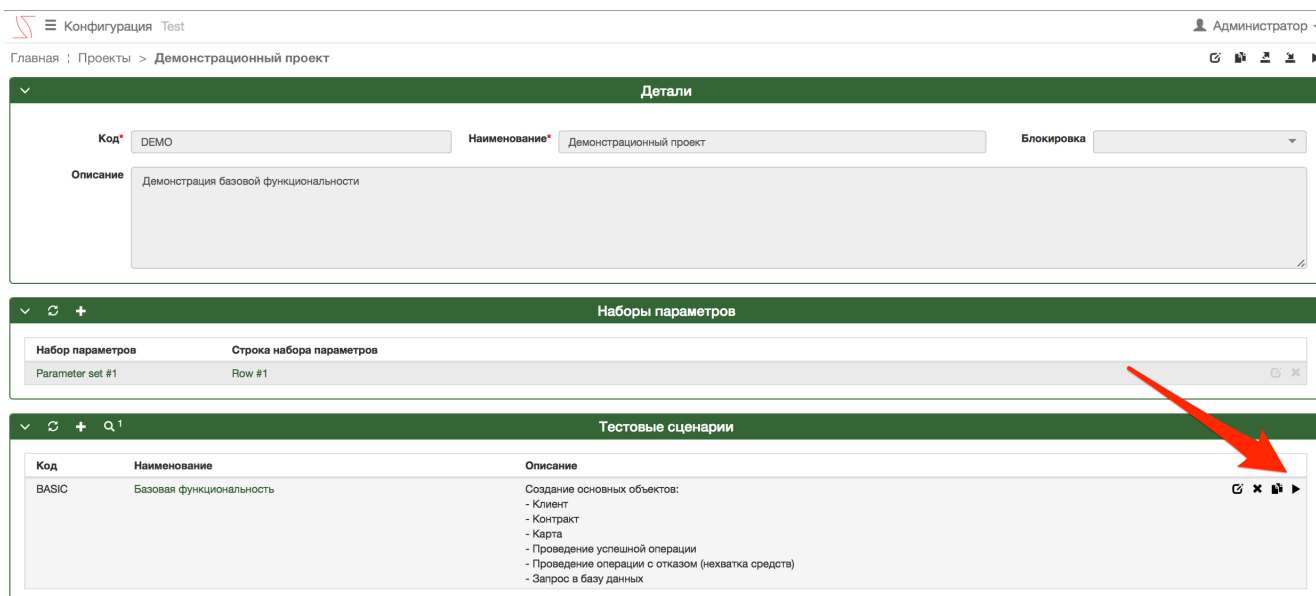


Рисунок 73 Запуск сценария со страницы проекта

3. Пакетный запуск всех сценариев проекта по кнопке ► (Запустить проект) на странице **Детали** выбранного проекта (см. рисунок 74) или в таблице **Проекты**:

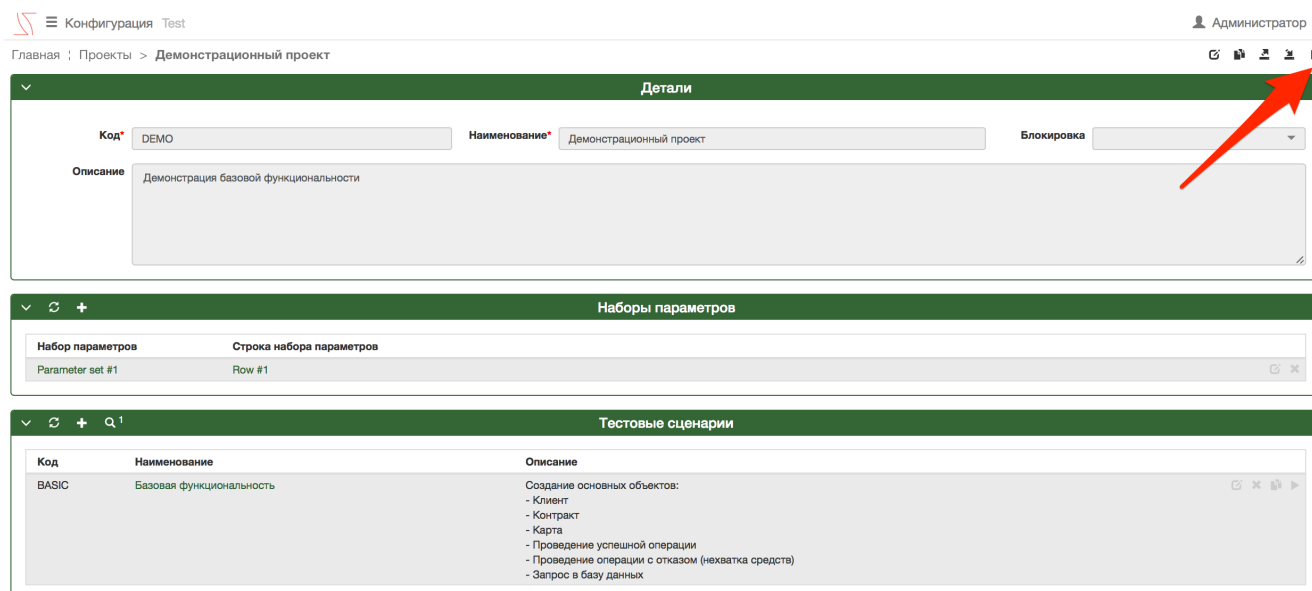


Рисунок 74 Пакетный запуск всех сценариев проекта

При запуске сценария появляется всплывающее окно (см. рисунок 75), в котором вы можете изменить описание запуска, выбрать окружение, установить/снять флажок **Прерываться на ошибке** и установить флажок **Отправить отчет на почту**.

Если в пользовательских настройках задано окружение по умолчанию, оно автоматически предлагается в поле **Окружение**.

Указанное описание отображается в поле **Описание** в результатах запуска (см. рисунки 76, 77 и 78).

Флажок **Прерываться на ошибке** управляет режимом выполнения сценария в случае возникновения ошибки: при установленном флажке выполнение сценария прекращается, при снятом — продолжается.

Если выбрана отправка отчета на почту, то по завершении выполнения сценария на адрес, заданный в профиле пользователя, отправляется стандартный отчет о выполнении.

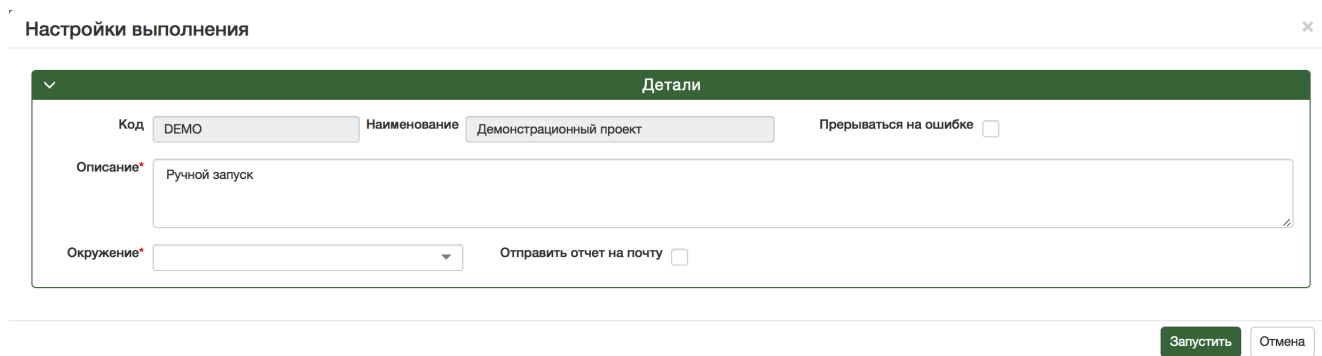


Рисунок 75 Настройки выполнения сценария

Сценарии, запускаемые в пакетном режиме (см. пункт), к которым привязаны наборы параметров без указания конкретной строки, запускаются несколько раз с разными строками параметров, выбираемыми в соответствии с заданным методом (случайный или последовательный).

6.2 Просмотр результатов запуска

Посмотреть результаты запуска можно в трех проекциях:

1. Все запуски выбранного сценария на странице **Тестовый сценарий**, как показано на рисунке 76. Информация о пользователях, запускавших сценарий, отображается в поле **Пользователь**.

Главная ; Проекты > Demo project > Базовая функциональность

Детали

Внешние задачи

Тестовые шаги

Финализация

Запуски сценария

Окружение	Описание	Пользователь	Дата начала	Дата окончания	Статус	
DEFAULT ENVIRONMENT	DEFAULT ENVIRONMENT: Ручной запуск	kranova	2019-03-11 17:30:51	2019-03-11 17:30:51	Ошибка	
DEFAULT ENVIRONMENT	DEFAULT ENVIRONMENT: Ручной запуск	kratov	2018-12-06 13:22:21	2018-12-06 13:22:21	Не пройден	
DEFAULT ENVIRONMENT	DEFAULT ENVIRONMENT: Ручной запуск	kranova	2018-12-04 16:20:07	2018-12-04 16:20:08	Не пройден	
DEFAULT ENVIRONMENT	DEFAULT ENVIRONMENT: Ручной запуск	kranova	2018-12-04 16:01:01	2018-12-04 16:01:03	Не пройден	
DEFAULT ENVIRONMENT	DEFAULT ENVIRONMENT: Ручной запуск	kranova	2018-11-22 01:14:07	2018-11-22 01:14:07	Успешно	
DEFAULT ENVIRONMENT	DEFAULT ENVIRONMENT: Manually started	isadovskaya	2018-10-05 14:14:20	2018-10-05 14:14:20	Успешно	
DEFAULT ENVIRONMENT	DEFAULT ENVIRONMENT: Manually started	kranova	2018-10-05 12:46:52	2018-10-05 12:46:53	Успешно	
DEFAULT ENVIRONMENT	DEFAULT ENVIRONMENT: Manually started	kranova	2018-10-05 12:32:35	2018-10-05 12:32:36	Успешно	
DEFAULT ENVIRONMENT	DEFAULT ENVIRONMENT: Manually started	isadovskaya	2018-10-05 10:46:13	2018-10-05 10:46:14	Успешно	
DEFAULT ENVIRONMENT	DEFAULT ENVIRONMENT: Manually started	vkhovanskiy	2018-10-04 15:40:01	2018-10-04 15:40:02	Успешно	
DEFAULT ENVIRONMENT	DEFAULT ENVIRONMENT: Manually started	vkhovanskiy	2018-10-04 15:38:08	2018-10-04 15:38:10	Успешно	

Рисунок 76 Результаты всех запусков выбранного сценария

2. Запуски всех сценариев в рамках проекта на странице **Запуск проекта**, как показано на рисунке 77. Информация о пользователях, запускавших проект, отображается на странице проекта в форме **Запуски проекта** в поле **Пользователь**.

Главная ; Проекты > Демонстрационный проект > Запуск проекта

Детали запуска проекта

Проект: Демонстрационный проект Дата начала: 09/02/2018 17:49:01 Дата окончания: 09/02/2018 17:49:02

Версия: Релиз №1

Успешно: 1 Неуспешно: 1 Пропущено: 0 Защищенный:

Описание: DEFAULT ENVIRONMENT: Ручной запуск

Выполненные сценарии

Сценарий	Описание	Дата начала	Дата окончания	Статус	
[BASIC_1] Базовая функциональность	DEFAULT ENVIRONMENT: Ручной запуск	2018-02-09 17:49:02	2018-02-09 17:49:02	Успешно	
[BASIC_2] Базовая функциональность - тест завершается ошибкой	DEFAULT ENVIRONMENT: Ручной запуск	2018-02-09 17:49:01	2018-02-09 17:49:02	Не пройден	

Рисунок 77 Результаты запусков всех сценариев проекта

3. Запуски всех сценариев всех проектов в рамках версии системы на странице версии системы, как показано на рисунке 78.

Детали					
Код	DEFAULT_RELEASE	Наименование	DEFAULT RELEASE		
Архивная	<input type="checkbox"/>				
Описание					

Запуски сценариев					
Сценарий	Описание	Дата начала	Дата окончания	Статус	И1
[BASIC_1] Базовая функциональность	DEFAULT ENVIRONMENT: Ручной запуск	2018-02-09 17:49:02	2018-02-09 17:49:02	Успешно	
[BASIC_2] Базовая функциональность - тест завершается ошибкой	DEFAULT ENVIRONMENT: Ручной запуск	2018-02-09 17:49:01	2018-02-09 17:49:02	Не пройден	
[BASIC_1] Базовая функциональность	DEFAULT ENVIRONMENT: Ручной запуск	2018-02-08 15:17:10	2018-02-08 15:17:11	Успешно	
[BASIC_2] Базовая функциональность - тест завершается ошибкой	DEFAULT ENVIRONMENT: Ручной запуск	2018-02-08 12:31:04	2018-02-08 12:31:05	Не пройден	
[BASIC_1] Базовая функциональность	DEFAULT ENVIRONMENT: Ручной запуск	2018-02-06 21:32:28	2018-02-06 21:32:29	Успешно	
[BASIC_2] Базовая функциональность - тест завершается ошибкой	DEFAULT ENVIRONMENT: Ручной запуск	2018-02-06 21:32:27	2018-02-06 21:32:28	Не пройден	

Рисунок 78 Результаты запусков всех сценариев в рамках версии системы

Если в приложении настроена работа с несколькими окружениями, в поле **Описание** результатов запуска тестового сценария отображается название окружения, в котором был запущен данный сценарий (см. пометки на рисунках 76, 77 и 78).

6.3 Анализ результатов запуска

6.3.1 Статус выполнения

Состояние запуска сценария описывается одним из следующих статусов:

- **Ожидает** — сценарий ожидает выполнения. Статус отображается, если для сценария установлен атрибут блокировки, но в данный момент уже выполняется другой сценарий в режиме блокировки (см. таблицу 5-2). После того как завершится сценарий, выполняемый в режиме блокировки, выполняется ожидающий сценарий.
- **Выполняется** — сценарий находится в процессе выполнения. Сценарий без установленного атрибута блокировки переходит в статус «Выполняется» сразу после запуска.
- **Успешно** — запуск сценария успешно завершен.
- **Не пройден** — запуск сценария завершен с ошибкой по причине несоответствия полученных данных ожиданиям. Например, не прошла проверка; данные не соответствуют указанному формату; XML-сообщение приведено в некорректном формате и не может быть разобрано.
- **Ошибка** — запуск сценария завершен с непредвиденной ошибкой. Данная ошибка возникает, когда на одном из шагов запрос не соответствует структуре базы данных. В случае получения данного статуса следует обратиться в службу поддержки;
- **Превышено время ожидания** — ответ от внешней системы не поступил в течение ожидаемого времени.

Окружение	Описание	Пользователь	Дата начала	Дата окончания	Статус
DEFAULT ENVIRONMENT	DEFAULT ENVIRONMENT: Ручной запуск	vkhovanskiy	2019-03-11 12:12:22	2019-03-11 12:12:22	Ошибка
DEFAULT ENVIRONMENT	DEFAULT ENVIRONMENT: Ручной запуск	akopylov	2019-01-15 19:11:02	2019-01-15 19:11:02	Не пройден
DEFAULT ENVIRONMENT	DEFAULT ENVIRONMENT: Ручной запуск	akopylov	2019-01-15 19:10:14	2019-01-15 19:10:15	Не пройден
DEFAULT ENVIRONMENT	DEFAULT ENVIRONMENT: Ручной запуск	akopylov	2019-01-15 19:09:17	2019-01-15 19:09:17	Не пройден
DEFAULT ENVIRONMENT	DEFAULT ENVIRONMENT: Ручной запуск	skochelaeva	2019-01-10 19:53:35	2019-01-10 19:53:35	Не пройден
DEFAULT ENVIRONMENT	DEFAULT ENVIRONMENT: Ручной запуск	kratov	2018-12-06 13:21:44	2018-12-06 13:21:44	Не пройден
DEFAULT ENVIRONMENT	DEFAULT ENVIRONMENT: Ручной запуск	isadovskaya	2018-10-29 16:11:31	2018-10-29 16:11:32	Успешно
DEFAULT ENVIRONMENT	DEFAULT ENVIRONMENT: Ручной запуск	isadovskaya	2018-10-25 18:19:44	2018-10-25 18:19:45	Успешно
DEFAULT ENVIRONMENT	DEFAULT ENVIRONMENT: Ручной запуск	isadovskaya	2018-10-25 18:15:15	2018-10-25 18:15:15	Успешно

Рисунок 79 Статусы запуска сценария

6.3.2 Детали выполнения сценария

Для каждого запуска сценария можно посмотреть более детальную информацию, перейдя по ссылке в поле **Описание** (см. рисунок 80).

Конфигурация Demo Администратор

Главная ; Проекты > Демонстрационный проект > Базовая функциональность > Запуск сценария

Детали запуска сценария

Сценарий: Базовая функциональность Дата начала: 25/04/2017 19:00:52 📅 Дата окончания: 25/04/2017 19:00:55 📅

Релиз: Релиз №1

Описание: Демо: Ручной запуск

Статус: Успешно Защищенный

Результаты выполнения шагов сценария

Шаг: Статус:

№	Шаг	Тип	Статус	Сообщение системы	Дата начала	Дата окончания
10	Создание клиента	Сообщение	Успешно	XML-сообщение успешно отправлено	2017-04-25 19:00:52	2017-04-25 19:00:53
20	Создание контракта	Сообщение	Успешно	XML-сообщение успешно отправлено	2017-04-25 19:00:53	2017-04-25 19:00:53
30	Создание карты	Сообщение	Успешно	XML-сообщение успешно отправлено	2017-04-25 19:00:53	2017-04-25 19:00:53
40	Успешная покупка	Шаблон действия	Успешно	Успешно	2017-04-25 19:00:53	2017-04-25 19:00:54

Рисунок 80 Детали запуска сценария

В разделе **Детали запуска сценария** находится следующая информация:

- **Сценарий** — ссылка на страницу сценария,
- **Дата начала** — системная дата начала выполнения сценария,
- **Дата окончания** — системная дата окончания выполнения сценария,
- **Описание** — описание запуска. Поле зарезервировано для будущих версий, на данный момент всегда принимает значение «Ручной запуск».
- **Окружение** — ссылка на окружение, в котором выполнен запуск;
- **Используемые системы** — ссылка на систему и ее версию, в которой выполнен запуск;
- **Используемые каналы** — список каналов, используемых в сценарии;
- **Используемые наборы параметров** — ссылка на используемые наборы параметров;
- **Защищенный** — флажок, означающий режим защиты результатов запуска от удаления;
- **Статус** — статус запуска сценария.

В таблице **Результаты выполнения шагов сценария** приведена информация по каждому шагу, выполненному в рамках сценария. Шаги, вызывающие шаблон действия, выводятся в следующей последовательности: сначала шаги шаблона действия, затем основной шаг сценария.

Для выбора шагов по названию или с определенным статусом используйте фильтр.

- **№** — порядковый номер шага с учетом вложенности,
- **Шаг** — ссылка на детали конкретного шага (см. 6.3.3 «Детали выполнения шага»),
- **Тип** — тип шага. Возможные значения:
 - Отправка сообщения — шаг, на котором отправляется сообщение в процессинговую систему;
 - Получение сообщения — шаг, на котором принимается сообщение от процессинговой системы;
 - Отправка сообщения (Отложено) — шаг с отложенным выполнением, на котором отправляется сообщение в процессинговую систему;
 - Получение сообщения (Отложено) — шаг с отложенным выполнением, на котором ожидается сообщение от процессинговой системы;
 - Шаблон действия — шаг, вызывающий шаблон действия.
- **Статус** — статус выполнения шага,
- **Сообщение системы** — системное сообщение, например, сообщение об ошибке,
- **Дата начала** — системная дата начала выполнения выбранного шага,
- **Дата окончания** — системная дата окончания выполнения выбранного шага.

В форме **Действия** отображается кнопка **Отчет о выполнении**, которая позволяет сформировать и загрузить отчет о результатах выполнения в формате PDF.

В таблице **Состояние контекста** отображаются значения переменных контекста до и после выполнения сценария (см. рисунок 81).

Главная | Проекты > Демонстрационный проект > Базовая функциональность > Запуск сценария

Детали запуска сценария

Результаты выполнения шагов сценария

Шаг Статус

Действия

Состояние контекста

Переменная	Тип	Значение до	Значение после	Статус	И1
/ps1.b	Целое число	55555	55555	Без изменений	
/ps1.a	Строка	new value	new value	Без изменений	
/ps2.d	Целое число	12345	12345	Без изменений	
/ps1.c	Логическое да/нет	false	false	Без изменений	

Рисунок 81 Детали запуска сценария

6.3.3 Детали выполнения шага

6.3.3.1 Детали

Раздел **Детали** содержит подробную информацию о текущем шаге, как показано на рисунке 82.

Конфигурация Demo Администратор

Главная : Проекты > Демонстрационный проект > Базовая функциональность > Запуск сценария > Операция покупки

Детали

Шаг Операция покупки **Дата начала** 10/04/2017 12:02:25 **Дата окончания** 10/04/2017 12:02:25

Канал MasterCard

Описание Шаблонная операции покупки.
Параметры:
- rap

Тип Сообщение **Статус** Успешно

Сообщение системы ISO-сообщение было успешно отправлено

Исходящее сообщение

Поле	Значение
000 - MTID	0100
002 - Primary Account Number (PAN)	5505508855684149
003 - Processing Code	000000
004 - Amount, Transaction	000000090000

Входящее сообщение

Поле	Значение
000 - MTID	0110
002 - Primary Account Number (PAN)	5505508855684149
003 - Processing Code	000000
004 - Amount, Transaction	000000090000

Рисунок 82 Детали выполнения шага

- **Шаг** — наименование шага,
- **Дата начала** — системная дата начала выполнения шага,
- **Дата окончания** — системная дата окончания выполнения выбранного шага,
- **Канал** (только для типа шага Сообщение) — название канала, в который отправлялось сообщение на данном шаге,
- **Описание** — описание шага,
- **Окружение** — ссылка на окружение, в котором выполнен запуск;
- **Используемые системы** — ссылка на систему и ее версию, в которой выполнен запуск;
- **Используемые каналы** — список каналов, используемых на шаге. Для шага, вызывающем шаблон действия, содержит все каналы, используемые на шагах шаблона;
- **Тип** — тип шага. Возможные значения:
 - Сообщение — шаг, выполняющий действие (например, отправка сообщения в процессинговую систему);
 - Шаблон действия — шаг, вызывающий шаблон действия,
- **Статус** — статус выполнения шага.
- **Сообщение системы** — системное сообщение, например, сообщение об ошибке,
- **Исходящее сообщение** (только для типа шага Сообщение) — сообщение, которое было отправлено в процессинговую систему с подставленными переменными,
- **Входящее сообщение** (только для типа шага Сообщение) — ответное сообщение, полученное из процессинговой системы.

6.3.3.2 Параметры шаблона действия

Таблица **Параметры шаблона действия** содержит информацию о параметрах, переданных шаблону на данном шаге (используется только для шагов, вызывающих шаблон действия).

Параметры шаблона действия				
Параметр	Выражение	Значение	Статус	Сообщение системы
rap	\${cardNumber}	5505500094162623	Успешно	Переменная "\${cardNumber}" была скопирована в переменную "/rap" без изменений
cardExp	\${expiryDate}	0220	Успешно	Переменная "\${expiryDate}" была скопирована в переменную "/cardExp" без изменений

Рисунок 83 Параметры шаблона действия

- **Параметр** – название параметра шаблона действия,
- **Выражение** – выражение, переданное в параметр,
- **Значение** – значение выражения,
- **Статус** – статус обработки параметра,
- **Сообщение системы** – системное сообщение, например, сообщение об ошибке.

6.3.3.3 Переменные контекста шаблона действия

Таблица **Переменные контекста шаблона действия** содержит информацию о переменных, установленных в контекст внутри шаблона действия (актуально только для шагов, вызывающих шаблон действия), как показано на рисунке 84.

Переменные контекста шаблона действия		
Переменная	Тип	Значение
/DE39	Строка	51
/cardExp	Строка	0220
/rap	Строка	5505500094162623

Рисунок 84 Переменные контекста шаблона действия

- **Переменная** – название переменной,
- **Тип** – тип переменной,
- **Значение** – значение переменной после выполнения шага.

6.3.3.4 Установка переменных

Таблица **Установка переменных** содержит информацию о контекстных переменных, установленных на данном шаге, как показано на рисунке 85

Установка переменных					
Наименование	Переменная	До установки	После установки	Статус	Сообщение системы
Номер клиента	clientNumber		CN-3863ed88-0cc	Успешно	Значение выражения "/Envelope/Body/ClientCreateResponse/ClientNumber" во входящем сообщении было успешно присвоено переменной "/clientNumber"

Рисунок 85 Установка контекстных переменных

- **Наименование** – описание переменной,
- **Переменная** – название переменной,
- **Значение до установки** – значение переменной до выполнения шага,
- **Значение после установки** – значение переменной после выполнения шага,
- **Статус** – статус установки переменной,
- **Сообщение системы** – системное сообщение, например, сообщение об ошибке.

6.3.3.5 Проверки

Таблица **Проверки** содержит информацию по проверкам, выполненным на данном шаге. Пример таблицы приведен на рисунке 86.

Проверки						
Описание	Действительное значение	Проверка	Ожидаемое значение	Статус	Сообщение системы	⌵
Код ответа	51	Равно	51	Успешно	"51" совпадает с ожидаемым значением "51"	⌵

Рисунок 86 Проверки

- Описание – описание проверки,
- Действительное значение – значение выражение после выполнения шага,
- Проверка – тип проверки,
- Ожидаемое значение – ожидаемое значение выражения,
- Статус – статус выполнения проверки,
- Сообщение системы – системное сообщение, например, сообщение об ошибке.

6.3.3.6 Состояние контекста

Таблица **Состояние контекста** содержит информацию по всем переменным, находящимся в контексте сценария на момент окончания выполнения данного шаг, как показано на рисунке 87.

Состояние контекста					
Переменная	Тип	Значение до	Значение после	Статус	⌵
/cardNumber	Строка	5505500094162623	5505500094162623	Без изменений	
/clientNumber	Строка	CN-3863ed88-0cc	CN-3863ed88-0cc	Без изменений	
/contractNumber	Строка	ACC-f88f49c9-0cd	ACC-f88f49c9-0cd	Без изменений	
/expiryDate	Строка	0220	0220	Без изменений	
/ps1.a	Строка	new value	new value	Без изменений	
/ps1.b	Целое число	55555	55555	Без изменений	
/ps1.c	Логическое да/нет	false	false	Без изменений	
/ps2.d	Целое число	12345	12345	Без изменений	

Рисунок 87 Состояние контекста

- Переменная – название переменной,
- Тип – тип переменной,
- Значение до шага – значение переменной до выполнения шага,
- Значение после шага – значение переменной после выполнения шага,
- Статус – статус переменной. Возможные значения:
 - Без изменений,
 - Новая,
 - Изменена.

6.4 Удаление устаревших результатов запуска

Результаты запуска сценариев, проектов и планов тестирования записываются в базу данных и отображаются в соответствующих формах приложения. Со временем данные предыдущих периодов становятся не нужны, и их дальнейшее накопление лишь приводит к увеличению размера базы данных и снижению производительности сервера. Для удобства работы с приложением и поддержания производительности на высоком уровне рекомендуется регулярно проводить очистку данных.

Очистку данных обеспечивает модуль SOLAR Housekeeping. В модуле таблицы приложения Testing Toolkit объединены в группу prf_execution_res. Можно настроить параметры

очистки как на уровне группы, так и для отдельных таблиц. Для настройки конфигурации очистки данных и запуска процедуры очистки у пользователя должен быть доступ к модулю hsk-cfg, а для просмотра результатов очистки — к модулю hsk.

Подробнее о настройке и запуске процедуры очистки данных см. в документе «Solar Housekeeping. Руководство пользователя».

6.4.1 Поддерживаемые таблицы

Процедура очистки данных применяется к следующим таблицам:

- prf_setter_res — результаты установки контекстных переменных,
- prf_validation_res — результаты выполнения проверок,
- prf_step_res_channel — используемые в запуске шага каналы,
- prf_step_res_release — используемые в запуске шага релизы (версии систем),
- prf_step_res — результаты выполнения шагов,
- prf_scenario_res_channel — используемые в запуске сценария каналы,
- prf_scenario_res_release — используемые в запуске сценария релизы (версии систем),
- prf_scenario_res_ps_row — используемые в запуске сценария наборы параметров,
- prf_scenario_res — результаты выполнения сценариев,
- prf_project_res — результаты выполнения проектов,
- prf_plan_res — результаты выполнения планов.

7 Работа с файлами

Приложение SOLAR Testing Toolkit поддерживает следующие действия с файлами:

- Загрузка файла из файловой системы (см. раздел 7.1),
- Выгрузка файла в файловую систему (см. раздел 7.2),
- Проверка наличия файла в файловой системе (см. раздел 7.3).

7.1 Загрузка файлов

7.1.1 Основные сведения о загрузке файлов

Загрузка файлов основана на использовании одного адаптера AnyFile для всех типов файлов. Убедитесь, что параметр `allowDynamicConfiguration` в конфигурации адаптера имеет значение `true`.

Подробнее о поддерживаемых форматах см. раздел 7.1.2.4 «Поддерживаемые типы файлов».

В конфигурации приложения должен присутствовать диалект File In и каналы, основанные на этом диалекте, как показано на рисунке 88.



Код	Наименование	Диалект	Описание	
anyDsvFile	Загрузка файлов DSV с адресацией поля по заданному имени	Загрузка файла	Загрузка файлов DSV	🔍 ✖
anyDsvFile_Index	Загрузка файлов DSV с адресацией поля по индексу колонки	Загрузка файла	Загрузка файлов DSV	🔍 ✖
anyHtmlFile	Загрузка файлов HTML	Загрузка файла	Загрузка файлов HTML	🔍 ✖
anyTextFile	Загрузка текстовых файлов	Загрузка файла	Загрузка текстовых файлов	🔍 ✖
anyXlsxFile	Загрузка файлов xlsx	Загрузка файла	Загрузка файлов xlsx	🔍 ✖
anyXmlFile	Загрузка файлов xml	Загрузка файла	Загрузка файлов xml	🔍 ✖

Рисунок 88 Каналы для загрузки файлов

Один канал предназначен для загрузки файлов определенного формата.

Файлы в некоторых форматах, например DSV или XLSX, могут отличаться дополнительными параметрами. Например, одни DSV-файлы могут иметь заголовки, другие — нет. Чтобы учесть данные различия, необходимо создать отдельный канал с соответствующими настройками для каждой разновидности файлов.

После настройки конфигурации загрузки файлов необходимо создать сценарий и добавить в него тестовый шаг, который загружает файл требуемого вида. Для проверки формата загруженного файла к шагу следует добавить проверки (см. раздел 7.1.5). Для загружаемых XML-файлов поддерживается валидация в соответствии с XSD-схемой (см. раздел 7.1.6).

Загружаемые файлы конвертируются из исходного формата (включая JSON) в XML, что позволяет обращаться к элементам файла с использованием синтаксиса языка XPath 2.0.

7.1.2 Настройка загрузки файлов

Чтобы настроить загрузку файлов:

1. Убедитесь, что в конфигурации приложения присутствует диалект File In. Иначе обратитесь в службу поддержки.
2. Создайте канал на основе диалекта File In (Загрузка файла) для каждого типа загружаемых файлов.

Подробнее см. разделы 4.1.1 «Создание канала» и 7.1.2.1 «Как выбрать тип файла».

3. Задайте базовую директорию.
Подробнее см. раздел 7.1.2.2 «Как указать директорию для загрузки файлов».
4. Создайте сценарий и добавьте к нему тестовый шаг.
Подробнее см. разделы 5.3.3 «Создание тестового сценария», 5.4.1 «Создание тестового шага» и 7.1.2.3 «Как указать имя файла».
Обратите внимание, что направление шага должно быть исходящее, а ответ — обязательный.
5. Добавьте к шагу необходимые проверки (см. раздел 7.1.5).
(Опционально) Для загружаемых XML-файлов настройте валидацию по XSD-схеме (см. раздел 7.1.6).

7.1.2.1 Как выбрать тип файла

Формат файла выбирается в списке **Тип сообщения** в свойствах канала, как показано на рисунке 89.

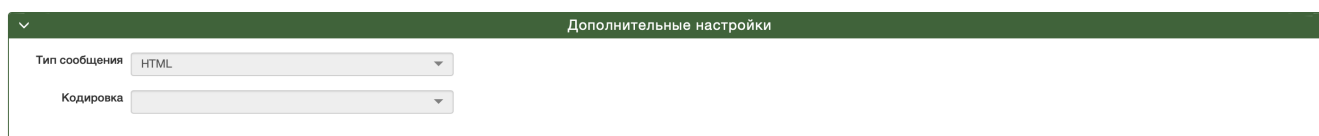


Рисунок 89 Выбор формата файла в списке «Тип сообщения»

Подробнее о поддерживаемых форматах см. раздел 7.1.3 «Поддерживаемые форматы загружаемых файлов».


Выберите в списке требуемый формат файла и заполните дополнительные параметры, как описано в соответствующем разделе.

- DSV (Delimiter-separated value) (см. 7.1.3.1 «Параметры DSV-файлов»)
- HTML (см. 7.1.3.2 «Параметры HTML-файлов»)
- JSON (см. 7.1.3.3 «Параметры JSON-файлов»)
- Plain text (см. 7.1.3.4 «Параметры простых текстовых файлов»)
- XLSX (см. 7.1.3.5 «Параметры XLSX-файлов»)
- XML (см. 7.1.3.6 «Параметры XML-файлов»)

7.1.2.2 Как указать директорию для загрузки файлов

Директория для загрузки файлов задается в свойствах канала для конкретного окружения. Таким образом, можно использовать разные директории для разных окружений. Настройка базовой директории избавляет от необходимости указывать полный путь к файлу в свойствах шага, как описано в разделе 7.1.2.3.

Чтоб выбрать базовую директорию, используемую каналом в некотором окружении:

1. Откройте форму **Каналы (Конфигурация → Каналы)**.
2. Выберите канал, перейдя по ссылке в колонке **Наименование**.
3. Перейдите на вкладку **Конфигурация для окружений**.
4. Найдите окружение и перейдите по ссылке в колонке **Наименование**.
5. Нажмите кнопку  (Редактировать) в правом верхнем углу страницы.
6. Введите в поле **Базовая директория** путь к базовой директории.

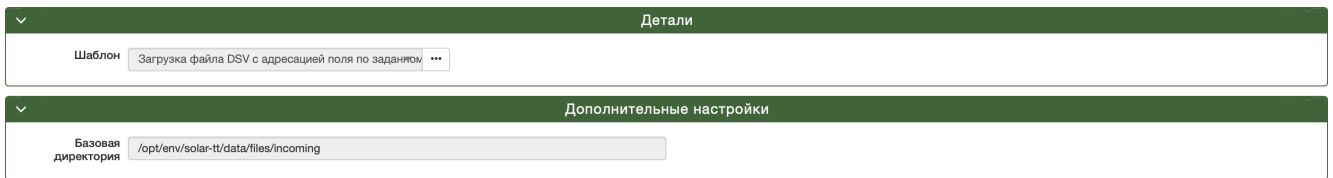


Рисунок 90 Настройка базовой директории

Базовая директория должна содержать следующие директории: *incoming*, *error* и *archived*. Файлы для загрузки следует поместить в директорию *<BASE DIR>/incoming*.

7.1.2.3 Как указать имя файла

Имя загружаемого файла указывается в форме **Данные шага** в свойствах шага. Это может быть абсолютный путь (если базовая директория не задана) или путь относительно директории *<BASE_DIR>/incoming* (как показано на рисунке 91).

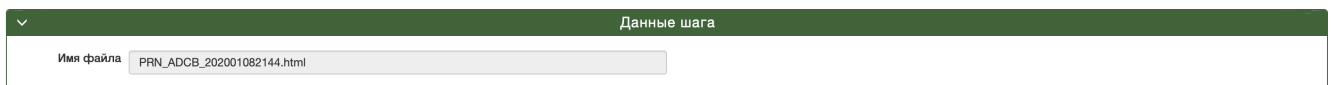


Рисунок 91 Указание имени файла в свойствах шага

7.1.2.4 Поддерживаемые типы файлов

7.1.3 Поддерживаемые форматы загружаемых файлов

Приложение работает с файлами следующих форматов:

- [Delimiter-separated values \(DSV\)](#)
- [HTML](#)
- [JSON](#)
- [Plain text](#)
- [XLSX](#)
- [XML](#)

7.1.3.1 Параметры DSV-файлов

В таблице 7-1 описаны параметры DSV-файлов.

Таблица 7-1 Параметры DSV-файлов

Параметр	Описание
Тип сообщения	Delimiter-Separated Values
Кодировка	Кодировка файла — одно из predefined значений: <ul style="list-style-type: none"> • US_ASCII, • ISO-8859-1, • UTF-8, • UTF-16, • UTF-16BE, • UTF-16LE, • windows-1251

Параметр	Описание
Тип разделителя	<p>Тип разделителя:</p> <ul style="list-style-type: none"> • Запятая, • Точка с запятой, • Таб, • Пробел, • Вертикальный бар, • Пользовательский
Пропустить строки в начале	Количество строк в начале файла, не относящихся в формату, которые отрезаются
Пропустить строки в конце	Количество строк в конце файла, не относящихся в формату, которые отрезаются
Заголовок	<p>Наличие заголовка:</p> <ul style="list-style-type: none"> • Отсутствует, • Присутствует <p>Если заголовок присутствует, он должен соответствовать формату и быть первой читаемой строкой</p>
Адресация поля	<p>В отсутствие заголовка параметр определяет метод обращения к элементам и структуру (имена тегов) преобразованного файла.</p> <ul style="list-style-type: none"> • По индексу колонки (0, 1, 2, ...) — файл преобразуется к виду <code><file><row><column></column></row></file></code>, • По заданному имени («Имена полей») — файл преобразуется к виду <code><file><row><fieldName1></fieldName1><fieldName2></fieldName2>...</row></file></code>, где <code>fieldName1</code>, <code>fieldName2</code> — заданные имена полей. <p>Для обеспечения удобства при настройке проверок рекомендуется выбрать параметр «По заданному имени», поскольку в этом случае обращение к элементам становится проще</p>

При наличии заголовка названия его полей используются в качестве тегов преобразованного XML-файла. Предположим, DSV-файл выглядит, как показано в примере 7-1.

Пример 7-1 DSV-файл с заголовком

```
FirstName|LastName|BirthDate|ClientNumber
John|Smith|1980-09-18|12345678
Anna|Weber|1990-05-15|13213213
Jan|Weiss|2000-01-01|34512324
```

Преобразованный XML-файл имеет формат, показанный в примере 7-2.

Пример 7-2 XML-представление DSV-файла с заголовком

```
<?xml version="1.0" encoding="UTF-8"?>
<file>
```

```
<row>
  <FirstName>John</FirstName>
  <LastName>Smith</LastName>
  <BirthDate>1980-09-18</BirthDate>
  <ClientNumber>12345678</ClientNumber>
</row>
<row>
  <FirstName>Anna</FirstName>
  <LastName>Weber</LastName>
  <BirthDate>1990-05-15</BirthDate>
  <ClientNumber>13213213</ClientNumber>
</row>
<row>
  <FirstName>Jan</FirstName>
  <LastName>Weiss</LastName>
  <BirthDate>2000-01-01</BirthDate>
  <ClientNumber>34512324</ClientNumber>
</row>
</file>
```

Если исходный DSV-файл не содержит заголовка, а адресация полей осуществляется по индексу колонки, тогда преобразованный XML-файл выглядит, как показано в примере 7-3.

Пример 7-3 XML-представление DSV-файла с адресацией полей по индексу колонки

```
<?xml version="1.0" encoding="UTF-8"?>
<file>
  <row>
    <column>John</column>
    <column>Smith</column>
    <column>1980-09-18</column>
    <column>12345678</column>
  </row>
  <row>
    <column>Anna</column>
    <column>Weber</column>
    <column>1990-05-15</column>
    <column>13213213</column>
  </row>
  <row>
    <column>Jan</column>
    <column>Weiss</column>
    <column>2000-01-01</column>
    <column>34512324</column>
  </row>
</file>
```

Когда адресация полей осуществляется по имени, можно определить имена полей, которые служат в качестве тегов преобразованного XML-файла. Предположим, заданы имена полей column1, column2, column3 и column 4. Преобразованный файл выглядит, как показано в примере 7-4.

Пример 7-4 XML-представление DSV-файла с адресацией полей по имени поля

```
<?xml version="1.0" encoding="UTF-8"?>
<file>
```

```
<row>
  <column1>John</column1>
  <column2>Smith</column2>
  <column3>1980-09-18</column3>
  <column4>12345678</column4>
</row>
<row>
  <column1>Anna</column1>
  <column2>Weber</column2>
  <column3>1990-05-15</column3>
  <column4>13213213</column4>
</row>
<row>
  <column1>Jan</column1>
  <column2>Weiss</column2>
  <column3>2000-01-01</column3>
  <column4>34512324</column4>
</row>
</file>
```

7.1.3.2 Параметры HTML-файлов

В таблице 7-2 описаны параметры HTML-файлов.

Таблица 7-2 Параметры HTML-файлов

Параметр	Описание
Тип сообщения	HTML
Кодировка	Кодировка файла — одно из predefined значений: <ul style="list-style-type: none">• US_ASCII,• ISO-8859-1,• UTF-8,• UTF-16,• UTF-16BE,• UTF-16LE,• windows-1251

В примере 7-5 показан входящий HTML-файл.

Пример 7-5 входящий HTML-файл

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <body>
    <table>
      <tr>
        <td>John</td>
        <td>Smith</td>
        <td>1980-09-18</td>
        <td>12345678</td>
      </tr>
      <tr>
        <td>Anna</td>
        <td>Weber</td>
```

```

        <td>1990-05-15</td>
        <td>13213213</td>
    </tr>
    <tr>
        <td>Jan</td>
        <td>Weiss</td>
        <td>2000-01-01</td>
        <td>34512324</td>
    </tr>
</table>
</body>
</html>

```

7.1.3.3 Параметры JSON-файлов

В таблице 7-3 описаны параметры JSON-файлов.

Таблица 7-3 Параметры JSON-файлов

Параметр	Описание
Тип сообщения	JSON
Кодировка	Кодировка файла — одно из predefined значений: <ul style="list-style-type: none"> • US_ASCII, • ISO-8859-1, • UTF-8, • UTF-16, • UTF-16BE, • UTF-16LE, • windows-1251

7.1.3.4 Параметры простых текстовых файлов

В таблице 7-4 описаны параметры простых текстовых файлов.

Таблица 7-4 Параметры простых текстовых файлов

Параметр	Описание
Тип сообщения	Простой текст
Кодировка	Кодировка файла — одно из predefined значений: <ul style="list-style-type: none"> • US_ASCII, • ISO-8859-1, • UTF-8, • UTF-16, • UTF-16BE, • UTF-16LE, • windows-1251

В примере 7-6 показан входящий файл в простом текстовом формате.

Пример 7-6 Простой текстовый файл

```

John Smith 1980-09-18 12345678
Anna Weber 1990-05-15 13213213

```


Jan Weiss 2000-01-01 34512324

Его XML-представление показано в примере 7-7.

Пример 7-7 XML-представление текстового файла

```
<?xml version="1.0" encoding="UTF-8"?>
<file>
  <row>
    <data>John Smith 1980-09-18 12345678</data>
  </row>
  <row>
    <data>Anna Weber 1990-05-15 13213213</data>
  </row>
  <row>
    <data>Jan Weiss 2000-01-01 34512324</data>
  </row>
</file>
```

7.1.3.5 Параметры XLSX-файлов

В таблице 7-5 описаны параметры XLSX-файлов.

Таблица 7-5 Параметры XLSX-файлов

Параметр	Описание
Тип сообщени я	XLSX
Пропусти ть строки в начале	Количество строк в начале файла, не относящихся в формату, которые отрезаются
Пропусти ть строки в конце	Количество строк в конце файла, не относящихся в формату, которые отрезаются
Заголово к	Наличие заголовка: <ul style="list-style-type: none"> Отсутствует, Присутствует Если заголовок присутствует, он должен соответствовать формату и быть первой читаемой строкой
Адресаци я поля	В отсутствие заголовка параметр определяет метод обращения к элементам и структуру (имена тегов) преобразованного файла. <ul style="list-style-type: none"> По индексу колонки (0, 1, 2, ...) — файл преобразуется к виду <file><row><column></column></row></file> По имени колонки (A, B, C, ...) — файл преобразуется к виду <file><row><A>...</row></file> По заданному имени («Имена полей») — файл преобразуется к виду <file><row><fieldName1></fieldName1><fieldName2></fieldName2>...</ row></file>, где fieldName1, fieldName2 — заданные имена полей

При наличии заголовка названия его полей используются в качестве тегов преобразованного XML-файла. Предположим, XLSX-файл выглядит, как показано на рисунке 7.1.3.5.

	A	B	C	D
1	FirstName	LastName	BirthDate	ClientNumber
2	John	Smith	1980-09-18	12345678
3	Anna	Weber	1990-05-15	13213213
4	Jan	Weiss	2000-01-01	34512324
5				

Рисунок 92 Исходный XLSX-файл с заголовком

Преобразованный XML-файл имеет формат, показанный в примере 7-8.

Пример 7-8 XML-представление XLSX-файла с заголовком

```
<?xml version="1.0" encoding="UTF-8"?>
<file>
  <row>
    <FirstName>John</FirstName>
    <LastName>Smith</LastName>
    <BirthDate>1980-09-18</BirthDate>
    <ClientNumber>12345678</ClientNumber>
  </row>
  <row>
    <FirstName>Anna</FirstName>
    <LastName>Weber</LastName>
    <BirthDate>1990-05-15</BirthDate>
    <ClientNumber>13213213</ClientNumber>
  </row>
  <row>
    <FirstName>Jan</FirstName>
    <LastName>Weiss</LastName>
    <BirthDate>2000-01-01</BirthDate>
    <ClientNumber>34512324</ClientNumber>
  </row>
</file>
```

Если исходный XLSX-файл не содержит заголовка, а адресация полей осуществляется по индексу колонки, тогда преобразованный XML-файл выглядит, как показано в примере 7-9.

Пример 7-9 XML-представление XLSX-файла с адресацией полей по индексу колонки

```
<?xml version="1.0" encoding="UTF-8"?>
<file>
  <row>
    <column>John</column>
    <column>Smith</column>
    <column>1980-09-18</column>
    <column>12345678</column>
  </row>
  <row>
    <column>Anna</column>
    <column>Weber</column>
```

```
<column>1990-05-15</column>
<column>13213213</column>
</row>
<row>
  <column>Jan</column>
  <column>Weiss</column>
  <column>2000-01-01</column>
  <column>34512324</column>
</row>
</file>
```

Когда адресация полей осуществляется по имени колонки, имена колонок служат в качестве тегов преобразованного XML-файла. Преобразованный файл выглядит, как показано в примере 7-10.

Пример 7-10 XML-представление XLSX-файла с адресацией полей по имени колонки

```
<?xml version="1.0" encoding="UTF-8"?>
<file>
  <row>
    <A>John</A>
    <B>Smith</B>
    <C>1980-09-18</C>
    <D>12345678</D>
  </row>
  <row>
    <A>Anna</A>
    <B>Weber</B>
    <C>1990-05-15</C>
    <D>13213213</D>
  </row>
  <row>
    <A>Jan</A>
    <B>Weiss</B>
    <C>2000-01-01</C>
    <D>34512324</D>
  </row>
</file>
```

Когда адресация полей осуществляется по заданному имени, можно определить имена полей, которые служат в качестве тегов преобразованного XML-файла.

Предположим, заданы имена полей column1, column2, column3 и column 4.

Преобразованный файл выглядит, как показано в примере 7-11.

Пример 7-11 XML-представление XLSX-файла с адресацией полей по заданному имени

```
<?xml version="1.0" encoding="UTF-8"?>
<file>
  <row>
    <column1>John</column1>
    <column2>Smith</column2>
    <column3>1980-09-18</column3>
    <column4>12345678</column4>
  </row>
  <row>
    <column1>Anna</column1>
    <column2>Weber</column2>
```

```

    <column3>1990-05-15</column3>
    <column4>13213213</column4>
</row>
<row>
    <column1>Jan</column1>
    <column2>Weiss</column2>
    <column3>2000-01-01</column3>
    <column4>34512324</column4>
</row>
</file>

```

7.1.3.6 Параметры XML-файлов

В таблице 7-6 описаны параметры XML-файлов.

Таблица 7-6 Параметры XML-файлов

Параметр	Описание
Тип сообщения	XML
Кодировка	Кодировка файла — одно из predefined значений: <ul style="list-style-type: none"> • US_ASCII, • ISO-8859-1, • UTF-8, • UTF-16, • UTF-16BE, • UTF-16LE, • windows-1251

7.1.4 Загрузка файла и проверка результатов

Чтобы загрузить файл требуемого типа, запустите соответствующий сценарий (см. раздел 6.1 «Запуск сценария»).

Результат запуска сценария отображается в форме **Запуска сценария** (см. раздел 6.2 «Просмотр результатов запуска»).

Чтобы посмотреть результат выполнения шага:

1. Откройте форму **Запуск сценария** по ссылке в колонке **Описание**.
2. В форме **Результаты выполнения шагов сценария** перейдите по ссылке в колонке **Шаг**.
3. В форме **Входящее сообщение** отображается имя файла и его XML-представление.

7.1.5 Проверки загруженных файлов

После загрузки файла необходимо проверить, что файл имеет правильный формат. Для этого нужно добавить все необходимые проверки в форму **Проверки** соответствующего тестового шага.

Поскольку загруженный файл представлен в формате XML, при заполнении поля **Выражение** в проверках и установке контекстных переменных можно обращаться к элементам файла с использованием синтаксиса языка XPath 2.0.

Язык XPath разработан консорциумом W3C для организации доступа к частям документа XML. XPath обращается к узлам XML-документа, используя навигацию по иерархическому дереву документа.

Подробнее о языке XPath 2.0, см. спецификацию <https://www.w3.org/TR/xpath20/>.

Подробнее о проверках, см. раздел 5.4.7 «Проверки».

Подробнее о контекстных переменных, см. раздел 5.4.6 «Контекстные переменные».

В примере 7-12 показан XML-документ, используемый в последующих разделах для демонстрации различных способов адресации полей.

Пример 7-12 XML-документ

```
<?xml version="1.0" encoding="UTF-8"?>
<file>
  <row>
    <column1>123456789</column1>
    <column2>A</column2>
    <column3>JOHN WATSON</column3>
    <column4>100</column4>
    <column5>552006xxxxxx4545</column5>
  </row>
  <row>
    <column1>1234567890</column1>
    <column2>B</column2>
    <column3>SCOTT PILIGRIM</column3>
    <column4>200</column4>
    <column5>552006xxxxxx5050</column5>
  </row>
</file>
```

7.1.5.1 Проверка количества тегов

Предположим, необходимо убедиться в правильном количестве строк (элементов row). Для этой цели используйте функцию `count()` следующим образом.

```
count(/file/row)
```

Значение этого выражения — 2, поскольку файл в примере 7-12 содержит две строки. Заполненная форма **Проверка** показана на рисунке 93.

Детали			
Наименование	<input type="text" value="Число строк"/>	Источник	<input type="text" value="Входящее сообщение"/>
Выражение	<input type="text" value="count(/file/row)"/>	Проверка	<input type="text" value="Равно"/>
Тип данных	<input type="text" value="Строка"/>	Проверка	<input type="text" value="Равно"/>
Ожидаемое значение	<input type="text" value="2"/>	Проверка	<input type="text" value="Равно"/>

Рисунок 93 Проверка количества строк

Теперь допустим, необходимо убедиться в правильном количестве колонок в конкретной строке. Для этой цели используйте следующее выражение XPath, в котором применена адресация полей по индексу.

```
count(/file/row[1]/*)
```

Значение этого выражения — 5. Соответствующая форма **Проверка** показана на рисунке 94.

Детали

Наименование: Число колонок в первой строке

Источник: Входящее сообщение

Выражение: `count(/file/row[1]/*)`

Тип данных: Строка

Проверка: Равно

Ожидаемое значение: 5

Формат:

Формат:

Рисунок 94 Проверка количества колонок в первой строке

Если требуется убедиться, что каждая строка содержит 5 колонок, можно использовать следующее выражение XPath.

```
count(/file/row[count(./*) = 5]) = count(/file/row)
```

Значение выражения — `true`, если все строки содержат ровно 5 колонок. Соответствующая форма **Проверка** показана на рисунке 95.

Детали

Наименование: Число колонок в каждой строке = 5

Источник: Входящее сообщение

Выражение: `count(/file/row[count(*) = 5]) = count(/file/row)`

Тип данных: Строка

Проверка: Равно

Ожидаемое значение: 5

Формат:

Формат:

Рисунок 95 Проверка наличия 5 колонок в каждой строке

7.1.5.2 Проверка длины строки

Предположим, необходимо проверить, что значение поля `column1` первой строки имеет определенную длину. Для этой цели необходимо использовать следующее выражение XPath, в котором применена адресация поля по заданному имени (`column1`) в строке, выбранной по индексу (`row[1]`).

```
string-length(/file/row[1]/column1/text())
```

Значение выражения — 9 (длина строки `'123456789'`). Соответствующая форма **Проверка** показана на рисунке 96.

Детали

Наименование: Длина строки

Источник: Входящее сообщение

Выражение: `string-length(/file/row[1]/column1/text())`

Тип данных: Строка

Проверка: Равно

Ожидаемое значение: 9

Формат:

Формат:

Рисунок 96 Проверка длины строки

7.1.5.3 Проверка строки

Предположим, необходимо проверить значение поля `column3` (имя держателя карты) в строке, которая имеет 10 цифр в поле `column1`. Для этой цели можно применить функцию `matches(string, regex)`. В данном выражении XPath применяется адресация строки по содержимому поля, которое выбирается по заданному имени (`column1`).

```
/file/row[matches(column1, '\d{10}')]/column3
```

Значение выражения — `'SCOTT PILGRIM'`. Соответствующая форма **Проверка** показана на рисунке 97.

Детали	
Наименование	Проверка строки
Источник	Входящее сообщение
Выражение	/file/row[matches(column1, ^d(10))]/column3
Тип данных	Строка
Проверка	Равно
Ожидаемое значение	SCOTT PILGRIM

Рисунок 97 Проверка строки

Возможен другой вариант выражения XPath, в котором используется адресация поля по заданному имени (column3), а строка выбирается по содержимому ('1234567890') поля, адресованному по индексу ([1]).

```
/file/row[.*[1] = '1234567890']/column3
```

Значение выражения — также 'SCOTT PILGRIM'.

7.1.5.4 Проверка нескольких строк

Предположим, что несколько строк соответствуют выражению и необходимо проверить их все. Например, рассмотрим следующее выражение XPath.

```
//column3
```

Ему соответствуют две строки: 'JOHN WATSON' и 'SCOTT PILGRIM'. Чтобы проверить эти строки, можно применить функцию `string-join(strings, delimiter)`.

```
string-join(//column3, ',')
```

Значение выражения — строка 'JOHN WATSON, SCOTT PILGRIM'. Соответствующая форма **Проверка** показана на рисунке 98.

Детали	
Наименование	Проверка имен держателей карт
Источник	Входящее сообщение
Выражение	string-join(//column3, ',')
Тип данных	Строка
Проверка	Равно
Ожидаемое значение	JOHN WATSON, SCOTT PILGRIM

Рисунок 98 Проверка нескольких строк

7.1.5.5 Проверка результата объединения тегов

Если требуется проверить значения нескольких тегов, используйте следующее выражение XPath.

```
string-join(//row[1]/(column1 | column2 | column3), '')
```

Значение выражения — строка '123456789 A JOHN WATSON'. Соответствующая форма **Проверка** показана на рисунке 99.

Детали	
Наименование	Проверка колонок 1,2,3 в первой строке
Источник	Входящее сообщение
Выражение	string-join(//row[1]/(column1 column2 column3), '')
Тип данных	Строка
Проверка	Равно
Ожидаемое значение	123456789 A JOHN WATSON

Рисунок 99 Проверка результата объединения тегов

7.1.5.6 Проверка суммы числовых значений

Предположим, необходимо проверить сумму числовых значений нескольких тегов. Для этой цели используйте следующее выражение XPath.

```
sum(//row/column4)
```

Значение выражения — 300 (сумма значений 100 и 200 тегов `column4`. Соответствующая форма **Проверка** показана на рисунке 100.

Детали			
Наименование	Проверка суммы	Источник	Входящее сообщение
Выражение	sum(//row/column4)		Формат
Тип данных	Строка	Проверка	Равно
Ожидаемое значение	300		Формат

Рисунок 100 Проверка суммы числовых значений

7.1.5.7 Проверка всех значений файла с помощью сравнения таблиц

Если требуется проверить все значения из файла, можно применить проверку типа «Сравнение таблиц». Чтобы выбрать все значения из XML-файла, показанного в примере 7-12, используйте следующее выражение 7-12

```
/file/row
```

При создании проверки выберите следующие параметры (см. рисунок 101):

- **Тип данных** – Таблица,
- **Проверка** – Сравнение таблиц,
- **Ожидаемое значение**
 - Добавьте названия переменных (в описанном случае это `column1`, `column2` и т.д.), укажите тип данных и тип проверки,
 - Добавьте число строк, соответствующее числу строк в файле, и укажите ожидаемые значения элементов таблицы.

Детали																																													
Наименование	Проверка таблиц	Источник	Входящее сообщение																																										
Выражение	file/row																																												
Тип данных	Таблица	Проверка	Сравнение таблиц																																										
Ожидаемое значение	<table border="1"> <thead> <tr> <th>Переменная</th> <th>Тип данных</th> <th>Тип проверки</th> <th></th> </tr> </thead> <tbody> <tr> <td>column1</td> <td>Строка</td> <td>Равно</td> <td> </td> </tr> <tr> <td>column2</td> <td>Строка</td> <td>Равно</td> <td></td> </tr> <tr> <td>column3</td> <td>Строка</td> <td>Равно</td> <td></td> </tr> <tr> <td>column4</td> <td>Строка</td> <td>Равно</td> <td></td> </tr> <tr> <td>column5</td> <td>Строка</td> <td>Равно</td> <td></td> </tr> <tr> <td>column1</td> <td>column2</td> <td>column3</td> <td>column4</td> <td>column5</td> <td> </td> </tr> <tr> <td>123456789</td> <td>A</td> <td>JOHN WATSON</td> <td>100</td> <td>552006xxxxxx4545</td> <td></td> </tr> <tr> <td>1234567890</td> <td>B</td> <td>SCOTT PILGRIM</td> <td>200</td> <td>552006xxxxxx5050</td> <td></td> </tr> </tbody> </table>			Переменная	Тип данных	Тип проверки		column1	Строка	Равно		column2	Строка	Равно		column3	Строка	Равно		column4	Строка	Равно		column5	Строка	Равно		column1	column2	column3	column4	column5		123456789	A	JOHN WATSON	100	552006xxxxxx4545		1234567890	B	SCOTT PILGRIM	200	552006xxxxxx5050	
Переменная	Тип данных	Тип проверки																																											
column1	Строка	Равно																																											
column2	Строка	Равно																																											
column3	Строка	Равно																																											
column4	Строка	Равно																																											
column5	Строка	Равно																																											
column1	column2	column3	column4	column5																																									
123456789	A	JOHN WATSON	100	552006xxxxxx4545																																									
1234567890	B	SCOTT PILGRIM	200	552006xxxxxx5050																																									

Рисунок 101 Проверка всех значений файла с помощью сравнения таблиц

7.1.5.8 Проверка значения атрибута

При работе с HTML-файлами часто требуется проверить значение атрибута. Рассмотрим следующий HTML-файл, показанный в примере 7-13.

Пример 7-13 HTML-файл для проверки

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <body>
    <table>
      <tr>
        <td cardId="10">552006xxxxxx4545</td>
        <td>IVAN IVANOV</td>
      </tr>
      <tr>
        <td cardId="20">552006xxxxxx5050</td>
        <td>JOHN SMITH</td>
      </tr>
      <tr>
        <td cardId="30">552006xxxxxx5555</td>
        <td>PETER JOHNSON</td>
      </tr>
    </table>
  </body>
</html>
```

Предположим, требуется проверить значение атрибута cardId значения в первой колонке второй строки. Для этой цели используйте следующее выражение XPath.

```
//table/tr[2]/td[@cardId]/@cardId
```

Значение выражения — 10. Соответствующая форма **Проверка** показана на рисунке 102.

Детали			
Наименование	<input type="text" value="Проверка ID карты во второй строке таблицы"/>	Источник	<input type="text" value="Входящее сообщение"/>
Выражение	<input type="text" value="//table/tr[2]/td[@cardId]/@cardId"/>	Формат	<input type="text"/>
Тип данных	<input type="text" value="Строка"/>	Проверка	<input type="text" value="Равно"/>
Ожидаемое значение	<input type="text" value="10"/>	Формат	<input type="text"/>

Рисунок 102 Проверка значения атрибута

Если требуется одновременно проверить значения атрибута cardId у всех элементов, примените функцию string-join, как показано в следующем выражении XPath.

```
string-join(//table/tr/td[@cardId]/@cardId, ' ')
```

Значение выражения — строка '10 20 30'. Соответствующая форма **Проверка** показана на рисунке 103.

Детали			
Наименование	<input type="text" value="Проверка ID карт во всех строках таблицы"/>	Источник	<input type="text" value="Входящее сообщение"/>
Выражение	<input data-bbox="279 1704 922 1727" type="text" value="string-join(//table/tr/td[@cardId]/@cardId, ' ')"/>	Формат	<input type="text"/>
Тип данных	<input type="text" value="Строка"/>	Проверка	<input type="text" value="Равно"/>
Ожидаемое значение	<input type="text" value="10 20 30"/>	Формат	<input type="text"/>

Рисунок 103 Проверка всех значений файла с помощью сравнения таблиц

Ниже приведены примеры выражений XPath, полезные при работе с таблицами.

```
//table[*] — выбрать значение всех потомков тега 'table'
```

```
//table[1]/tr[last()] — выбрать последнюю строку первой таблицы
```

```
//table/tr/td[not(@cardId)] - выбрать значения всех колонок, у которых нет атрибута 'cardId'  
  
//table/tr/td[@cardId] - выбрать значения всех колонок, у которых есть атрибут 'cardId'  
  
//table/tr/td[matches(@cardId, '\d+')] - выбрать все номера карт, у которых есть числовой атрибут 'cardId'  
  
//table/tr/td[matches(., '\d{6}x{6}\d{4}')]/@cardId - выбрать значения атрибута 'cardId' колонок таблицы, содержимое которых соответствует заданной маске  
  
//table/tr[./td[@cardId = '20']/td[2] - выбрать имя держателя карты с атрибутом 'cardId' равным 20
```

7.1.5.9 Проверка содержимого подстроки

При работе с простыми текстовыми файлами может потребоваться проверить содержимое подстроки строки, на которую ссылаются с помощью содержимого другой подстроки.

Рассмотрим следующий XML-файл, показанный в примере 7-14, который является XML-представлением простого текстового файла.

Пример 7-14 XML-файл для проверки

```
<?xml version="1.0" encoding="UTF-8"?>  
<file>  
  <row>  
    <data>000051598752114967920000M96792</data>  
  </row>  
  <row>  
    <data>000047414299901735170000 M73517</data>  
  </row>  
  <row>  
    <data> 000051598703126018530000M01853 </data>  
  </row>  
  <row>  
    <data>000047414299908697420000 M69742</data>  
  </row>  
  <row>  
    <data/>  
  </row>  
</file>
```

Если требуется проверить содержимое подстроки строки, на которую ссылаются с помощью содержимого другой подстроки, используйте функцию `substring(str, start, len)`.

```
substring(/file/row[substring(./data, 25, 6) = 'M96792']/data, 1, 24)
```

Значение выражения — 000051598752114967920000. Соответствующая форма **Проверка** показана на рисунке 104.

Детали			
Наименование	Проверка первых 24 символов в строке, заканчивающей	Источник	Входящее сообщение
Выражение	substring(file/row[substring(/data, 25, 6) = 'M96792']/data, 1, 24)		Формат
Тип данных	Строка	Проверка	Содержит подстроку
Ожидаемое значение	000051598752114967920000		Формат

Рисунок 104 Проверка содержимого подстроки

Если требуется проверить значение перед пробелом во второй строке, используйте функцию `substring-before(string, delimiter)`.

```
substring-before(/file/row[2]/data, ' ')
```

Значение выражения — 000047414299901735170000. Соответствующая форма **Проверка** показана на рисунке 105.

Детали			
Наименование	Значение перед пробелом во второй строке	Источник	Входящее сообщение
Выражение	substring-before(/file/row[2]/data, ' ')		Формат
Тип данных	Строка	Проверка	Содержит подстроку
Ожидаемое значение	000047414299901735170000		Формат

Рисунок 105 Проверка подстроки перед разделителем

Если требуется проверить значение после пробела в четвертой строке, используйте функцию `substring-after(string, delimiter)`.

```
substring-after(/file/row[4]/data, ' ')
```

Значение выражения — M69742. Соответствующая форма **Проверка** показана на рисунке 106.

Детали			
Наименование	Проверка значения в четвертой строке после пробела	Источник	Входящее сообщение
Выражение	substring-after(/file/row[4]/data, ' ')		Формат
Тип данных	Строка	Проверка	Содержит подстроку
Ожидаемое значение	M69742		Формат

Рисунок 106 Проверка подстроки после разделителя

Если требуется проверить обрезанное значение в третьей строке, используйте функцию `normalize-space(string)`.

```
normalize-space(/file/row[3]/data)
```

Значение выражения — 000051598703126018530000M01853 (значение элемента data в третьей строке без пробелов в начале и конце).

7.1.6 Настройка валидации XML-файлов по XSD-схеме

Для валидации XML-файлов по XSD-схеме необходимо указать в свойствах канала путь к XSD-схеме (путь задается для конкретного окружения), а в свойствах шага установить флажок **Проверять формат**.

Чтобы указать путь к XSD-схеме:

1. Откройте форму **Каналы (Конфигурация → Каналы)**.
2. Выберите канал, перейдя по ссылке в колонке **Наименование**.

3. Перейдите на вкладку **Окружения с использованием канала**.
4. Выберите окружение, перейдя по ссылке в колонке **Наименование**.
5. В поле **Путь к XML схеме файла** введите путь относительно директории *CONFIG_DIR*.

Чтобы выбрать опцию проверки формата:

1. Откройте форму сценария загрузки XML-файла.
2. Перейдите на форму шага по ссылке в колонке **Наименование**.
3. Установите флажок **Проверить формат**.

7.2 Выгрузка файлов в файловую систему

Приложение поддерживает выгрузку файлов заданных форматов (см. раздел 7.2.1) и произвольного формата.

7.2.1 Поддерживаемые форматы выгружаемых файлов

Приложение работает с файлами следующих форматов:

- XML,
- JSON,
- Простой текст,
- HTML.

7.2.2 Выгрузка файла произвольного формата

Чтобы выгрузить файл произвольного формата, выполните следующие действия:

1. Создайте канал с диалектом RAW_OUT.
Подробнее о каналах см. в разделе 4.1 «Каналы».
2. Создайте сценарий и добавьте к нему шаг, производящий выгрузку файла.
При создании шага выберите канал с диалектом RAW_OUT, укажите имя файла и скопируйте содержимое файла в поле **Данные шага**.
3. Запустите сценарий.

7.2.3 Выгрузка файла заданного формата

Подробнее о поддерживаемых форматах файлов см. в разделе 7.2.1

Чтобы выгрузить файл заданного формата, выполните следующие действия:

1. Создайте канал с диалектом FILE_OUT и типом сообщения, соответствующего формату файла.
Подробнее о каналах см. в разделе 4.1 «Каналы».
2. Создайте сценарий и добавьте к нему шаг, производящий выгрузку файла.
При создании шага выберите канал с диалектом FILE_OUT, укажите имя файла и скопируйте выгружаемые данные в соответствующем формате в поле **Данные шага**.
3. Запустите сценарий.

Чтобы продолжить выгружать файлы в формате, использовавшемся в версии 2.6.0 и ранее, при создании канала не нужно выбирать тип сообщения, а в поле **Данные шага** необходимо задать XML-представление файла.

Подробнее об XML-представлении см. в разделе 7.2.3.1.

7.2.3.1 XML-представление выгружаемого файла

XML-представление задает структуру выгружаемых данных. Корневым элементом обязательно должен быть элемент `<fileDefinition>`, который содержит один или несколько

элементов <records> с вложенными элементами <record>. Разрешается использовать заполнители, которые при исполнении шага заменяются переменными контекста.

Вариант XML-представления показан в примере 7-15.

Пример 7-15 XML-представление выгружаемого файла

```
<?xml version="1.0" encoding="UTF-8"?>
<fileDefinition xmlns="">
  <records>
    <record>
      <RC>RD</RC>
      <RN>002</RN>
      <data>someData2</data>
    </record>
  </records>
</fileDefinition>
```

7.3 Проверка наличия файла в файловой системе

Чтобы проверить, существует ли файл в файловой системе, выполните следующие действия:

1. Создайте канал с диалектом CHECK_FILE.
Подробнее о каналах см. в разделе 4.1 «Каналы»;
2. Создайте сценарий и добавьте к нему шаг, производящий проверку наличия файла.
Подробнее о сценариях и тестовых шагах см. в разделах 5.3 «Тестовые сценарии» и 5.4 «Тестовые шаги».
При создании шага выберите канал с диалектом CHECK_FILE и укажите имя файла: абсолютный путь в файловой системе или путь относительно папки, выбранной администратором системы для хранения файлов данного формата.
3. Создайте проверку для интерпретации значения, полученного в ответном сообщении.
Подробнее о проверках см. в разделе 5.4.7 «Проверки».
4. Запустите сценарий.

8 Поддержка криптографических действий

Для поддержки криптографических действий используются следующие контекстные функции:

- deriveIccSessionKey,
- deriveIccMasterKey,
- rawGenerateAC,
- isoGenerateAC,
- cvc3,
- encryptClearPinBlockIso0.

Типы ключей криптограмм, которые должны быть прописаны на стороне процессинга, перечислены в таблице 8-1.

Таблица 8-1 Типы ключей криптограмм

Функция	Тип ключа криптограммы на стороне процессинга
deriveIccSessionKey	ТС МК
deriveIccMasterKey	ТС МК
rawGenerateAC	ТС МК
isoGenerateAC	ТС МК
cvc3	CVK
encryptClearPinBlockIso0	ZPK (PEK,IWK,AWK), TPK

Функции можно использовать везде, где применимы контекстные переменные. Синтаксис обращения к ним одинаков:

```
${function('param1', 'param2')}
```

Все функции работают с чистыми значениями ключей. Обратите внимание, что в промышленной среде используются ключи в виде криптограмм под LMK.

В случае когда одновременно используются чистые ключи и ключи под LMK, LMK должен быть тестовым (см. раздел 8.1). Категорически не допускается использование промышленного ключа LMK во избежание его компрометации.

8.1 Тестовый ключ LMK

Каждое устройство Thales HSM поставляется в комплекте со стандартным тестовым ключом LMK, для которого подготовлены все виды ключей в открытом виде и в виде криптограммы.

Таблица 8-2 содержит криптограммы тестового ключа (разных типов Thales HSM) под стандартным ключом LMK. Все ключи, кроме ZMK, в открытом виде представляют собой значение 0123456789ABCDEF0123456789 (проверочное значение — EE21F1), криптограмма этого значения под ZMK — 8479E4A3DBEB93D778100919833FC92F.

Значение ZMK в чистом виде: 23232323232323252525252525252525.

ZMK, разбитый по компонентам:

ZMK 1 = ABCDEF01234567890123456789ABCDEF, KCV = 5CAFF2

ZMK 2 = 0123456789ABCDEFABCDEF0123456789, KCV = EE21F1

ZMK 3 = 89CD894589CD89458FCB8F438FCB8F43, KCV = EDE54C

Таблица 8-2 Криптограммы тестового ключа

Key Type	HSM (Scheme U)	HSM (Scheme X)	Key Check Value
ZMK (00 0)	U1D1225FC6487FCDB995CC6DACE1 14171	—	AD2F4 F
TPK (00 2)	U1750CDFB0757D3B329B70456A44 EC9FA	FCBA7CF5972CF0DDE0F8093DDC99 9D12	EE21F 1
TMK (00 2)	U1750CDFB0757D3B329B70456A44 EC9FA	FCBA7CF5972CF0DDE0F8093DDC99 9D12	EE21F 1
PVK (00 2)	U1750CDFB0757D3B329B70456A44 EC9FA	FCBA7CF5972CF0DDE0F8093DDC99 9D12	EE21F 1
PVK 1 (00 2)	FCBA7CF5972CF0DD	—	D5D4 4F
PVK 2 (00 2)	E0F8093DDC999D12	—	7229F 9
TAK (00 3)	U7AD0BB2A32ADE1156886524A5F1 F6384	X74385088A93CEAD592B487F281D DA0FE	EE21F 1
TDK (00 3)	U7AD0BB2A32ADE1156886524A5F1 F6384	X74385088A93CEAD592B487F281D DA0FE	EE21F 1
CVK (40 2)	U9B4934384B19946BB8A51FE7340 9DC16	0A61E674E88C6A7E29DC82464E58 5A01	EE21F 1
CVK A (40 2)	0A61E674E88C6A7E	—	D5D4 4F

Key Type	HSM (Scheme U)	HSM (Scheme X)	Key Check Value
CVK B (40 2)	29DC82464E585A01	—	7229F9
TC MK (10 9)	U7475636CC30B93B417194425DC51A6AC	842D970B0F7BAB4ABB4547D5F08EF5F7	EE21F1
MAC MK (20 9)	U352796B7EF432BE048C1E2D050997DD1	72DDDBE3F5B41E13AE050DA1C22506CE	EE21F1
Enc MK (30 9)	U8430868258D183959E80B4600F63F5B6	8A55D00FB556C20CDD9386CF4D4CC311	EE21F1
DAC MK (40 9)	UA1E8089C9128CA1011A846666A395711	3517A711780BDD517F8307AE3039AF89	EE21F1
DN MK (50 9)	U7BCA7BDC2CFD97223BC9042785690E5C	09DDB90A89CAB543CED79D4F67DC0D84	EE21F1
DAC MK (60 9)	UFE4161E9CF7373ADC27316713ABF05DC	8F4CA70DA0AA0EBAFC01F0067E597A07	EE21F1
DN MK (70 9)	U2E774AEF908AFBA1C8F0382A8AB919BC	790E83DB75ACB4773C48B42AAF89B93B	EE21F1
KEK (10 7)	UA940CC330472671D0CDA49CCF19924DD	4A5ADBEA51E41D985E487AB030234D42	EE21F1

Key Type	HSM (Scheme U)	HSM (Scheme X)	Key Check Value
ZPK (PE K) (00 1)	U063A0E7C0F2124E5E7CBB04D9CC D22F3	—	EE21F1

8.2 Функции deriveIccSessionKey и deriveIccMasterKey

Функция deriveIccSessionKey предназначена для формирования секретного сессионного ключа (Icc Session Key). Параметры функции описаны в таблице 8-3.

Таблица 8-3 Параметры функции deriveIccSessionKey

Параметр	Описание
dialect	VISA, MC, MIR или контекстная переменная
version	Для диалекта VISA: 10, 14, 17 или 18. Для диалектов MC и MIR: пустая строка
transactionCounter	Значение счетчика транзакций АТС (Application Transaction Counter). Шестнадцатеричное значение длиной 2 байта, например 0001
iccMasterKey	Значение главного ключа (ICC Master Key), выдаваемое функцией deriveIccMasterKey
unpredictableNumber	Числовое значение для повышения криптографической стойкости шифрования. Шестнадцатеричное значение длиной 4 байта, например 00000000

Функция deriveIccMasterKey предназначена для формирования главного ключа (Icc Master Key). Параметры функции описаны в таблице 8-4.

Таблица 8-4 Параметры функции deriveIccMasterKey

Параметр	Описание
dialect	VISA, MC, MIR или контекстная переменная
version	Для диалекта VISA: 10, 14, 17 или 18. Для диалектов MC и MIR: пустая строка
pan	Номер карты (PAN)
panSequenceNumber	Последовательный номер карты
issuerMasterKey	Значение ключа Issuer Master Key, выданное финансовой организацией

8.3 Функции rawGenerateAC и isoGenerateAC

Функция rawGenerateAC предназначена для генерации зашифрованного текста на основании параметров, описанных в таблице 8-5.

Таблица 8-5 Параметры функции rawGenerateAC

Параметр	Описание
dialect	VISA, MC, MIR или контекстная переменная
version	Для диалекта VISA: 10, 14, 17 или 18. Для диалектов MC и MIR: пустая строка
iccSessionKey	Значение сессионного ключа, которое может быть получено при помощи функции deriveIccSessionKey
terminalData	Данные, предоставляемые терминалом
iccData	Icc data

Вызов функции rawGenerateAC с параметрами, указанными в примере 8-1, дает результат 775297A996052288.

Пример 8-1 Использование функции rawGenerateAC

```
$(rawGenerateAC('VISA', '14', '16F402C1CD45C207B65000A10BD5FE34',
'000000001000000000000000071000000000007101302050030901B6A',
'3C00005503A4A082'))
```

Функция isoGenerateAC также предназначена для генерации зашифрованного текста. Функция имеет обязательный аргумент issuerMasterKey и извлекает остальные необходимые параметры из данных ISO- сообщения, как описано в таблице 8-6.

Необязательный аргумент generateKeyByEmv влияет на способ генерации ключа ICC Session Key, который формируется в процессе работы функции и влияет на результат:

- generateKeyByEmv равен true — ICC Session Key генерируется по стандарту EMV 4.2,
- отсутствует или равен false — ICC Session Key генерируется в соответствии со стандартом текущего диалекта.

Для генерации ICC Session Key по стандарту EMV 4.2 используется только ICC Master Key, полученный из Issuer Master Key, и Transaction Counter (9F36), а для диалектов Mastercard и Mir дополнительно используется Unpredictable Number (9F37).

Таблица 8-6 Служебные параметры функции isoGenerateAC

Параметр	Источник
dialect	Из кода диалекта. Код диалекта ISO должен совпадать с именем соответствующего диалекта SOLAR ISO8583
version	Для диалекта VISA: третий байт тега "9F10": '0A' — 10, '0E' — 14. Для диалектов MC и MIR: пустая строка

Параметр	Источник
issuerMasterKey	Секретный ключ Issuer Master Key для формирования главного ключа ICC Master Key
terminalData	Объединение значений тегов поля 55: "9F02", "9F03", "9F1A", "95", "5F2A", "9A", "9C", "9F37"
iccData	Объединение значений тегов поля 55: "82", "9F36", "9F10". Для VISA версий 10, 14 и 18 из тега "9F10" берется только CVR (с 4 по 7 байт включительно), VISA 17 версии — с 5 по 6 байт

В диалекте VISA данные чипа и терминала иногда передаются не в тегах поля 55, а в отдельных трехзначных полях. Если в сообщении отсутствуют теги поля 55, функция isoGenerateAC извлекает данные чипа из отдельных полей. Информацию о соответствии тегов поля 55 и отдельных полей можно найти в спецификации платежной системы VISA. В примере 8-2 показаны значение параметра issuerMasterKey и содержимое полей ISO сообщения.

Пример 8-2 Использование функции isoGenerateAC

```
issuerMasterKey: 0123456789ABCDEF0123456789ABCDEF
DE02: 0123456789ABCDEF0123456789ABCDEF
DE23: 00
DE55:
01005E9F34034203009F3303E0F8C8950500800480009F37046E56718C9F101206010A03A0201A0A
020000000000321B8DB39F2608${isoGenerateAC('0123456789ABCDEF0123456789ABCDEF')}
9F3602000282027C009C01009F1A0206439A031510229F02060000000366005F2A020643
DE137: 0001
```

Поле 55 будет обновлено с учетом результата вызова функции isoGenerateAC, как показано в примере ниже:

```
DE55:
01005E9F34034203009F3303E0F8C8950500800480009F37046E56718C9F101206010A03A0201A0A
020000000000321B8DB39F260801F2577A6C2645D5
9F3602000282027C009C01009F1A0206439A031510229F02060000000366005F2A020643
```

8.4 Функция svcs3

Функция svcs3 предназначена для вычисления кода svcs3. Параметры функции описаны в таблице 8-7.

Таблица 8-7 Параметры функции svcs3

Параметр	Описание
ivsv3	Инициализационный вектор. Шестнадцатеричное значение длиной 2 байта, например 0001
unpredictableNumber	Числовое значение для повышения криптографической стойкости шифрования. Шестнадцатеричное значение длиной 4 байта, например 00000000

Параметр	Описание
transactionCounter	Значение счетчика транзакций АТС (Application Transaction Counter). Шестнадцатеричное значение длиной 2 байта, например 0001
encryptionKey	Ключ шифрования

Вызов функции `svc3` с параметрами, указанными в примере 8-3, дает результат 9089.

Пример 8-3 Использование функции `svc3`

```
#{svc3('0D14', '00000899', '005E', '462FC416E0E93D042CD0B00731AB4637')}
```

8.5 Функция `encryptClearPinBlockIso0`

Функция `encryptClearPinBlockIso0` предназначена для шифрования пин-блока. Параметры функции описаны в таблице 8-8.

Таблица 8-8 Параметры функции `encryptClearPinBlockIso0`

Параметр	Описание
pin	Персональный идентификационный номер (Pin). 4-12 цифр
pan	Номер карты (PAN). По крайней мере 14 шестнадцатеричных символов, последний из которых — проверочный
paddingCharacter	Шестнадцатеричный символ-заполнитель (0-9, A-F). Обычно используется F
encryptionKey	Ключ шифрования для алгоритма 3DES (48 шестнадцатеричных символов)

9 Нагрузочное тестирование

9.1 Основные понятия

Модуль нагрузочного тестирования предназначен для оценки производительности системы с помощью запуска нагрузочных проектов и анализа получаемых данных.

По сравнению с проектом, предназначенным для функционального тестирования, нагрузочный проект имеет ряд особенностей:

- Проект выполняется по этапам.
- Продолжительность выполнения проекта определяется суммарной длительностью этапов.
- Все сценарии выполняются одновременно и многократно на протяжении активного этапа.
- По завершении этапа происходит переход к следующему, а при его отсутствии проект завершается.

Этап нагрузки — период времени, в течение которого выполняются все сценарии проекта, и с заданной периодичностью сохраняются промежуточные результаты.

По завершении проекта на основе промежуточных результатов формируется итоговая статистика.

Начальные шаги или шаги инициализации — шаги нагрузочного сценария, выполняемые однократно в начале проекта.

Финальные шаги или шаги финализации — шаги нагрузочного сценария, выполняемые однократно в конце проекта после выполнения основных шагов.

В отличие от функционального сценария, нагрузочный имеет следующие особенности:

- Отсутствуют сущности, относящиеся к области функционального тестирования: функциональные области, модули, требования, приоритеты.
- Общее количество шагов сценария не должно превышать 5, включая шаги шаблонов.
- В шагах должны использоваться только простые проверки. Категорически не рекомендуется использовать табличные сравнения и регулярные выражения.
- Для каждого сценария должен быть задан его профиль нагрузки.

Профиль нагрузки проекта в каждый момент времени состоит из профилей нагрузки сценариев, выполняемых в рамках активного этапа.

Нагрузочный проект выполняется в следующем порядке:

1. В начале этапа последовательно выполняются шаги инициализации всех сценариев. При возникновении ошибки хотя бы в одном сценарии выполнение проекта завершается. При успешном завершении фазы инициализации происходит переход к основным шагам.
2. Основные шаги повторяются многократно вплоть до прерывания или завершения этапа.
3. Выполняются шаги финализации.
4. Если этап не был прерван вручную и он не является последним, происходит переход к следующему этапу.

Для использования в нагрузочном тестировании готовых тестовых данных, например, номеров карт или терминалов, рекомендуется ввести эти данные в систему в виде наборов параметров, а затем привязать требуемый набор параметров к проекту.

Для ввода наборов параметров в систему используйте:

- интерфейс приложения (см. раздел 4.6.1 «Создание набора параметров» и 4.6.2 «Загрузка набора параметров из базы данных»)
- файловый интерфейс `saveParameterSetRowBatch` (см. Приложение В «Интерфейс `saveParameterSetRowBatch`»).

Для переноса нагрузочных проектов в другие системы используйте функции экспорта и импорта проекта (см. разделы 5.1.2 «Экспорт проекта» и 5.1.3 «Импорт проекта»)

Для быстрого создания нового нагрузочного проекта на основе существующего используйте функцию копирования проекта (см. раздел 9.7).

9.2 Настройка профиля нагрузки

Чтобы настроить профиль нагрузки:

1. Создайте нагрузочный проект (см. раздел 9.2.1).
2. (Опционально) Привяжите набор параметров (см. раздел 9.2.2).
3. Добавьте один или несколько нагрузочных сценариев (см. раздел 9.2.4).

9.2.1 Создание нагрузочного проекта

Чтобы создать нагрузочный проект:

1. Откройте форму **Нагрузочные проекты**, доступную через меню **Нагрузочное тестирование** → **Нагрузочные проекты**.
2. Добавьте запись в форму и укажите следующие параметры:
 - **Код**,
 - **Наименование**,
 - **Описание**,
 - Добавьте один или несколько этапов и укажите следующие параметры:
 - **№** — порядковый номер этапа,
 - **Наименование**,
 - **Ограничение длительности** — период времени в секундах, в течение которого продолжается этап;
 - **Интервал обновления** — период времени в секундах, за который сохраняются промежуточные результаты.
3. Нажмите **Сохранить**.

9.2.2 Привязка набора параметров

Чтобы привязать к проекту набор параметров, используйте стандартную процедуру, описанную в разделе 5.2, в которой вместо выбора конкретной строки укажите метод, при помощи которого при каждом запуске сценария, использующего набор параметров, выбирается строка:

- **Последовательный** — строка выбирается по порядку циклически,
- **Случайный** — строка выбирается случайным образом.

9.2.3 Нагрузочный сценарий

Основное назначение нагрузочного сценария — обеспечение нагрузочного воздействия на тестируемую систему. С этой целью для каждого сценария необходимо задать профили

нагрузки для каждого этапа проекта. Используя настройки профиля, можно задать нагрузочное воздействие на систему, увеличивающееся с каждым этапом.

Обязательные параметры профиля нагрузки — ограничение по количеству операций в секунду и тип перехода к текущему этапу (см. таблицу 9-1).

Если указать количество повторов сценария, то по достижении указанного количества повторов сценарий завершится, даже если этап продолжается.

Сценарий содержит следующие виды шагов:

1. Начальные шаги.
2. Основные шаги.
3. Финальные шаги.

Использование наборов параметров возможно только на основных шагах.

9.2.4 Создание нагрузочного сценария

Чтобы добавить в проект нагрузочный сценарий:


1. Перейдите на страницу проекта по ссылке в колонке **Наименование**.
2. Добавьте запись в форму **Нагрузочные сценарии** и укажите следующие параметры:
 - **Код**,
 - **Наименование**,
 - **Действие**,
 - **Описание**.
3. Нажмите **Сохранить**.
4. Перейдите на страницу сценария по ссылке в колонке **Наименование** и нажмите кнопку  (**Редактировать**).
5. Укажите параметры профиля нагрузки сценария для каждого этапа проекта (см. таблицу 9-1):
 - (Опционально) **Ограничение повторов**,
 - **Операции в секунду**,
 - **Тип перехода**,
 - (Опционально) **Потоки**.
6. В форму **Тестовые шаги** добавьте необходимое количество тестовых шагов (см. раздел 5.4).

Таблица 9-1 Параметры профиля нагрузки сценария

Параметр	Описание
Ограничение повторов	Максимальное количество повторов сценария
Операции в секунду	Ограничение на количество операций, выполняемых в секунду
Тип перехода	Параметр определяет метод выбора количества операций в секунду и количества потоков при переходе к текущему этапу: <ul style="list-style-type: none"> • Ступенчатый — фиксированное количество, • Линейный — постепенно изменяемое количество от значения предыдущего этапа (0 операций, если это первый этап) до заданного в текущем профиле нагрузки

Параметр	Описание
Потоки	Количество потоков канала приложения. Рекомендуется использовать для каналов с отдельным подключением на каждый запрос, например, POS, SOAP и т.п. Позволяет эмулировать работу канала на предельных режимах и количестве подключенных устройств

9.3 Проверки метрик производительности

В оценке производительности используются два вида метрик:

- Количество операций в секунду (TPS),
- Время выполнения операции.

Проверки метрик производительности выполняются в конце этапа. Проверки настраиваются для каждого этапа отдельно (см. раздел 9.3.1). Если проверка не прошла, в форме **Этапы** отображается статус «Не пройден».

Используя проверки метрик производительности, вы можете организовать автоматическое нагрузочно-регрессионное тестирование, которое в ряде случаев позволит избавиться от необходимости ручного анализа результатов.

Для оценки метрик производительности используйте переменные, описанные в таблице 9-2.

Таблица 9-2 Переменные для использования в проверках производительности на этапах

Переменная	Описание
runtime/stage/execution/minimum_rate	Минимальное количество операций в секунду
runtime/stage/execution/maximum_rate	Максимальное количество операций в секунду
runtime/stage/execution/mean_rate	Среднее количество операций в секунду за время тестирования
runtime/stage/execution/median_rate	Медиана количества операций в секунду — такое количество операций, что половина значений метрики на этапе меньше данного значения, а другая половина — больше
runtime/stage/execution/quantile_rate_1	Квантиль №1 количества операций в секунду — количество операций в секунду, которое не превышает значениями метрики на этапе с вероятностью, заданной параметром этапа «Квантиль №1»

Переменная	Описание
runtime/stage/execution/quantile_rate_2	Квантиль №2 количества операций в секунду — количество операций в секунду, которое не превышает значения метрики на этапе с вероятностью, заданной параметром этапа «Квантиль №2»
runtime/stage/execution/minimum_duration	Минимальное время выполнения
runtime/stage/execution/maximum_duration	Максимальное время выполнения
runtime/stage/execution/mean_duration	Среднее время выполнения за время тестирования
runtime/stage/execution/median_duration	Медиана времени выполнения — такое время выполнения операции, что половина значений метрики на этапе меньше данного значения, а другая половина — больше
runtime/stage/execution/quantile_duration_1	Квантиль №1 времени выполнения — время выполнения операции, которое не превышает значения метрики на этапе с вероятностью, заданной параметром этапа «Квантиль №1»
runtime/stage/execution/quantile_duration_2	Квантиль №2 времени выполнения — время выполнения операции, которое не превышает значения метрики на этапе с вероятностью, заданной параметром этапа «Квантиль №2»

9.3.1 Настройка проверок метрик производительности

Чтобы настроить проверку метрики производительности:

1. Откройте форму **Нагрузочные проекты**, доступную через меню **Нагрузочное тестирование** → **Нагрузочные проекты**.
2. Перейдите на страницу проекта по ссылке в колонке **Наименование**.
3. В форме **Этапы** перейдите на страницу этапа по ссылке в колонке **Наименование**.
4. Добавьте запись в форму **Проверки**, используя процедуру, описанную в разделе 5.4.7.
5. Чтобы удалить ненужную проверку, нажмите кнопку **✕** в правой части строки с ее данными.

9.3.2 Пример проверки метрики «Минимальное количество операций в секунду»

После обновления программного обеспечения рекомендуется проводить нагрузочное тестирование, чтобы убедиться, что производительность системы не ухудшилась или деградация системы находится в допустимых пределах.

Например, если значение по результатам теста уменьшилось более чем на 3%, считать деградацию производительности недопустимой и выдавать ошибку. Предположим, 25 операций в секунду является эталонным значением.

Для настройки такой проверки укажите следующие данные:

- **Источник** — Выражение,
- **Выражение** — $\{100*\text{divide}(25\text{-runtime}/\text{stage}/\text{execution}/\text{minimum_rate}),25\}$, где 25 — эталонное значение;
- **Проверка** — Меньше или равно,
- **Ожидаемое значение** — 3.

Если значение выражения больше 3, данная проверка завершается с ошибкой.

9.4 Просмотр результатов выполнения

После завершения проекта и сохранения промежуточных результатов формируется итоговая статистика:

- для проекта,
- для каждого этапа проекта,
- для каждого сценария, выполненного на этапе;
- для результатов всех проверок.

Статистика этапа содержит суммарные результаты сценариев, статистика проекта — результаты этапов.

Статистика сценария содержит также данные всех проверок — успешных и не пройденных — с разделением по типам ошибок и видам некорректных результатов. Дополнительно указывается канал, в котором произошла ошибка.

Пример с итоговой статистикой для проекта изображен на рисунке 107.

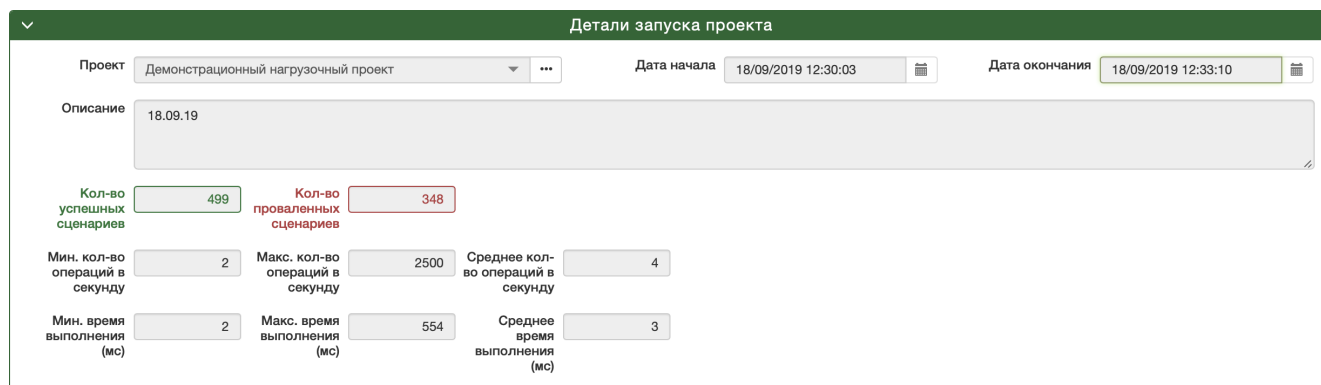


Рисунок 107 Итоговая статистика для проекта

Состав статистических данных описан в таблице 9-3.

Таблица 9-3 Состав статистических данных

Параметр	Описание
Количество успешных сценариев	Количество запусков сценариев, завершившихся успешно

Параметр	Описание
Количество проваленных сценариев	Количество запусков сценариев, завершившихся с ошибкой
Мин. кол-во операций в секунду	Минимальное количество операций в секунду
Макс. кол-во операций в секунду	Максимальное количество операций в секунду
Среднее кол-во операций в секунду	Среднее количество операций в секунду за время тестирования
Мин. время выполнения (мс)	Минимальное время выполнения одной операции в миллисекундах
Макс. время выполнения (мс)	Максимальное время выполнения одной операции в миллисекундах
Среднее время выполнения (мс)	Среднее время выполнения одной операции в миллисекундах за время тестирования

Чтобы просмотреть итоговую статистику:

1. Откройте форму **Нагрузочные проекты** и перейдите на страницу проекта по ссылке в колонке **Наименование**.
2. В форме **Запуски проекта** перейдите по ссылке в колонке **Описание**: статистика для проекта отображается в форме **Детали запуска проекта**.
3. В форме **Этапы** перейдите по ссылке в колонке **Этап**: статистика для этапа отображается в форме **Детали запуска этапа**.
4. В форме **Сценарии** на странице запуска этапа перейдите по ссылке в колонке **Сценарий**: статистика для сценария на данном этапе отображается в форме **Детали запуска сценария**. Статистика выполнения шагов сценария отображается в форме **Тестовые шаги**.
 - Для шагов, на которых в тестируемую систему отправляется сообщение, отображается среднее, минимальное и максимальное время ответа от системы.

9.5 Динамическое управление нагрузкой

Нагрузкой на систему можно управлять динамически во время выполнения проекта.

Чтобы изменить нагрузку:

1. Откройте форму **Нагрузочные проекты** и перейдите на страницу проекта по ссылке в колонке **Наименование**.
2. В форме **Запуски проекта** нажмите **Изменить конфигурацию** в правой части строки. Появится окно **Изменить конфигурацию**, изображенное на рисунке 108.



Рисунок 108 Динамическое изменение нагрузки

- Для каждого сценария выберите тип перехода и укажите новое ограничение скорости.
- Нажмите **Изменить**.

После сохранения новых настроек проект начинает работать с изменившейся нагрузкой.

9.6 Отчет о выполнении проекта

При наличии в приложении модуля SOLAR Reporting можно сгенерировать отчет в формате XLSX, содержащий статистические данные нагрузочного тестирования, сохраненные с заданной периодичностью.

Чтобы сформировать отчет:

- В форме **Запуска проекта** перейдите по ссылке в колонке **Описание** на страницу с результатами запуска проекта.
- В форме **Действия** нажмите кнопку **Отчет о выполнении**.
- В окне **Результат выполнения** нажмите **Заккрыть**.

Файл с именем performance_statistics по умолчанию сохраняется в папке, выбранной для загрузки файлов.

9.7 Копирование нагрузочного проекта

Копирование нагрузочного проекта позволяет быстро создать новый проект на основе существующего: создайте копию проекта и модифицируйте ее.

Чтобы создать копию нагрузочного проекта, используйте процедуру, описанную для функционального проекта в разделе 5.1.1.

Если в настройках копирования не указать новый код и наименование проекта, то к коду и наименованию нового проекта добавляется суффикс, например PRF_Test_1 (1) (см. рис. 109).

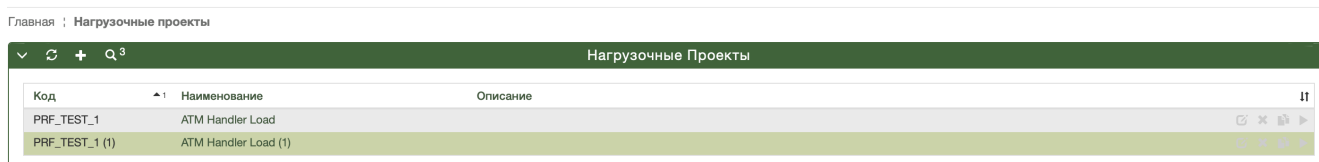


Рисунок 109 Нагрузочные проекты

Новый проект содержит все этапы первоначального и доступен для редактирования. Вы можете добавить новые и изменить существующие этапы и сценарии.

10 Compliance-тестирование процессинговой системы

Платежные системы устанавливают собственные требования к формированию и обработке операций.

Compliance-тестирование предназначено для подтверждения соответствия процессинговой системы требованиям платежных систем.

Рекомендуется проводить Compliance-тестирование как перед первым введением хоста в промышленную эксплуатацию, так и в рамках тестирования релизов платежных систем.

Для каждой платежной системы создайте отдельный проект с набором сценариев:

1. Тестовые сценарии, покрывающие формирование и обработку операций.
2. Базовые проверки корректности форматов (длина полей, формат, обязательность).
3. Дополнительные (условные) проверки, специфичные для тестируемой процессинговой системы.

Поскольку базовые проверки и большая часть условных многократно используются в разных сценариях, рекомендуется настроить их в виде шаблонов.

11 Поддержка протокола NDC

Для тестирования системы с помощью протокола Advance NDC в конфигурации приложения должен присутствовать адаптер ATM Client.

Адаптер ATM Client предоставляет следующие возможности:

- эмуляция банкоматов — поштучно или пулами,
- хранение информации о состоянии банкомата и входящих в него устройств,
- генерация сообщений: Solicited Messages, Unsolicited Messages;
- эмуляция Transaction Request,
- обработка реакции хоста (Host Command).

Для всех сообщений используется диалект JSON.

Поддерживаются два режима работы банкоматов:

- EMULATED (по умолчанию) используется для функционального тестирования транзакций, т.е. взаимодействия банкомата с хостом для проведения транзакций. В данном режиме приложение должно инициировать и подтверждать транзакции. Для остальных видов взаимодействия применяется эталонное поведение эмулятора.
- MANUAL используется для любого взаимодействия с хостом, например, для обновления значений счетчиков или для перехода в режим администратора, для отправки Send Fitness Statuses Command. В данном режиме приложение должно отвечать на все запросы от хоста.

11.1 Настройка проекта тестирования

Для создания проектов и тестовых сценариев NDC используйте стандартные процедуры, описанные в данном документе.

Чтобы настроить проект:

1. Создайте канал, соответствующий тестируемой системе, если он еще не создан (см. раздел 4.1). В качестве диалекта выберите JSON.
2. Создайте проект (см. раздел 5.1).
3. Добавьте один или несколько сценариев (см. раздел 5.3.3).
4. Добавьте тестовые шаги (см. разделы 5.4.1 и 11.2).

11.2 Данные шага для преобразования в формат NDC

Сообщения в формате NDC, используемые для взаимодействия с процессинговой системой при выполнении тестовых шагов, получают преобразованием формата JSON, который указан в секции **Данные шага** и содержит поля, описанные в таблице 11-1.

Таблица 11-1 Состав полей сообщения JSON

Параметр	Описание
responseType	Тип ответного сообщения
responseMessage	Текст ответного сообщения
luno	Уникальный номер банкомата

Параметр	Описание
command	Наименование команды. Перечень поддерживаемых команд NDC содержится в разделе 11.3, управляющих команд — в разделе 11.4
params	Набор параметров команды, соответствующих спецификации протокола NDC
rawFields	Специфические поля операций, задаваемые парами "<номер поля>": "<значение поля>". В результате выполнения команды полю с заданным номером присваивается указанное значение

11.3 Перечень поддерживаемых команд NDC

Адаптер ATM Client поддерживает следующие команды NDC:

- Configuration Id Number Command,
- Date And Time Command,
- Dispenser Cassette Table Command,
- Fit Data Command,
- Screen Keyboard Data Command,
- State Tables Command,
- Ready Status Message,
- Send Configuration Id Status Message,
- Go In Service Command,
- Go Out Of Service Command,
- Send Configuration Id Command,
- Send Configuration Information Command,
- Send Supply Counters Command,
- Transaction Request Message,
- Transaction Reply Command,
- Report Cash Deposit Definition Message,
- Send Basic Supply Counters Message,
- Send Extended Supply Counters Message,
- Fitness Data Message,
- UnsolicitedStatusMessage,
- PowerFailureUnsolicitedMessage,
- InteractiveTransactionResponseCommand.

11.4 Перечень управляющих команд

В текущей версии приложения поддерживаются следующие управляющие команды:

- connect — подключить банкомат,
- disconnect — отключить банкомат,
- setMode — установить режим работы банкомата (MANUAL, EMULATED).

Момент подтверждения выполнения команды connect зависит от текущего уровня работы банкомата:

- EMULATED — эмулятор подтверждает выполнение команды connect только после перевода банкомата в режим обслуживания, т.е. когда банкомат готов для выполнения транзакций;
- MANUAL — сразу же после подключения эмулятора к хосту.

11.5 Примеры эмуляции команд

11.5.1 Пример команды Transaction Request Message для операции Cash-in

Команда Transaction Request Message отправляется с банкомата хосту и содержит данные, необходимые для авторизации хостом действия, запрошенного держателем карты.

Пример 11-1 содержит данные шага, при выполнении которого отправляется команда Transaction Request Message (Cash-in). Параметры "operCodeData": "DADD", "14": "w01008" означают, что на магнитную карту внесли 8 купюр из кассеты 1 (по 100 рублей).

Пример 11-1 Transaction Request Message: Операция Cash-in

```
{
  "luno": "000101603",
  "command": "Transaction Request Message",
  "params": {
    "topOfReceiptFlag" : "PRINT_DATA",
    "amountEntry" : "00080000",
    "track2Data": ";2200870010012200=211012016505414000?",
    "operCodeData": "DADD",
    "bufferA": "31184855=5:7:999"
  },
  "rawFields": {
    "14": "w01008"
  }
}
```

Для чиповой карты аналогичная команда содержит элемент rawFields с данными поля 15.

11.5.2 Пример команды Transaction Request Message для выдачи наличных

Пример 11-2 содержит данные шага, при выполнении которого отправляется команда Transaction Request Message, соответствующая операции выдачи наличных по магнитной карте.

Пример 11-2 Transaction Request Message: Операция Cash-out

```
{
  "luno": "000101621",
  "command": "Transaction Request Message",
  "params": {
    "topOfReceiptFlag" : "PRINT_DATA",
    "amountEntry" : "00050000",
    "track2Data": ";2200870010012200=211012016505414000?",
    "operCodeData": "BABC",
    "bufferA": "31184855=5:7:999"
  }
}
```

Для чиповой карты аналогичная команда содержит элемент rawFields с данными поля 15.

11.5.3 Ready Status Message (Transaction Reply Confirmation)

Пример 11-3 содержит сообщение JSON, которое эмулирует подтверждение, отправляемое с банкомата хосту в ответ на Transaction Reply Command.

11.5.7 Переключение между режимами работы

Для переключения режима работы банкомата используется команда `setMode`.

Чтобы установить на банкомате ручной режим, следует выполнить шаг, на котором задано сообщение, приведенное в примере 11-7.

Пример 11-7 Сообщение для включения ручного режима работы

```
{
  "luno": "${luno}",
  "command": "setMode",
  "params": {
    "solicitedHandlingMode": "MANUAL"
  }
}
```

Чтобы установить на банкомате эмулируемый режим, следует выполнить шаг, на котором задано сообщение, приведенное в примере 11-8.

Пример 11-8 Сообщение для включения эмулируемого режима работы

```
{
  "luno": "${luno}",
  "command": "setMode",
  "params": {
    "solicitedHandlingMode": "EMULATED"
  }
}
```

12 Работа с клиринговыми файлами

Клиринговый модуль приложения предоставляет следующие возможности:

- загрузка клиринговых файлов (AMEX, Mastercard, UPI, Visa, Way4, МИР),
- выгрузка клиринговых файлов (AMEX, Mastercard, Tsys Prime, UPI, Visa, Way4, МИР).

12.1 Настройка загрузки и выгрузки клиринговых файлов

12.1.1 Настройка зависимостей в файле pom.xml

Для работы с диалектами клиринга необходимо добавить соответствующие зависимости в файл *pom.xml*.

- Диалект Mastercard IPM

```
<dependency>
  <groupId>com.solanteq.solar.clearing</groupId>
  <artifactId>solar-clearing-dialect-mastercard</artifactId>
  <version>${solar.clearing.version}</version>
</dependency>
```

- Диалект VISA CTF

```
<dependency>
  <groupId>com.solanteq.solar.clearing</groupId>
  <artifactId>solar-clearing-dialect-visa</artifactId>
  <version>${solar.clearing.version}</version>
</dependency>
```

- Диалект МИР DPOE

```
<dependency>
  <groupId>com.solanteq.solar.clearing</groupId>
  <artifactId>solar-clearing-dialect-mir</artifactId>
  <version>${solar.clearing.version}</version>
</dependency>
```

- Диалект Amex

```
<dependency>
  <groupId>com.solanteq.solar.clearing</groupId>
  <artifactId>solar-clearing-dialect-amex</artifactId>
  <version>${solar.clearing.version}</version>
</dependency>
```

- Диалект UPI

```
<dependency>
  <groupId>com.solanteq.solar.clearing</groupId>
  <artifactId>solar-clearing-dialect-upi</artifactId>
  <version>${solar.clearing.version}</version>
</dependency>
```

- Диалект Way4 Base II Like

```
<dependency>
  <groupId>com.solanteq.solar.clearing</groupId>
  <artifactId>solar-clearing-dialect-w4</artifactId>
```

```
<version>${solar.clearing.version}</version>
</dependency>
```

12.1.2 Создание канала

Для загрузки и выгрузки клиринговых файлов создайте канал для каждого типа загружаемых и выгружаемых клиринговых файлов.

1. В поле **Диалект** выберите **Загрузка файла** или **Выгрузка файла**.
2. В дополнительных настройках в поле **Тип сообщения** выберите **Clearing**.
3. В поле **Формат файла** выберите диалект клиринга.
4. Для диалектов Mastercard и МИР в поле **Кодировка** выберите кодировку клирингового файла: ASCII или EBCDIC.

Подробнее о создании канала см. в разделе 4.1.1.

После создания канала выберите его в форме **Каналы** и в дополнительных настройках укажите для каждого окружения базовую директорию (см. рисунок 110).

☆ Каналы / Clearing: mastercard / Default Environment: Test 🔗

The screenshot shows a configuration interface for a channel. It has two main sections: 'Детали' (Details) and 'Дополнительные настройки' (Additional settings). In the 'Детали' section, there is a 'Шаблон' (Template) field with the value 'Clearing: mastercard'. In the 'Дополнительные настройки' section, there is a 'Базовая директория' (Base directory) field with the value '/opt/nfs/ref-acq/test/clearing'.

Рисунок 110 Настройка базовой директории для окружения

Адаптер загружает файлы из директории `<BASE_DIR>/incoming` и выгружает файлы в директорию `<BASE_DIR>`.

Обратите внимание: SOLAR Testing Toolkit загружает клиринговый файл в формате VISA outgoing CTF и выгружает файл в формате VISA incoming CTF.

12.2 Структура клирингового файла Mastercard

Для выгрузки клирингового файла Mastercard в процессинговую систему в приложении необходимо подготовить файл со следующей структурой (см. пример 12-1).

Файл содержит один или несколько элементов `<row>`, соответствующих одной операции. Операция состоит из простых и составных полей (см. раздел 12.2.1).

Пример 12-1 Структура клирингового файла Mastercard

```
<?xml version="1.0" encoding="UTF-8"?>
<file>
  <row>
    <_0>1644</_0>
    <_24>697</_24>
    <_48>
      <_122>P</_122>
      <_105>0022107130000001193736036</_105>
    </_48>
    <_71>00000001</_71>
```

```
</row>
<row>
  <_0>1240</_0>
  <_2>5447148423093274</_2>
  <_3>200000</_3>
  <_4>000000199951</_4>
  <_12>210713180238</_12>
  <_14>2312</_14>
  <_22>M10101C11124</_22>
  <_23>001</_23>
  <_24>200</_24>
  <_26>5411</_26>
  <_31>75442181194164008847408</_31>
  <_33>00000011937</_33>
  <_37>119421149684</_37>
  <_38>896451</_38>
  <_40>201</_40>
  <_41>40000543</_41>
  <_42>40000543</_42>
  <_43>V.P. Leont`ev\pr-d Pobedi,3\pos.Verbilki\141930      RUSRUS</_43>
  <_48>
    <_148>6432</_148>
    <_165>M                                     </_165>
    <_158>                                     L1</_158>
    <_23>POI</_23>
  </_48>
  <_49>643</_49>
  <_55>
    <_84>A0000000041010</_84>
    <_9F33>6008C8</_9F33>
    <_9F34>010002</_9F34>
    <_9F1A>0643</_9F1A>
    <_82>1B80</_82>
    <_5F2A>0643</_5F2A>
    <_9F02>000000199212</_9F02>
    <_9C>00</_9C>
    <_9A>210327</_9A>
    <_95>0020008000</_95>
    <_9F36>0080</_9F36>
    <_9F37>A3AB28AB</_9F37>
    <_9F10>0314A541000200000000006674A2171E00FF</_9F10>
    <_9F27>80</_9F27>
    <_9F26>846887025EF7CEFD</_9F26>
  </_55>
  <_63> MNW0985430713  </_63>
  <_71>00000002</_71>
  <_94>00000011937</_94>
</row>
<row>
  <_0>1644</_0>
  <_24>695</_24>
  <_48>
    <_306>00000000</_306>
    <_301>0000000000000000</_301>
    <_105>0022107130000001193736036</_105>
```

```
</_48>
<_71>00000003</_71>
</row>
</file>
```

12.2.1 Поля клирингового диалекта Mastercard

В таблице 12-1 перечислены поля клирингового диалекта Mastercard IPM, используемые для описания операции в клиринговом файле.

Тип значения COMP означает, что поле является составным и состоит из нескольких полей.

Тип значений EBCDIC используется по умолчанию. Использование кодировки ASCII возможно только после настройки конфигурации батч адаптера с кодировкой ASCII.

Подробнее о значении полей см. в спецификации Mastercard «IPM Clearing Formats».

Таблица 12-1 Состав полей операции в клиринговом диалекте Mastercard

Код поля	Тип значения	Наименование
0	EBCDIC	Message type indicator (MTI)
2	EBCDIC	Primary Account Number (PAN)
3	EBCDIC	Processing Code
4	EBCDIC	Amount, Transaction
5	EBCDIC	Amount, Reconciliation
6	EBCDIC	Amount, Cardholder Billing
9	EBCDIC	Conversion Rate, Reconciliation
10	EBCDIC	Conversion Rate, Cardholder Billing
12	EBCDIC	Date and Time, Local Transaction
14	EBCDIC	Date, Expiration
22	EBCDIC	Point-of-Service (POS) Data Code
23	EBCDIC	Card Sequence Number
24	EBCDIC	Function Code
25	EBCDIC	Message Reason Code
26	EBCDIC	Card Acceptor Business Code (MCC)
30	EBCDIC	Amounts, Original
31	EBCDIC	Acquirer Reference Data
32	EBCDIC	Acquiring Institution ID Code
33	EBCDIC	Forwarding Institution ID Code
37	EBCDIC	Retrieval Reference Number

Код поля	Тип значения	Наименование
38	EBCDIC	Approval Code
40	EBCDIC	Service Code
41	EBCDIC	Card Acceptor Terminal ID
42	EBCDIC	Card Acceptor ID Code
43	EBCDIC	Card Acceptor Name/Location
47	EBCDIC	Additional Data—National Use
48	COMP	Additional Data—Private Use
49	EBCDIC	Currency Code, Transaction
50	EBCDIC	Currency Code, Reconciliation
51	EBCDIC	Currency Code, Cardholder Billing
52	EBCDIC	Currency Code, Cardholder Billing
53	EBCDIC	Security-Related Control Information
54	EBCDIC	Amounts, Additional
55	BIN	Integrated Circuit Card (ICC) System-Related Data
62	COMP	Additional Data 2
63	EBCDIC	Transaction Life Cycle ID
71	EBCDIC	Message Number
72	EBCDIC	Data Record
73	EBCDIC	Date, Action
93	EBCDIC	Transaction Destination Institution ID Code
94	EBCDIC	Transaction Originator Institution ID Code
95	EBCDIC	Card Issuer Reference Data
100	EBCDIC	Receiving Institution ID Code
123	COMP	Additional Data 3
124	COMP	Additional Data 4
125	COMP	Additional Data 5

12.3 Структура клирингового файла МИР

Для выгрузки клирингового файла МИР в процессинговую систему в приложении необходимо подготовить файл со следующей структурой (см. пример 12-2).

Файл содержит один или несколько элементов <row>, соответствующих одной операции. Операция состоит из простых и составных полей (см. раздел 12.3.1).

Пример 12-2 Структура клирингового файла МИР

```
<?xml version="1.0" encoding="UTF-8"?><file>
  <row>
    <_0>1644</_0>
    <_2>2222200000000108</_2>
    <_24>697</_24>
    <_38>720372</_38>
    <_48>
      <_122>P</_122>
      <_105>0022205190000001612000519</_105>
    </_48>
    <_63>28277537</_63>
    <_71>00000001</_71>
  </row>
</file>
```

12.3.1 Поля клирингового диалекта МИР

В таблице 12-2 перечислены поля клирингового диалекта МИР, используемые для описания операции в клиринговом файле. Тип значения COMP означает, что поле является составным и состоит из нескольких полей.

Подробнее о значении полей см. в документе «Клиринг ОПКЦ НСПК. Обмен Документами и Платежными Требованиями (ОДПТ) ПС.12.32.1-22.00.00 Версия 1.8»

Таблица 12-2 Составляющие поля операции клирингового диалекта МИР

Код поля	Тип значения	Наименование
0	EBCDIC	Message type indicator (MTI)
2	EBCDIC	Primary Account Number (PAN)
3	EBCDIC	Processing Code
4	EBCDIC	Amount, Transaction
5	EBCDIC	Amount, Reconciliation
6	EBCDIC	Amount, Cardholder Billing
9	EBCDIC	Conversion Rate, Reconciliation
10	EBCDIC	Conversion Rate, Cardholder Billing
12	EBCDIC	Date and Time, Local Transaction
14	EBCDIC	Date, Expiration
22	EBCDIC	Point-of-Service (POS) Data Code
23	EBCDIC	Card Sequence Number
24	EBCDIC	Function Code

Код поля	Тип значения	Наименование
25	EBCDIC	Message Reason Code
26	EBCDIC	Card Acceptor Business Code (MCC)
30	EBCDIC	Amounts, Original
31	EBCDIC	Acquirer Reference Data
32	EBCDIC	Acquiring Institution ID Code
33	EBCDIC	Forwarding Institution ID Code
37	EBCDIC	Retrieval Reference Number
38	EBCDIC	Approval Code
40	EBCDIC	Service Code
41	EBCDIC	Card Acceptor Terminal ID
42	EBCDIC	Card Acceptor ID Code
43	EBCDIC	Card Acceptor Name/Location
48	COMP	Additional Data
49	EBCDIC	Currency Code, Transaction
50	EBCDIC	Currency Code, Reconciliation
51	EBCDIC	Currency Code, Cardholder Billing
52	EBCDIC	Currency Code, Cardholder Billing
53	EBCDIC	Security-Related Control Information
54	EBCDIC	Amounts, Additional
55	BIN	Integrated Circuit Card (ICC) System-Related Data
62	COMP	Additional Data 2
63	EBCDIC	Transaction Life Cycle ID
71	EBCDIC	Message Number
72	EBCDIC	Data Record
73	EBCDIC	Date, Action
93	EBCDIC	Transaction Destination Institution ID Code
94	EBCDIC	Transaction Originator Institution ID Code
95	EBCDIC	Card Issuer Reference Data

Код поля	Тип значения	Наименование
100	EBCDIC	Receiving Institution ID Code
123	COMP	Additional Data 3
124	COMP	Additional Data 4
125	COMP	Additional Data 5

12.4 Структура клирингового файла Visa

Клиринговый файл состоит из заголовка (опционально), одной или нескольких записей, соответствующих операциям, и завершителя файла.

Чтобы выгрузить клиринговый файл Visa в процессинговую систему, в приложении необходимо подготовить файл со следующей структурой (см. пример 12-3). Обратите внимание на присутствующие в тегах пробелы.

Пример 12-3 Структура клирингового файла Visa

```
<?xml version="1.0" encoding="UTF-8"?><root>
<file>
  <row>
    <TC>90</TC>
    <Processing_BIN>971354</Processing_BIN>
    <Processing_Date>15353</Processing_Date>
    <Reserved1> </Reserved1>
    <Test_Option> </Test_Option>
    <Reserved2> </Reserved2>
    <Security_Code> </Security_Code>
    <Reserved3> </Reserved3>
    <Outgoing_File_ID>000</Outgoing_File_ID>
    <Reserved4> </Reserved4>
  </row>
  <row>
    <TC>07</TC>
    <Transaction_Code_Qualifier>0</Transaction_Code_Qualifier>
    <Transaction_Component_Sequence_Number>1</Transaction_Component_Sequence_
Number>
    <Business_Format_Code> </Business_Format_Code>
    <Token_Assurance_Level> </Token_Assurance_Level>
    <Reserved1> </Reserved1>
    <Chargeback_Reference_Number>000000</Chargeback_Reference_Number>
    <Documentation_Indicator> </Documentation_Indicator>
    <Member_Message_Text>
</Member_Message_Text>
    <Special_Condition_Indicators> </Special_Condition_Indicators>
    <Fee_Program_Indicator> </Fee_Program_Indicator>
    <Issuer_Charge>S</Issuer_Charge>
    <Reserved2> </Reserved2>
    <Card_Acceptor_ID>535074701178068</Card_Acceptor_ID>
    <Terminal_ID> </Terminal_ID>
    <National_Reimbursement_Fee>000000000000</National_Reimbursement_Fee>
    <Electronic_Commerce_Indicator> </Electronic_Commerce_Indicator>
    <Special_Chargeback_Indicator> </Special_Chargeback_Indicator>
```

```
<Interface_Trace_Number>000000</Interface_Trace_Number>
<Acceptance_Terminal_Indicator> </Acceptance_Terminal_Indicator>
<Prepaid_Card_Indicator> </Prepaid_Card_Indicator>
<Service_Development_Field>0</Service_Development_Field>
<AVS_Response_Code> </AVS_Response_Code>
<Authorization_Source_Code> </Authorization_Source_Code>
<Purchase_Identifier_Format> </Purchase_Identifier_Format>
<Account_Selection>0</Account_Selection>
<Installment_Payment_Count> </Installment_Payment_Count>
<Purchase_Identifier> </Purchase_Identifier>
<Cashback>000000000</Cashback>
<Chip_Condition_Code>1</Chip_Condition_Code>
<POS_Environment> </POS_Environment>
</row>
...
</file>
```

12.4.1 Поля клирингового диалекта Visa

Поля клирингового диалекта Visa соответствуют параметрам спецификации «VISA BASE II Clearing Interchange Formats».

Для параметра Transaction Code используется тег TC. Для остальных параметров название тега получается из названия соответствующего параметра спецификации, в котором пробелы заменены на символ подчеркивания.

Например, тег Transaction_Component_Sequence_Number соответствует параметру Transaction Component Sequence Number, который еще называется TCR (Transaction Component Record).

Чтобы добавить операцию в клиринговый файл, необходимо указать данные всех тегов, которые указаны в спецификации формата STF Incoming для операции данного типа, т.е. операции с заданным TC и TCR.

На входящих шагах, работающих с клиринговыми файлами, загружаемыми из процессинговой системы, для создания проверок следует применить аналогичную процедуру, а именно использовать названия тегов, соответствующих параметрам спецификации формата STF Outgoing для операции с заданным TC и TCR.

12.5 Структура клирингового файла UPI

Клиринговый файл состоит из заголовка (опционально), одной или нескольких записей, соответствующих операциям, и завершителя файла.

Чтобы выгрузить клиринговый файл UPI в процессинговую систему, в приложении необходимо подготовить файл со следующей структурой (см. пример 12-4). Обратите внимание на присутствующие в тегах пробелы.

Пример 12-4 Структура клирингового файла UPI

```
<?xml version="1.0" encoding="UTF-8"?><file>
  <row>
    <TC>000</TC>
    <Block_Bitmap>8000</Block_Bitmap>
    <IIN>34190643 </IIN>
    <Batch_Date>20170202</Batch_Date>
    <Settlement_Date_GSCS> </Settlement_Date_GSCS>
    <Version_Tag>PROD</Version_Tag>
```

```
<Version_Number>00000001</Version_Number>
</row>
<row>
  <TC>100</TC>
  <Block_Bitmap>C000</Block_Bitmap>
  <PAN>6250942000000012 </PAN>
  <Amount_Transaction>00000005000</Amount_Transaction>
  <Currency_Code_Transaction>643</Currency_Code_Transaction>
  <Transaction_Date_Time>0112221513</Transaction_Date_Time>
  <System_Trace_Audit_Number>004491</System_Trace_Audit_Number>
  <Authorization_Identification_Response>003308</Authorization_
Identification_Response>
  <Authorization_Date>0202</Authorization_Date>
  <Retrieval_Reference_Number>701217004492</Retrieval_Reference_Number>
  <Acquiring_IIN>34190643 </Acquiring_IIN>
  <Forwarding_IIN>34190643 </Forwarding_IIN>
  <Merchant_Type>5992</Merchant_Type>
  <Card_Acceptor_Terminal_Identification>811794 </Card_Acceptor_Terminal_
Identification>
  <Card_Acceptor_Identification_Code>230000014016 </Card_Acceptor_
Identification_Code>
  <Card_Acceptor_Name_Location>IP KHVATOVA M G KAZAN
  <Card_Acceptor_Name_Location>
  <Original_Transaction_Information>0000000000000000000000</Original_
Transaction_Information>
  <Message_Reason_Code>0000</Message_Reason_Code>
  <Single_or_Dual_Message_Identifier>1</Single_or_Dual_Message_Identifier>
  <GSCS_Serial_Number>000000000</GSCS_Serial_Number>
  <Receiving_Institution_Identification_Code> </Receiving_
Institution_Identification_Code>
  <Issuing_Institution_Identification_Code> </Issuing_
Institution_Identification_Code>
  <Identifier_of_GSCS_Notice>0</Identifier_of_GSCS_Notice>
  <Transaction_Initiating_Channel>03</Transaction_Initiating_Channel>
  <Identifier_of_Transaction_Features> </Identifier_of_Transaction_
Features>
  <Reserved_Block0> </Reserved_Block0>
  <Reserved_for_GSCS> </Reserved_for_GSCS>
  <Installment_Payment_Terms> </Installment_Payment_Terms>
  <Stand-in_Authorization_Identifier> </Stand-in_Authorization_Identifier>
  <POS_Condition_Code> </POS_Condition_Code>
  <Merchant_Country_Code> </Merchant_Country_Code>
  <Transaction_Initiation_Method> </Transaction_Initiation_Method>
  <Authorization_Type> </Authorization_Type>
  <Card_Level> </Card_Level>
  <Code_of_Pricing_Scheme> </Code_of_Pricing_Scheme>
  <Special_Settlement_Currency> </Special_Settlement_Currency>
  <Ecommerce_Indicator> </Ecommerce_Indicator>
  <Card_Product> </Card_Product>
  <Account_Attribute> </Account_Attribute>
  <UnionPay_Standard_or_Nonstandard_Card_Indicator> </UnionPay_Standard_or_
Nonstandard_Card_Indicator>
  <B2B_Business_Type> </B2B_Business_Type>
  <B2B_Payment_Medium> </B2B_Payment_Medium>
  <Special_Pricing_Indicator> </Special_Pricing_Indicator>
```

```
<POS_Entry_Mode>901</POS_Entry_Mode>
<Floor_Limit_Identifier>0</Floor_Limit_Identifier>
<Type_of_Payment_Service_Requested>0 </Type_of_Payment_Service_Requested>
<Amount_Settlement>000000000000</Amount_Settlement>
<Currency_Code_Settlement> </Currency_Code_Settlement>
<Conversion_Rate_Settlement>00000000</Conversion_Rate_Settlement>
<Amount_Cardholder_Billing>000000000000</Amount_Cardholder_Billing>
<Currency_Code_Cardholder_Billing> </Currency_Code_Cardholder_Billing>
<Conversion_Rate_Cardholder_Billing>00000000</Conversion_Rate_Cardholder_
Billing>
<Net_Fee_Amount>D000000000000</Net_Fee_Amount>
<IRF_Billing_Currency>000</IRF_Billing_Currency>
<Exchange_Rate_from_IRF_Billing_Currency_to_Settlement_
Currency>00000000</Exchange_Rate_from_IRF_Billing_Currency_to_Settlement_
Currency>
<Abbreviation_of_International_Organization_and_Foreign_Institutions>
</Abbreviation_of_International_Organization_and_Foreign_Institutions>
<Mainland_China_Transaction_Indicator> </Mainland_China_Transaction_
Indicator>
<Amount_Transaction_Fee> </Amount_Transaction_Fee>
<QRC_Voucher_Number> </QRC_Voucher_Number>
<Reserved_Block_1> </Reserved_Block_1>
</row>
<row>
<TC>100</TC>
<Block_Bitmap>C000</Block_Bitmap>
<PAN>6210947000000014 </PAN>
<Amount_Transaction>000000005000</Amount_Transaction>
<Currency_Code_Transaction>643</Currency_Code_Transaction>
<Transaction_Date_Time>0112221513</Transaction_Date_Time>
<System_Trace_Audit_Number>004494</System_Trace_Audit_Number>
<Authorization_Identification_Response>000000</Authorization_
Identification_Response>
<Authorization_Date>0202</Authorization_Date>
<Retrieval_Reference_Number>701217004495</Retrieval_Reference_Number>
<Acquiring_IIN>34190643 </Acquiring_IIN>
<Forwarding_IIN>34190643 </Forwarding_IIN>
<Merchant_Type>5992</Merchant_Type>
<Card_Acceptor_Terminal_Identification>811794 </Card_Acceptor_Terminal_
Identification>
<Card_Acceptor_Identification_Code>230000014016 </Card_Acceptor_
Identification_Code>
<Card_Acceptor_Name_Location>IP KHVATOVA M G KAZAN
RUS</Card_Acceptor_Name_Location>
<Original_Transaction_Information>0000000000000000000000</Original_
Transaction_Information>
<Message_Reason_Code>0000</Message_Reason_Code>
<Single_or_Dual_Message_Identifier>1</Single_or_Dual_Message_Identifier>
<GSCS_Serial_Number>000000000</GSCS_Serial_Number>
<Receiving_Institution_Identification_Code> </Receiving_
Institution_Identification_Code>
<Issuing_Institution_Identification_Code> </Issuing_
Institution_Identification_Code>
<Identifier_of_GSCS_Notice>0</Identifier_of_GSCS_Notice>
<Transaction_Initiating_Channel>03</Transaction_Initiating_Channel>
```

```
<Identifier_of_Transaction_Features> </Identifier_of_Transaction_Features>
<Reserved_Block0> </Reserved_Block0>
<Reserved_for_GSCS> </Reserved_for_GSCS>
<Installment_Payment_Terms> </Installment_Payment_Terms>
<Stand-in_Authorization_Identifier> </Stand-in_Authorization_Identifier>
<POS_Condition_Code> </POS_Condition_Code>
<Merchant_Country_Code> </Merchant_Country_Code>
<Transaction_Initiation_Method> </Transaction_Initiation_Method>
<Authorization_Type> </Authorization_Type>
<Card_Level> </Card_Level>
<Code_of_Pricing_Scheme> </Code_of_Pricing_Scheme>
<Special_Settlement_Currency> </Special_Settlement_Currency>
<Ecommerce_Indicator> </Ecommerce_Indicator>
<Card_Product> </Card_Product>
<Account_Attribute> </Account_Attribute>
<UnionPay_Standard_or_Nonstandard_Card_Indicator> </UnionPay_Standard_or_Nonstandard_Card_Indicator>
<B2B_Business_Type> </B2B_Business_Type>
<B2B_Payment_Medium> </B2B_Payment_Medium>
<Special_Pricing_Indicator> </Special_Pricing_Indicator>
<POS_Entry_Mode>901</POS_Entry_Mode>
<Floor_Limit_Identifier>0</Floor_Limit_Identifier>
<Type_of_Payment_Service_Requested>0 </Type_of_Payment_Service_Requested>
<Amount_Settlement>000000000000</Amount_Settlement>
<Currency_Code_Settlement> </Currency_Code_Settlement>
<Conversion_Rate_Settlement>00000000</Conversion_Rate_Settlement>
<Amount_Cardholder_Billing>000000000000</Amount_Cardholder_Billing>
<Currency_Code_Cardholder_Billing> </Currency_Code_Cardholder_Billing>
<Conversion_Rate_Cardholder_Billing>00000000</Conversion_Rate_Cardholder_Billing>
<Net_Fee_Amount>D000000000000</Net_Fee_Amount>
<IRF_Billing_Currency>000</IRF_Billing_Currency>
<Exchange_Rate_from_IRF_Billing_Currency_to_Settlement_Currency>00000000</Exchange_Rate_from_IRF_Billing_Currency_to_Settlement_Currency>
<Abbreviation_of_International_Organization_and_Foreign_Institutions> </Abbreviation_of_International_Organization_and_Foreign_Institutions>
<Mainland_China_Transaction_Indicator> </Mainland_China_Transaction_Indicator>
<Amount_Transaction_Fee> </Amount_Transaction_Fee>
<QRC_Voucher_Number> </QRC_Voucher_Number>
<Reserved_Block_1> </Reserved_Block_1>
</row>
<row>
<TC>100</TC>
<Block_Bitmap>C000</Block_Bitmap>
<PAN>6250944000000667 </PAN>
<Amount_Transaction>000000009000</Amount_Transaction>
<Currency_Code_Transaction>643</Currency_Code_Transaction>
<Transaction_Date_Time>0112221513</Transaction_Date_Time>
<System_Trace_Audit_Number>004498</System_Trace_Audit_Number>
<Authorization_Identification_Response>003309</Authorization_Identification_Response>
<Authorization_Date>0202</Authorization_Date>
```

```
<Retrieval_Reference_Number>701217004499</Retrieval_Reference_Number>
<Acquiring_IIN>34190643 </Acquiring_IIN>
<Forwarding_IIN>34190643 </Forwarding_IIN>
<Merchant_Type>5992</Merchant_Type>
<Card_Acceptor_Terminal_Identification>811794 </Card_Acceptor_Terminal_
Identification>
<Card_Acceptor_Identification_Code>230000014016 </Card_Acceptor_
Identification_Code>
<Card_Acceptor_Name_Location>IP KHVATOVA M G KAZAN
RUS</Card_Acceptor_Name_Location>
<Original_Transaction_Information>0000000000000000000000</Original_
Transaction_Information>
<Message_Reason_Code>0000</Message_Reason_Code>
<Single_or_Dual_Message_Identifier>1</Single_or_Dual_Message_Identifier>
<GSCS_Serial_Number>000000000</GSCS_Serial_Number>
<Receiving_Institution_Identification_Code> </Receiving_
Institution_Identification_Code>
<Issuing_Institution_Identification_Code> </Issuing_
Institution_Identification_Code>
<Identifier_of_GSCS_Notice>0</Identifier_of_GSCS_Notice>
<Transaction_Initiating_Channel>03</Transaction_Initiating_Channel>
<Identifier_of_Transaction_Features> </Identifier_of_Transaction_
Features>
<Reserved_Block0> </Reserved_Block0>
<Reserved_for_GSCS> </Reserved_for_GSCS>
<Installment_Payment_Terms> </Installment_Payment_Terms>
<Stand-in_Authorization_Identifier> </Stand-in_Authorization_Identifier>
<POS_Condition_Code> </POS_Condition_Code>
<Merchant_Country_Code> </Merchant_Country_Code>
<Transaction_Initiation_Method> </Transaction_Initiation_Method>
<Authorization_Type> </Authorization_Type>
<Card_Level> </Card_Level>
<Code_of_Pricing_Scheme> </Code_of_Pricing_Scheme>
<Special_Settlement_Currency> </Special_Settlement_Currency>
<Ecommerce_Indicator> </Ecommerce_Indicator>
<Card_Product> </Card_Product>
<Account_Attribute> </Account_Attribute>
<UnionPay_Standard_or_Nonstandard_Card_Indicator> </UnionPay_Standard_or_
Nonstandard_Card_Indicator>
<B2B_Business_Type> </B2B_Business_Type>
<B2B_Payment_Medium> </B2B_Payment_Medium>
<Special_Pricing_Indicator> </Special_Pricing_Indicator>
<POS_Entry_Mode>011</POS_Entry_Mode>
<Floor_Limit_Identifier>0</Floor_Limit_Identifier>
<Type_of_Payment_Service_Requested>0 </Type_of_Payment_Service_Requested>
<Amount_Settlement>000000000000</Amount_Settlement>
<Currency_Code_Settlement> </Currency_Code_Settlement>
<Conversion_Rate_Settlement>00000000</Conversion_Rate_Settlement>
<Amount_Cardholder_Billing>000000000000</Amount_Cardholder_Billing>
<Currency_Code_Cardholder_Billing> </Currency_Code_Cardholder_Billing>
<Conversion_Rate_Cardholder_Billing>00000000</Conversion_Rate_Cardholder_
Billing>
<Net_Fee_Amount>D000000000000</Net_Fee_Amount>
<IRF_Billing_Currency>000</IRF_Billing_Currency>
<Exchange_Rate_from_IRF_Billing_Currency_to_Settlement_
```

```
Currency>00000000</Exchange_Rate_from_IRF_Billing_Currency_to_Settlement_
Currency>
  <Abbreviation_of_International_Organization_and_Foreign_Institutions>
</Abbreviation_of_International_Organization_and_Foreign_Institutions>
  <Mainland_China_Transaction_Indicator> </Mainland_China_Transaction_
Indicator>
  <Amount_Transaction_Fee> </Amount_Transaction_Fee>
  <QRC_Voucher_Number> </QRC_Voucher_Number>
  <Reserved_Block_1> </Reserved_Block_1>
</row>
<row>
  <TC>100</TC>
  <Block_Bitmap>C000</Block_Bitmap>
  <PAN>6250942000000002 </PAN>
  <Amount_Transaction>000000010000</Amount_Transaction>
  <Currency_Code_Transaction>643</Currency_Code_Transaction>
  <Transaction_Date_Time>0112221513</Transaction_Date_Time>
  <System_Trace_Audit_Number>004501</System_Trace_Audit_Number>
  <Authorization_Identification_Response>003310</Authorization_
Identification_Response>
  <Authorization_Date>0202</Authorization_Date>
  <Retrieval_Reference_Number>701217004502</Retrieval_Reference_Number>
  <Acquiring_IIN>34190643 </Acquiring_IIN>
  <Forwarding_IIN>34190643 </Forwarding_IIN>
  <Merchant_Type>5992</Merchant_Type>
  <Card_Acceptor_Terminal_Identification>811794 </Card_Acceptor_Terminal_
Identification>
  <Card_Acceptor_Identification_Code>230000014016 </Card_Acceptor_
Identification_Code>
  <Card_Acceptor_Name_Location>IP KHVATOVA M G KAZAN
RUS</Card_Acceptor_Name_Location>
  <Original_Transaction_Information>0000000000000000000000</Original_
Transaction_Information>
  <Message_Reason_Code>0000</Message_Reason_Code>
  <Single_or_Dual_Message_Identifier>1</Single_or_Dual_Message_Identifier>
  <GSCS_Serial_Number>000000000</GSCS_Serial_Number>
  <Receiving_Institution_Identification_Code> </Receiving_
Institution_Identification_Code>
  <Issuing_Institution_Identification_Code> </Issuing_
Institution_Identification_Code>
  <Identifier_of_GSCS_Notice>0</Identifier_of_GSCS_Notice>
  <Transaction_Initiating_Channel>03</Transaction_Initiating_Channel>
  <Identifier_of_Transaction_Features> </Identifier_of_Transaction_
Features>
  <Reserved_Block0> </Reserved_Block0>
  <Reserved_for_GSCS> </Reserved_for_GSCS>
  <Installment_Payment_Terms> </Installment_Payment_Terms>
  <Stand-in_Authorization_Identifier> </Stand-in_Authorization_Identifier>
  <POS_Condition_Code> </POS_Condition_Code>
  <Merchant_Country_Code> </Merchant_Country_Code>
  <Transaction_Initiation_Method> </Transaction_Initiation_Method>
  <Authorization_Type> </Authorization_Type>
  <Card_Level> </Card_Level>
  <Code_of_Pricing_Scheme> </Code_of_Pricing_Scheme>
  <Special_Settlement_Currency> </Special_Settlement_Currency>
```



```
<Ecommerce_Indicator> </Ecommerce_Indicator>
<Card_Product> </Card_Product>
<Account_Attribute> </Account_Attribute>
<UnionPay_Standard_or_Nonstandard_Card_Indicator> </UnionPay_Standard_or_
Nonstandard_Card_Indicator>
<B2B_Business_Type> </B2B_Business_Type>
<B2B_Payment_Medium> </B2B_Payment_Medium>
<Special_Pricing_Indicator> </Special_Pricing_Indicator>
<POS_Entry_Mode>901</POS_Entry_Mode>
<Floor_Limit_Identifier>0</Floor_Limit_Identifier>
<Type_of_Payment_Service_Requested>0 </Type_of_Payment_Service_Requested>
<Amount_Settlement>000000000000</Amount_Settlement>
<Currency_Code_Settlement> </Currency_Code_Settlement>
<Conversion_Rate_Settlement>00000000</Conversion_Rate_Settlement>
<Amount_Cardholder_Billing>000000000000</Amount_Cardholder_Billing>
<Currency_Code_Cardholder_Billing> </Currency_Code_Cardholder_Billing>
<Conversion_Rate_Cardholder_Billing>00000000</Conversion_Rate_Cardholder_
Billing>
<Net_Fee_Amount>D00000000000</Net_Fee_Amount>
<IRF_Billing_Currency>000</IRF_Billing_Currency>
<Exchange_Rate_from_IRF_Billing_Currency_to_Settlement_
Currency>00000000</Exchange_Rate_from_IRF_Billing_Currency_to_Settlement_
Currency>
<Abbreviation_of_International_Organization_and_Foreign_Institutions>
</Abbreviation_of_International_Organization_and_Foreign_Institutions>
<Mainland_China_Transaction_Indicator> </Mainland_China_Transaction_
Indicator>
<Amount_Transaction_Fee> </Amount_Transaction_Fee>
<QRC_Voucher_Number> </QRC_Voucher_Number>
<Reserved_Block_1> </Reserved_Block_1>
</row>
<row>
<TC>100</TC>
<Block_Bitmap>C000</Block_Bitmap>
<PAN>6250944000000772 </PAN>
<Amount_Transaction>000000008000</Amount_Transaction>
<Currency_Code_Transaction>643</Currency_Code_Transaction>
<Transaction_Date_Time>0112221513</Transaction_Date_Time>
<System_Trace_Audit_Number>004506</System_Trace_Audit_Number>
<Authorization_Identification_Response>003311</Authorization_
Identification_Response>
<Authorization_Date>0202</Authorization_Date>
<Retrieval_Reference_Number>701217004507</Retrieval_Reference_Number>
<Acquiring_IIN>34190643 </Acquiring_IIN>
<Forwarding_IIN>34190643 </Forwarding_IIN>
<Merchant_Type>5992</Merchant_Type>
<Card_Acceptor_Terminal_Identification>811794 </Card_Acceptor_Terminal_
Identification>
<Card_Acceptor_Identification_Code>230000014016 </Card_Acceptor_
Identification_Code>
<Card_Acceptor_Name_Location>IP KHVATOVA M G KAZAN
RUS</Card_Acceptor_Name_Location>
<Original_Transaction_Information>0000000000000000000000</Original_
Transaction_Information>
<Message_Reason_Code>0000</Message_Reason_Code>
```

```
<Single_or_Dual_Message_Identifier>1</Single_or_Dual_Message_Identifier>
<GSCS_Serial_Number>000000000</GSCS_Serial_Number>
<Receiving_Institution_Identification_Code> </Receiving_
Institution_Identification_Code>
<Issuing_Institution_Identification_Code> </Issuing_
Institution_Identification_Code>
<Identifier_of_GSCS_Notice>0</Identifier_of_GSCS_Notice>
<Transaction_Initiating_Channel>03</Transaction_Initiating_Channel>
<Identifier_of_Transaction_Features> </Identifier_of_Transaction_
Features>
<Reserved_Block0> </Reserved_Block0>
<Reserved_for_GSCS> </Reserved_for_GSCS>
<Installment_Payment_Terms> </Installment_Payment_Terms>
<Stand-in_Authorization_Identifier> </Stand-in_Authorization_Identifier>
<POS_Condition_Code> </POS_Condition_Code>
<Merchant_Country_Code> </Merchant_Country_Code>
<Transaction_Initiation_Method> </Transaction_Initiation_Method>
<Authorization_Type> </Authorization_Type>
<Card_Level> </Card_Level>
<Code_of_Pricing_Scheme> </Code_of_Pricing_Scheme>
<Special_Settlement_Currency> </Special_Settlement_Currency>
<Ecommerce_Indicator> </Ecommerce_Indicator>
<Card_Product> </Card_Product>
<Account_Attribute> </Account_Attribute>
<UnionPay_Standard_or_Nonstandard_Card_Indicator> </UnionPay_Standard_or_
Nonstandard_Card_Indicator>
<B2B_Business_Type> </B2B_Business_Type>
<B2B_Payment_Medium> </B2B_Payment_Medium>
<Special_Pricing_Indicator> </Special_Pricing_Indicator>
<POS_Entry_Mode>901</POS_Entry_Mode>
<Floor_Limit_Identifier>0</Floor_Limit_Identifier>
<Type_of_Payment_Service_Requested>0 </Type_of_Payment_Service_Requested>
<Amount_Settlement>000000000000</Amount_Settlement>
<Currency_Code_Settlement> </Currency_Code_Settlement>
<Conversion_Rate_Settlement>00000000</Conversion_Rate_Settlement>
<Amount_Cardholder_Billing>000000000000</Amount_Cardholder_Billing>
<Currency_Code_Cardholder_Billing> </Currency_Code_Cardholder_Billing>
<Conversion_Rate_Cardholder_Billing>00000000</Conversion_Rate_Cardholder_
Billing>
<Net_Fee_Amount>D000000000000</Net_Fee_Amount>
<IRF_Billing_Currency>000</IRF_Billing_Currency>
<Exchange_Rate_from_IRF_Billing_Currency_to_Settlement_
Currency>00000000</Exchange_Rate_from_IRF_Billing_Currency_to_Settlement_
Currency>
<Abbreviation_of_International_Organization_and_Foreign_Institutions>
</Abbreviation_of_International_Organization_and_Foreign_Institutions>
<Mainland_China_Transaction_Indicator> </Mainland_China_Transaction_
Indicator>
<Amount_Transaction_Fee> </Amount_Transaction_Fee>
<QRC_Voucher_Number> </QRC_Voucher_Number>
<Reserved_Block_1> </Reserved_Block_1>
</row>
<row>
<TC>100</TC>
<Block_Bitmap>C000</Block_Bitmap>
```

```
<PAN>6250942000000002 </PAN>
<Amount_Transaction>000000020000</Amount_Transaction>
<Currency_Code_Transaction>643</Currency_Code_Transaction>
<Transaction_Date_Time>0112221513</Transaction_Date_Time>
<System_Trace_Audit_Number>004508</System_Trace_Audit_Number>
<Authorization_Identification_Response>003312</Authorization_
Identification_Response>
<Authorization_Date>0202</Authorization_Date>
<Retrieval_Reference_Number>701217004509</Retrieval_Reference_Number>
<Acquiring_IIN>34190643 </Acquiring_IIN>
<Forwarding_IIN>34190643 </Forwarding_IIN>
<Merchant_Type>5992</Merchant_Type>
<Card_Acceptor_Terminal_Identification>811794 </Card_Acceptor_Terminal_
Identification>
<Card_Acceptor_Identification_Code>230000014016 </Card_Acceptor_
Identification_Code>
<Card_Acceptor_Name_Location>IP KHVATOVA M G KAZAN
RUS</Card_Acceptor_Name_Location>
<Original_Transaction_Information>0000000000000000000000</Original_
Transaction_Information>
<Message_Reason_Code>0000</Message_Reason_Code>
<Single_or_Dual_Message_Identifier>1</Single_or_Dual_Message_Identifier>
<GSCS_Serial_Number>000000000</GSCS_Serial_Number>
<Receiving_Institution_Identification_Code> </Receiving_
Institution_Identification_Code>
<Issuing_Institution_Identification_Code> </Issuing_
Institution_Identification_Code>
<Identifier_of_GSCS_Notice>0</Identifier_of_GSCS_Notice>
<Transaction_Initiating_Channel>03</Transaction_Initiating_Channel>
<Identifier_of_Transaction_Features> </Identifier_of_Transaction_
Features>
<Reserved_Block0> </Reserved_Block0>
<Reserved_for_GSCS> </Reserved_for_GSCS>
<Installment_Payment_Terms> </Installment_Payment_Terms>
<Stand-in_Authorization_Identifier> </Stand-in_Authorization_Identifier>
<POS_Condition_Code> </POS_Condition_Code>
<Merchant_Country_Code> </Merchant_Country_Code>
<Transaction_Initiation_Method> </Transaction_Initiation_Method>
<Authorization_Type> </Authorization_Type>
<Card_Level> </Card_Level>
<Code_of_Pricing_Scheme> </Code_of_Pricing_Scheme>
<Special_Settlement_Currency> </Special_Settlement_Currency>
<Ecommerce_Indicator> </Ecommerce_Indicator>
<Card_Product> </Card_Product>
<Account_Attribute> </Account_Attribute>
<UnionPay_Standard_or_Nonstandard_Card_Indicator> </UnionPay_Standard_or_
Nonstandard_Card_Indicator>
<B2B_Business_Type> </B2B_Business_Type>
<B2B_Payment_Medium> </B2B_Payment_Medium>
<Special_Pricing_Indicator> </Special_Pricing_Indicator>
<POS_Entry_Mode>901</POS_Entry_Mode>
<Floor_Limit_Identifier>0</Floor_Limit_Identifier>
<Type_of_Payment_Service_Requested>0 </Type_of_Payment_Service_Requested>
<Amount_Settlement>000000000000</Amount_Settlement>
<Currency_Code_Settlement> </Currency_Code_Settlement>
```

```

    <Conversion_Rate_Settlement>00000000</Conversion_Rate_Settlement>
    <Amount_Cardholder_Billing>000000000000</Amount_Cardholder_Billing>
    <Currency_Code_Cardholder_Billing>    </Currency_Code_Cardholder_Billing>
    <Conversion_Rate_Cardholder_Billing>00000000</Conversion_Rate_Cardholder_
Billing>
    <Net_Fee_Amount>D000000000000</Net_Fee_Amount>
    <IRF_Billing_Currency>000</IRF_Billing_Currency>
    <Exchange_Rate_from_IRF_Billing_Currency_to_Settlement_
Currency>00000000</Exchange_Rate_from_IRF_Billing_Currency_to_Settlement_
Currency>
    <Abbreviation_of_International_Organization_and_Foreign_Institutions>
</Abbreviation_of_International_Organization_and_Foreign_Institutions>
    <Mainland_China_Transaction_Indicator> </Mainland_China_Transaction_
Indicator>
    <Amount_Transaction_Fee>                </Amount_Transaction_Fee>
    <QRC_Voucher_Number>                    </QRC_Voucher_Number>
    <Reserved_Block_1>                </Reserved_Block_1>
</row>
<row>
    <TC>001</TC>
    <Block_Bitmap>8000</Block_Bitmap>
    <Transaction_Record_Number>0000000008</Transaction_Record_Number>
    <MAK>                </MAK>
    <MAC>                </MAC>
</row>
</file>

```

12.5.1 Поля клирингового диалекта UPI

Поля клирингового диалекта UPI соответствуют параметрам спецификации «UnionPay International. Technical Specifications on Bankcard Interoperability. Part III File Interface».

Почти для всех параметров название тега получается из названия соответствующего параметра спецификации, в котором пробелы и запятые заменены на символ подчеркивания. Например, тег System_Trace_Audit_Number соответствует параметру System trace audit number, а тег Amount_Transaction — параметру Amount, transaction.

В таблице 12-3 перечислены теги, которые образуются не в строгом соответствии с описанным правилом.

Таблица 12-3 Соответствие тегов параметрам спецификации UnionPay

Тег	Параметр спецификации UnionPay
TC	Transaction code
Currency_Code_Transaction	Currency, transaction
Transaction_Date_Time	Transaction date and time
Authorization_Date	Date of authorization
Acquiring_IIN	Acquiring institution identification code
Forwarding_IIN	Forwarding institution identification code
POS_Condition_Code	Point of service condition code

Ter	Параметр спецификации UnionPay
Code_of_Pricing_Scheme	Pricing scheme code
Ecommerce_Indicator	Ecommerce identifier
UnionPay_Standard_or_Nonstandard_Card_Indicator	UnionPay standard/ non-standard card indicator

12.6 Выгрузка клирингового файла

Чтобы выгрузить клиринговый файл, выполните следующие действия:

1. Создайте канал с диалектом **Выгрузка файла**, соответствующий типу клирингового файла, если он еще не создан.
Подробнее о каналах см. в разделе 4.1 «Каналы».
2. Создайте сценарий и добавьте к нему шаг, производящий выгрузку файла.
При создании шага выберите канал с соответствующим диалектом, укажите имя файла и в поле **Содержимое файла** задайте XML-представление, соответствующее типу клирингового файла.
3. Запустите сценарий.

12.7 Загрузка клирингового файла

Чтобы загрузить файл, выполните следующие действия:

1. Создайте канал с диалектом **Загрузка файла**, соответствующий типу клирингового файла, если он еще не создан.
Подробнее о каналах см. в разделе 4.1 «Каналы»
2. Создайте сценарий и добавьте к нему шаг, производящий загрузку файла.
Подробнее о сценариях и тестовых шагах см. в разделах 5.3 «Тестовые сценарии» и 5.4 «Тестовые шаги».
При создании шага выберите канал с соответствующим диалектом и укажите имя файла: абсолютный путь в файловой системе или путь относительно папки, выбранной администратором системы для хранения файлов данного формата.
3. Запустите сценарий.

13 ISO Responder

ISO Responder предназначен для автоматической генерации ответов на сообщения ISO без необходимости запуска тестовых сценариев.

Модуль поддерживает следующие типы респондеров:

- Авторизационный респондер: эмуляция ответа платежной системы на авторизационное сообщение.
- Респондер детокенизации.
- Респондер клиринга.
- 3DS1 Responder: эмуляция DS-серверов платежных систем Visa, Mastercard, МИР и ACS-сервера эмитента.
- 3DS2 Responder: эмуляция

Правило генерации ответа состоит из набора данных и действия, выполняемого при выборе правила.

Для настройки правила пользователь должен обладать полным доступом к модулю gsp.

При получении сообщения по соответствующему каналу респондер выполняет поиск правила ответа на основе полей группы **Условия**:

- Для полей типа **Список** проверяется соответствие хотя бы одному из значений списка.
- Для PAN проверяется попадание в диапазон значений.
- Остальные поля проверяются на равенство с учетом отступов, заданных в спецификации диалекта.

Респондер использует первое найденное правило. Если в правиле настроено формирование ответа:

1. Респондер преобразует входящее сообщение в TxnModel и дополняет модель значениями полей данных группы **Данные**, заданных в правиле, кроме списка **Поля**.
2. Преобразует модель в ответное сообщение, и добавляет значения списка **Поля**. Ответ отправляется банку-эквайеру.

13.1 Настройка работы с авторизационным респондером

Для настройки авторизационного респондера необходимо:

- Создать ISO 8583 канал (см. раздел 4.1):
 - Тип подключения: Сервер,
 - Режим ответа: Автоматический.
- Настроить правила ответа (см. раздел 13.2).

13.2 Настройка правила генерации ответа ISO 8583

Чтобы настроить правило генерации ответа ISO 8583:

1. Откройте форму **Правила ответа (Конфигурация → Правила ответа ISO 8583)**.
2. Добавьте запись в форму. Появится всплывающее окно, изображенное на рисунке 111.

Правило ответа ISO 8583

Добавить строку (+)

004:Transaction Amount 000000000000 018:Список MCC MCC 022:Pos Entry Mode 000 025:Pos Condition Code 00

Добавить строку (+)

041:Card Acceptor Terminal Id 042:Card Acceptor Id Code 049:Currency Code 000

Платежные данные

006:Billing Amount 000000000000 010:Billing Conversion Rate 00000000 016:Billing Conversion Date MMdd 051:Billing Currency Code 000

Данные

Поля Путь* Значение*

Добавить строку (+)

Карточные данные

Генерировать ARPC Набор параметров карты ...

Сохранить Отмена

Рисунок 111 Форма «Правило ответа ISO 8583»

- Введите параметры, описанные в таблице 13.
- Нажмите **Сохранить**.

Таблица 13-1 Параметры правила формирования ответа ISO 8583

Параметр	Описание
Детали	
Код	Код правила ответа
Наименование	Наименование правила ответа
Описание	Описание правила ответа. Справочная информация
Условия — параметры выбора правила для входящего сообщения	
Действие	Действие, заданное для данного правила (обязательный). Возможные значения: <ul style="list-style-type: none"> Разрешить — отправить ответ с кодом ответа 00 в поле 39, Отклонить — отправить ответ с кодом ответа из выпадающего списка Authorization response code в группе Карточные данные, Таймаут — не отправлять ответ. Используется для эмуляции таймаута
Название диалекта	Наименование диалекта
Активный	Параметр указывает, используется ли правило ответа в поиске. Чтобы исключить правило ответа из поиска, необходимо снять флаг

Параметр	Описание
Детокенизация	Флажок управляет включением режима онлайн-детокенизации: <ul style="list-style-type: none"> • флажок установлен — режим онлайн-детокенизации включен. В формировании ответа используются поля группы Детокенизация. • флажок снят — режим онлайн-детокенизации выключен. В формировании ответа используются поля группы Данные
001: Список MTI	Индикатор типа сообщения (список значений)
002: Начальный BIN	Наименьшее значение диапазона BIN (обязательный)
002: Конечный BIN	Наибольшее значение диапазона BIN (обязательный)
003: Список Processing Code	6-значный код обработки (список значений)
004: Transaction Amount	Сумма транзакции
018: Список MCC	Код MCC (список значений)
022: Pos Entry Mode	Режим ввода данных карты
025: Pos Condition Code	Код условия проведения транзакции
041: Card Acceptor Terminal Id	Значение Card Acceptor Terminal Id
042: Card Acceptor Id Code	Значение Card Acceptor Id
049: Currency Code	Числовой код валюты транзакции
Платежные данные	
006: Billing Amount	Сумма транзакции, сконвертированная в валюту расчетов (ПЛ051)
010: Billing Conversion Rate	Курс конвертации суммы транзакции (ПЛ004) в валюту расчетов (ПЛ051)
016: Billing Conversion Date	Дата, на которую действует используемый курс конвертации, в формате MMdd. Например, 0518 — 18 мая
051: Billing Currency Code	Числовой код валюты расчетов
Данные — параметры группы используются для формирования ответа в обычном режиме без онлайн-детокенизации	

Параметр	Описание
Поля	Список пар Путь/Значение для подстановки в ответное сообщение
Карточчные данные	
Генерировать ARPC	Параметр управляет генерацией криптограммы ARPC и используется для поддержки карт с обязательной проверкой ARPC. Рассчитанное значение включается в ответ эквайеру в поле 55 тег 91 «Issuer Authentication Data»
Набор параметров карты	Сохраненный набор параметров карты. Используется только при генерации ARPC
Authorization response code	Авторизационный код ответа, используемый при генерации ARPC Method 1 (Visa)
Детокенизация	
Набор параметров токена	Сохраненный набор параметров токена. Используется в запросе после того, как произвели токенизацию, и в ответе после того, как произвели детокенизацию
Канал	Клиентский канал эмитента, в который отправляется детокенизированное сообщение
Поля эмитента	Поля эмитента. Набор пар Путь/Значение для подстановки в ответное сообщение. Для формирования значений можно использовать контекстные функции, поля строки набора параметров ($\{\text{кодНабораПараметров.названиеПоля}\}$) и поля входящего сообщения ($\{_номерПоля\}$)
Поля эквайера	Поля эквайера. Набор пар Путь/Значение для подстановки в ответное сообщение. Для формирования значений можно использовать контекстные функции, поля строки набора параметров ($\{\text{кодНабораПараметров.названиеПоля}\}$) и поля входящего сообщения ($\{_номерПоля\}$)

13.3 Настройка онлайн-детокенизации

Для использования онлайн-детокенизации необходимо создать два канала ISO 8583.

1. Канал для получения сообщения от эквайера. **Тип подключения: Сервер, Режим ответа: Автоматический** (упомянутый в разделе 13.1).
2. Канал для получения сообщения из МПС и отправки эмитенту. **Тип подключения: Клиент, Режим отправки: Синхронный.**

В правиле ответа следует установить флаг **Детокенизация**.

13.4 Формирование ответа в режиме онлайн-детокенизации

Для формирования ответа используются данные из группы **Детокенизация**.

Формирование ответа состоит из следующих этапов:

1. Поиск строки набора параметров по токену, который указан в поле 02 входящего сообщения. Обязательные поля набора параметров: cardNumber, cardExpirationDate, tokenNumber.
2. Замена tokenNumber на cardNumber и tokenExpirationDate на cardExpirationDate из найденной строки набора параметров. Проставление полей из списка **Поля эмитента**.
3. Отправка детокенизированного сообщения в клиентский канал эмитента.
4. Получение ответа. Проставление полей из списка **Поля эквайера**.

13.5 Просмотр журнала сообщений ISO 8583

Респондер записывает все запросы, которые к нему поступают, в журнал сообщений. Журнал сообщений доступен для просмотра из формы правила ответа, а также через меню (**Тестирование** → **Журнал сообщений ISO 8583**).

Чтобы открыть журнал сообщений:

1. Откройте форму **Правила ответа (Конфигурация** → **Правила ответа ISO 8583)** (см. рисунок 112).

Правила ответа ISO 8583

Код	Наименование	Диапазон BIN	Название диалекта	Действие	Код ответа	Описание
visaSmsRetail	visaSmsRetail	4051000000000178 - 4817760103695221	visa-sms	Разрешить	00 - Approved	
visa	visa	4000000000000000 - 4000000000000009	visavip	Разрешить	00 - Approved	
mir	mir	220499000000000016 - 220499000000000016	mir-online	Разрешить	00 - Approved	
mir_token	mir_token	2200290400005894394 - 2200290400005894394	mir-online	Разрешить	00 - Approved	

Рисунок 112 Правила ответа ISO 8583

2. Перейдите на страницу правила по ссылке в колонке **Наименование**.
3. Прокрутите содержимое формы вниз: список сообщений, обработанных правилом, отображается в форме **Результаты** (см. рисунок 113).

Результаты (Всего 25)

Окружение	Дата начала	Дата окончания	PAN	Trace reference number	Authorization code
test	2022-07-06 18:18:37	2022-07-06 18:18:37	2204990000000000016		OLQUPJ
test	2022-07-06 18:18:07	2022-07-06 18:18:07	2204990000000000016		UTCZEB
test	2022-07-06 18:11:30	2022-07-06 18:11:30	2204990000000000016		UNE6YJ
test	2022-07-06 17:54:55	2022-07-06 17:54:55	2204990000000000000		UOSFAT
test	2022-07-06 17:39:53	2022-07-06 17:39:53	2204990000000000000		CTBNQQ
test	2022-07-06 15:52:14	2022-07-06 15:52:14	2204990000000000016		EKZF6D
test	2022-07-06 15:50:58	2022-07-06 15:50:58	2204990000000000016		EUONGI
test	2022-07-06 12:22:56	2022-07-06 12:22:56	2900000000000000008		SFASGI
test	2022-07-06 12:05:12	2022-07-06 12:05:12	2900000000000000008		EBWVK8
test	2022-07-06 12:01:25	2022-07-06 12:01:26	2900000000000000008		HGJ8GX
test	2022-07-06 12:01:11	2022-07-06 12:01:11	2900000000000000008		1ZZU8U
test	2022-07-06 12:00:48	2022-07-06 12:00:57	2900000000000000008		FAJ0HL
test	2022-07-05 18:23:21	2022-07-05 18:23:21	22222000000001108		LQG4CY
test	2022-07-05 18:23:09	2022-07-05 18:23:16	22222000000001108		1YVTXX
test	2022-06-28 18:59:22	2022-06-28 18:59:22	22222000000001108		9IFXUV

Рисунок 113 Журнал сообщений ISO 8583

4. Перейдите на страницу сообщения по ссылке в колонке **Дата начала** (см. рисунок 114).

▼ **Детали**

Правило ответа

Начальный BIN Конечный BIN

Окружение Действие

Дата начала Дата окончания

Детокенизация

Название диалекта

▼ **Исходящее сообщение**

Поле / Ter	Значение
000 - MTI	0110
002 - Primary Account Number (PAN)	2204990000000000016
003/1 - Operation Type	00
003/2 - Originator Account Type	00
003/3 - Receiver Account Type	00
004 - Amount, Transaction	000000000001
007 - Transmission Date and Time	0706181837
011 - System Trace Audit Number (STAN)	615439
032 - Acquiring Institution ID Code	485016
033 - Forwarding Institution ID Code	485016
037 - Retrieval Reference Number (RRN)	218700001532
038 - Authorization ID Response	OLQUPJ

▼ **Входящее сообщение**

Поле / Ter	Значение
000 - MTI	0100
002 - Primary Account Number (PAN)	2204990000000000016
003/1 - Operation Type	00
003/2 - Originator Account Type	00
003/3 - Receiver Account Type	00
004 - Amount, Transaction	000000000001
007 - Transmission Date and Time	0706181837
011 - System Trace Audit Number (STAN)	615439
014 - Date, Card Expiration	2212
018 - Merchant Type	4121
022 - Point of Service Entry Mode	010
023 - Card Sequence Number	00
032 - Acquiring Institution ID Code	485016

Рисунок 114 Детали сообщения ISO 8583

В журнале сообщений отображаются детали правила, запрос, сопоставленный респондером, и сформированный респондером ответ (исходящее сообщение).

13.6 Загрузка клиринга

При загрузке клирингового файла выполняются проверки на соответствие транзакций, полученных в составе клирингового файла от банка-эквайера, операциям, обработанным респондером. Если соответствие найдено, то данные транзакции дополняются данными сообщения, которое пришло в респондер.

Кроме того, респондер может выполнить онлайн-детокенизацию. Чтобы настроить данную функциональность, до загрузки клиринга банк-эквайер должен прислать респондеру запрос на проведение онлайн-детокенизации.

Поиск результата респондера в режиме онлайн-детокенизации выполняется по следующему алгоритму:

1. Поиск соответствия по tokenNumber и traceReferenceNumber для каждой транзакции из клирингового файла.
2. Поиск соответствия по tokenNumber и authCode для каждой транзакции из клирингового файла, если не удалось установить однозначное соответствие в п.1.
3. Если в результате выполнения двух пунктов не удалось установить однозначное соответствие, считаем матчинг неудачным, и данные респондера не будут добавлены в выгружаемый клиринговый файл. Иначе к каждому элементу транзакции в выгружаемый клиринговый файл добавляется блок респондера, содержащий запрос из найденного результата респондера и строку набора параметров (см. примеры файлов до матчинга и после).

Пример 13-1 Пример клирингового файла до матчинга

```
<file>
...
  <row>
    <_0>1240</_0>
    <_2>5161160204817965</_2>
```

```

    <_3>280000</_3>
    <_4>000000100000</_4>
    <_5>000000100000</_5>
    <_6>000000100000</_6>
    <_9>40010000</_9>
    <_10>40010000</_10>
    ...
  </row>
</file>

```

Пример 13-2 Пример клирингового файла после проведения матчинга

```

<file>
...
  <row>
    <responder>
      <request>
        <_0>0100</_0>
        <_2>5161160204817965</_2>
        <_3>000000</_3>
        <_4>000000002000</_4>
        <_7>1110205958</_7>
        <_11>000172</_11>
        <_14>2212</_14>
        <_18>4121</_18>
        <_22>010</_22>
        <_32>010</_32>
        <_37>831400299010</_37>
        <_41>825187</_41>
        <_42>825186</_42>
        <_43>PAO TEST BANK MOSCOW RU</_43>
        <_49>643</_49>
        <_53>
          <_1>97</_1>
          <_2>01</_2>
          <_3>10</_3>
          <_4>0011</_4>
          <_5>00</_5>
          <_6>0000</_6>
        </_53>
        <_61>10100100019007040000000000</_61>
      </request>
      <parameterSetRow>
        <tokenNumber>5161160204817965</tokenNumber>
        <cardNumber>1234567890123456</cardNumber>
        <cardExpirationDate>2111</cardExpirationDate>
      </parameterSetRow>
    </responder>
    <_0>1240</_0>
    <_2>5161160204817965</_2>
    <_3>280000</_3>
    <_4>000000100000</_4>
    <_5>000000100000</_5>
    <_6>000000100000</_6>
    <_9>40010000</_9>
    <_10>40010000</_10>

```

```
...  
</row>  
</file>
```

13.7 3DS1 Responder

3DS1 Responder эмулирует DS-серверы платежных систем и ACS-сервер эмитента, тем самым позволяет проверить корректность работы домена эквайера без подключения к реальному эмитенту и платежной системе.

13.8 Настройка работы с 3DS1 Responder

Чтобы настроить работу с 3DS1 Responder:

- Создайте HTTP-канал (см. раздел 4.1 и пример конфигурации на рисунке 115):
 - Тип подключения: Сервер,
 - Режим ответа: Автоматический.

The screenshot shows a configuration form for an HTTP channel. It is divided into two sections: 'Детали' (Details) and 'Дополнительные настройки' (Additional settings).

Детали:

- Код: DS-server
- Наименование: DS-server
- Система: DEFAULT SYSTEM
- Диалект: HTTP
- Режим отправки: Синхронный
- Режим ответа: Автоматический
- Время ожидания ответа (мс): 9000
- Описание: (empty text area)

Дополнительные настройки:

- Тип подключения: Сервер
- Тип исходящего сообщения: XML
- Тип входящего сообщения: XML
- Анализировать заголовки запроса:

Рисунок 115 Пример конфигурации HTTP-канала для 3DS1 Responder

- Настройте правила ответа (см. раздел 13.9).

13.9 Настройка правила генерации ответа 3DS1

Чтобы настроить правило генерации ответа 3DS1:

1. Откройте форму **Правила ответа (Конфигурация → Правила ответа 3DS1)**.
2. Добавьте запись в форму. Появится всплывающее окно, изображенное на рисунке 116.

Правило ответа 3DS1 ✕

Детали

Код* Наименование*

Описание

Условия

Начальный BIN* Конечный BIN* Активный

Диалект DS канал*

Данные VERes

Возможность* аутентификации

ACS канал Произвольный ACS URL

Данные PAREs

Статус* транзакции

Данные Merchant Callback

Канал merchant Статус

Рисунок 116 Форма «Правило ответа 3DS1»

3. Введите параметры, описанные в таблице 13-2.
4. Нажмите **Сохранить**.

Таблица 13-2 Параметры правила формирования ответного сообщения 3DS1

Параметр	Описание
Детали	
Код	Код правила ответа (до 32 символов). Уникальный среди правил данного респондера
Наименование	Наименование правила ответа (до 256 символов)
Описание	Описание правила ответа. Справочная информация
Условия — параметры выбора правила для входящего сообщения	
Начальный BIN	Наименьшее значение диапазона BIN (обязательный). От 13 до 19 символов
Конечный BIN	Наибольшее значение диапазона BIN (обязательный). От 13 до 19 символов
Диалект	Наименование диалекта, которому соответствует обрабатываемое сообщение. Например, Visa. Необходимо для вычисления значения ЕСІ

Параметр	Описание
DS канал	<p>Код HTTP-канала с типом подключения Сервер и автоматическим режимом ответа, на который необходимо отправить сообщение VEReq для его дальнейшей обработки респондером. Входящее и исходящее сообщение — в формате XML.</p> <p>Пример конфигурации канала приведен на рисунке 115</p>
Активный	Параметр указывает, используется ли правило ответа. Чтобы исключить правило ответа из поиска, необходимо снять флаг
Данные VERes — параметры группы используются для формирования ответного сообщения VERes на запрос VEReq	
Возможность аутентификации	<p>Статус ответа, который определяет, будет ли отправляться PAREq.</p> <p>Возможные значения:</p> <ul style="list-style-type: none"> • Y - Доступна, • N - Не поддерживается, • U - Недоступна. <p>PAREq отправляется только при статусе Y</p>
ACS канал	<p>Код HTTP-канала с типом подключения Сервер и автоматическим режимом ответа, на который необходимо отправить сообщение PAREq для его дальнейшей обработки респондером.</p> <p>Входящее сообщение — в формате ключ-значение, исходящее — в формате HTML. Обязателен, если поле Возможность аутентификации имеет значение Y - Доступна</p>
Произвольный ACS URL	Флаг, который позволяет указать ACS канал как "сырой" URL
ACS URL	URL ACS-сервера. Отображается, если установлен флаг Произвольный ACS URL
Создавать IReq	<p>Параметр управляет включением дополнительных полей в сообщение VERes.</p> <p>Флаг отображается, если в поле Возможность аутентификации выбрано значение N - Не поддерживается или U - Недоступна</p>
Код IReq	<p>Причина, по которой запрос не может быть обработан. При этом сам запрос составлен корректно.</p> <p>Поле отображается, если установлен флаг Создавать IReq</p>
Детали IReq	<p>Поле, из-за которого запрос не может быть обработан.</p> <p>Поле отображается, если установлен флаг Создавать IReq</p>

Параметр	Описание
Код вендора	<p>Специфичный для конкретного вендора код ошибки или поясняющий текст, который поможет установить причину ошибки (максимальный размер 256 символов).</p> <p>Поле отображается, если установлен флаг Создавать IReq</p>
<p>Данные PAREs — параметры группы используются для формирования ответного сообщения PAREs на запрос PAREq</p>	
Статус транзакции	<p>Параметр определяет, является ли транзакция аутентифицированной.</p> <p>Возможные значения:</p> <ul style="list-style-type: none"> • Y - Успешная аутентификация, • N - Неудачная аутентификация, • U - Аутентификация не может быть выполнена, • A - Произведена попытка аутентификации
<p>Данные CAVV — параметры группы используются для генерации CAVV. CAVV генерируется при значениях поля Статус транзакции Y - Успешная аутентификация, A - Произведена попытка аутентификации</p>	
CAV ключ	Ключ, используемый при генерации CAVV. HEX-строка длиной 32 символа
Индикатор ключа	Уникальный индикатор ключа внутри ACS-сервера
<p>Данные Merchant Callback — параметры группы используются в формировании ответного сообщения на POST-запрос</p>	
Канал merchant callback	Код HTTP-канала с типом подключения Сервер и автоматическим режимом ответа, который принимает POST-запрос и отвечает сообщением с пустым телом
Статус	Статус ответного сообщения

13.10 Примеры сообщений VERes

Пример 13-3 содержит сообщение VERes, в формировании которого участвуют поля, приведенные на рисунке 117.

∨
Данные VERes

Возможность аутентификации

Произвольный ACS URL

ACS канал

Рисунок 117 Группа «Данные VERes»

Пример 13-3 Пример сообщения VERes

```
<?xml version="1.0" encoding="UTF-8"?>
<ThreeDSecure>
  <Message id="3364380">
    <VERes>
      <version>1.0.2</version>
      <CH>
        <enrolled>N</enrolled>
      </CH>
      <url>http://localhost:7071/ext/3ds1_acs/test.3ds1_acs_server</url>
      <protocol>ThreeDSecure</protocol>
    </VERes>
  </Message>
</ThreeDSecure>
```

В таблице 13-3 содержится описание полей сообщения VERes.

Таблица 13-3 Поля сообщения VERes

Поле	Описание
Message.id	Копируется из VEReq
version	Копируется из VEReq
CH.enrolled	Значение поля Возможность аутентификации на форме правила ответа
CH.acctID	Уникальное значение, генерируемое респондером
url	Составляется на основе данных канала, если заполнено поле ACS канал , или берется из поля ACS URL , если установлен флажок Произвольный ACS URL
protocol	Константа ThreeDSecure

Если в правиле ответа в поле **Возможность аутентификации** выбрано значение **N - Не поддерживается** или **U - Недоступна**, появляется опция **Создавать IReq** (см. рисунок 118), при активации которой появляются дополнительные поля, которые включаются в блок IReq сообщения VERes (см. пример 13-4).

▼ Данные VERes

Возможность аутентификации	<input type="text" value="N - Не поддерживается"/>	Создавать IReq	<input checked="" type="checkbox"/>
ACS канал	<input type="text"/>	Произвольный ACS URL	<input type="checkbox"/>
Код IReq	<input type="text" value="99 - Permanent system failure"/>	Детали IReq	<input type="text" value="Details"/>
Код вендора	<input type="text" value="Details"/>		

Рисунок 118 Группа «Данные VERes» с полями IReq

Пример 13-4 Пример сообщения VERes с блоком IReq

```
<?xml version="1.0" encoding="UTF-8"?>
<ThreeDSecure>
```

```
<Message id="3364380">
  <VERes>
    <version>1.0.2</version>
    <CH>
      <enrolled>N</enrolled>
    </CH>
    <IReq>
      <iReqCode>50</iReqCode>
      <iReqDetail>not participating</iReqDetail>
      <vendorCode>vendor code</vendorCode>
    </IReq>
  </VERes>
</Message>
</ThreeDSecure>
```

В таблице 13-4 содержится описание полей специфичного сообщения VERes.

Таблица 13-4 Поля специфичного сообщения VERes

Поле	Описание
Message.id	Копируется из VReq
version	Копируется из VReq
CH.enrolled	Значение поля Возможность аутентификации на форме правила ответа
IReq.iReqCode	Значение поля Код IReq на форме правила ответа
IReq.iReqDetail	Значение поля Детали IReq на форме правила ответа
IReq.vendorCode	Значение поля Код вендора на форме правила ответа

13.11 Примеры сообщений PAREs

Пример 13-3 содержит обычное сообщение PAREs.

Пример 13-5 Пример обычного PAREs

```
<?xml version="1.0" encoding="UTF-8"?>
<ThreeDSecure>
  <Message id="336507">
    <PAREs id="pares_336507">
      <version>1.0.2</version>
      <Merchant>
        <acqBIN>467872</acqBIN>
        <merID>M00000000001005</merID>
      </Merchant>
      <Purchase>
        <xid>MDAwMDAwMDAwMDAwMDAzMzY1MDg=</xid>
        <date>20190806 18:04:14</date>
        <purchAmount>100</purchAmount>
        <currency>840</currency>
        <exponent>2</exponent>
      </Purchase>
      <pan>0000000000002101</pan>
    </PAREs>
  </Message>
</ThreeDSecure>
```

```

                <time>20221030 19:51:33</time>
                <status>Y</status>
                <cavv>AAABBHNSlwAAAAABVKXAAAAAAA=</cavv>
                <eci>05</eci>
                <cavvAlgorithm>2</cavvAlgorithm>
            </TX>
        </PAREs>
    </Message>
</ThreeDSecure>
    
```

В таблице 13-3 содержится описание полей сообщения PAREs.

Таблица 13-5 Поля сообщения PAREs

Поле	Описание
Message.id	Копируется из PAREq
PAREs.id	Генерируется респондером как pares_ + Message.id
version	Копируется из PAREq
Merchant.acqBIN	Копируется из PAREq
Merchant.merID	Копируется из PAREq
Purchase.xid	Копируется из PAREq
Purchase.date	Копируется из PAREq
Purchase.purchAmount	Копируется из PAREq
Purchase.currency	Копируется из PAREq
Purchase.exponent	Копируется из PAREq
pan	Значение pan из VEREq, где все цифры кроме последних 4 заменены на 0
TX.time	Текущее время GMT в формате ууууММдд НН:мм:сс
TX.status	Значение поля Статус транзакции на форме правила ответа
TX.cavv	Генерируется респондером на основе данных из группы Данные CAVV и pan
TX.eci	Генерируется респондером в зависимости от диалекта и статуса транзакции. Visa: <ul style="list-style-type: none"> • Y - Успешная аутентификация — 05 • A - Произведена попытка аутентификации – 06 Mastercard, MIR: <ul style="list-style-type: none"> • Y - Успешная аутентификация — 02 • A - Произведена попытка аутентификации – 01
TX.cavvAlgorithm	Константа 2

Если в правиле ответа в поле **Статус транзакции** выбрано значение **N - Неудачная аутентификация** или **U - Аутентификация не может быть выполнена**, появляется опция **Создавать IReq** (см. рисунок 118), при активации которой появляются дополнительные поля, которые включаются в блок IReq сообщения PAREs (см. пример 13-6).

Пример 13-6 Пример сообщения PAREs с блоком IReq

```
<?xml version="1.0" encoding="UTF-8"?>
<ThreeDSecure>
  <Message id="336507">
    <PAREs id="pares_336507">
      <version>1.0.2</version>
      <Merchant>
        <acqBIN>467872</acqBIN>
        <merID>M00000000001005</merID>
      </Merchant>
      <Purchase>
        <xid>MDAwMDAwMDAwMDAwMDAzMzY1MDg=</xid>
        <date>20190806 18:04:14</date>
        <purchAmount>100</purchAmount>
        <currency>840</currency>
        <exponent>2</exponent>
      </Purchase>
      <pan>0000000000000000</pan>
      <TX>
        <time>20221030 20:40:33</time>
        <status>Y</status>
      </TX>
      <IReq>
        <IReqCode>50</IReqCode>
        <iReqDetail>not participating</iReqDetail>
        <vendorCode>vendor code</vendorCode>
      </IReq>
    </PAREs>
  </Message>
</ThreeDSecure>
```

В таблице 13-6 содержится описание полей специфичного сообщения VERes.

Таблица 13-6 Поля специфичного сообщения PAREs

Поле	Описание
Message.id	Копируется из PAREq
PAREs.id	Генерируется респондером как pares_ + Message.id
version	Копируется из PAREq
Merchant.acqBIN	Копируется из PAREq
Merchant.merID	Копируется из PAREq
Purchase.xid	Копируется из PAREq
Purchase.date	Копируется из PAREq

Поле	Описание
Purchase.purchAmount	Копируется из PAREq
Purchase.currency	Копируется из PAREq
Purchase.exponent	Копируется из PAREq
pan	Значение pan из VEREq, где все цифры заменены на 0
TX.time	Текущее время GMT в формате ууууММдд НН:мм:сс
TX.status	Значение поля Статус транзакции на форме правила ответа
IReq.iReqCode	Значение поля Код IReq на форме правила ответа
IReq.iReqDetail	Значение поля Детали IReq на форме правила ответа
IReq.vendorCode	Значение поля Код вендора на форме правила ответа

13.12 Просмотр журнала сообщений 3DS1

Журнал сообщений 3DS1 доступен для просмотра через меню (**Тестирование** → **Журнал сообщений 3DS1**).

Чтобы открыть журнал сообщений:

1. Выберите меню **Тестирование** → **Журнал сообщений 3DS1** (см. рисунок 119).

Журнал сообщений 3DS1

Дата начала	Дата окончания	Окружение	Правило ответа	PAN	Account id	Тип	Статус
2022-11-01 12:40:01	2022-11-01 12:40:02	test	3ds1_test			Merchant Callback	Успешно
2022-11-01 12:40:01	2022-11-01 12:40:01	test	3ds1_test	4704340000132101	tdng7+JgT1u6cTGX1esbFw==	3DS1	Успешно
2022-11-01 12:38:36	2022-11-01 12:38:36	test	3ds1_test			Merchant Callback	Успешно
2022-11-01 12:38:32	2022-11-01 12:38:35	test	3ds1_test	4704340000132101	zrQKVJXDSX2/N/oytlKhuw==	3DS1	Успешно
2022-11-01 12:16:05	2022-11-01 12:16:05	test	3ds1_test			Merchant Callback	Успешно
2022-11-01 12:16:01	2022-11-01 12:16:04	test	3ds1_test	4704340000132101	nBrimTRTHS9bEV+J8IQsg==	3DS1	Успешно
2022-10-31 12:16:18	2022-10-31 12:16:19	test	3ds1_test			Merchant Callback	Успешно
2022-10-31 12:16:15	2022-10-31 12:16:18	test	3ds1_test	4704340000132101	qfFOKCoDSHm8nSEBJCEzpw==	3DS1	Успешно
2022-10-28 13:03:33	2022-10-28 13:03:33	test	3ds1_test			Merchant Callback	Успешно
2022-10-28 13:03:31	2022-10-28 13:03:33	test	3ds1_test	4704340000132101	8mUfY9RJTHa+JujkmFuQMw==	3DS1	Успешно
2022-10-26 14:48:27	2022-10-26 14:48:27	test	3ds1_test	4704340000132101	VqkBy5vROeFGnLW+8MZnw==	3DS1	Ожидает
2022-10-26 14:48:27	2022-10-26 14:48:27	test	3ds1_test			Merchant Callback	Успешно
2022-10-26 14:48:23		test	Правило для тестирования негативного VeRes	4704340000132102		3DS1	Ожидает
2022-10-26 14:48:23	2022-10-26 14:48:27	test	3ds1_test	4704340000132101	/2yc9z5lRzq75fYzskRWHg==	3DS1	Успешно
2022-10-26 14:48:23	2022-10-26 14:48:27	test	3ds1_test	4704340000132101	FINNByUZTHqhDhaWGfhwsw==	3DS1	Успешно
2022-10-26 14:48:23	2022-10-26 14:48:27	test	Правило для тестирования негативного PaRes	4704340000132103	+oKwZ9qwSPqnyvjGbtTLwQ==	3DS1	Успешно

Рисунок 119 Журнал сообщений 3DS1

2. Найдите сообщение в списке и перейдите по ссылке в колонке **Дата начала**.

В журнале сообщений отображаются детали правила, входящее сообщение и сформированный респондером ответ (исходящее сообщение) в формате xml и json.

Детали

▼
Детали

Правило ответа	3ds1_test	Дата начала	2022-11-09 17:31:55	📅	Дата окончания	2022-11-09 17:31:55	📅
Начальный BIN	4704340000132101	Конечный BIN	4704340000132101	Тип	Merchant Callback		
Окружение	test	Статус	Успешно				

Таблица 13-7 Детали журнала сообщений 3DS1

Поле	Описание
Правило ответа	Правило ответа 3DS1, на основе которого формировались ответные сообщения
Дата начала	Дата начала обработки первого сообщения (VEReq)
Дата окончания	Дата окончания формирования последнего сообщения (может быть как PAREs, так и VERes)
Начальный BIN	Начальный BIN, который указан на форме правила ответа
Конечный BIN	Конечный BIN, который указан на форме правила ответа
Тип	Тип сообщения из журнала. Возможные значения: 3DS1, Merchant Callback
Окружение	Окружение серверного HTTP-канала, на котором обрабатывались сообщения
Статус	Текущий статус сообщения. Возможные значения: <ul style="list-style-type: none"> • Ожидает — VEReq получен, сформирован VERes. Ожидается PAREq • Ошибка — возникла непредвиденная ошибка. Можно проанализировать по логам, привязанным к конкретному сообщению из журнала. • Успешно — вся цепочка сообщений 3DS1 успешно обработана • Выполняется — респондер обрабатывает входящее сообщение

Сообщения VEReq, VERes

Сообщение VReq

```

1- <?xml version="1.0" encoding="UTF-8"?><ThreeDSecure>
2-
3-
4-   <Message id="3364380">
5-
6-
7-     <VReq>
8-
9-
10-       <version>1.0.2</version>
11-
12-
13-       <pan>4704340000132101</pan>
14-
15-
16-       <Merchant>
17-
18-
19-         <acqBIN>467872</acqBIN>
20-
21-
22-         <merID>M00000000001005</merID>
23-
24-
25-         <password>INdian14</password>
26-
27-
28-       </Merchant>
29-
30-

```

URL /ext/3ds1_ds/test.3ds1_ds_server

Метод POST

Наименование	Значение
Content-Type	text/xml
user-agent	ReactorNetty/1.0.18
host	localhost:7070
accept	/*
content-length	669

Сообщение VRes

```

1- <?xml version="1.0" encoding="UTF-8"?><ThreeDSecure>
2-   <Message id="3364380">
3-     <VRes>
4-       <version>1.0.2</version>
5-       <Cb>
6-         <enrolled>Y</enrolled>
7-         <acctID>pNCMaw65oSojH0+m54L3Q==</acctID>
8-       </Cb>
9-       <url>http://localhost:7071//ext/3ds1_acs//test.3ds1_acs_server</url>
10-     </VRes>
11-   </Message>
12- </ThreeDSecure>
13-

```

Статус 200 - OK

Наименование	Значение
Content-Type	text/xml

Сообщения VReq и VRes участвуют в эмуляции DS. Входящее сообщение передается в формате XML. Поиск правила ответа выполняется по полю pan входящего сообщения VReq среди активных каналов. В случае обычного ответа формируется сообщение VRes с полем url, на который в дальнейшем необходимо отправить PAREq для успешного завершения обработки транзакции. Также респондер генерирует уникальное значение acctID, которое передается в PAREq для формирования цепочки сообщений.

Сообщения PAREq, PAREs

Сообщение PAREq

```

1- {
2-   "PaReq" : "eNpdLu9vgjAQ/VcIZ2Vdsq2DRndUoWeYHnJn0ZB+x3JREQPkxh/9WgTd1qTJvevd6+u7wSSHIOvzIS45y
3-   "id" : "XUjAi iFDJRu4A3jw6GSXJA",
4-   "TermUrl" : "https://Pit.3dsecure.net/VbVTestSuiteService/pit1/reviewPAREs?csrf=vyvystokovso
5-   "MD" : "transactionID=17d7ccfc-745b-455e-8ee4-78971ae27b44&selectedTestCase=216"
6- }

```

URL /ext/3ds1_acs/test.3ds1_acs_server

Метод POST

Наименование	Значение
Content-Type	application/x-www-form-urlencoded
user-agent	ReactorNetty/1.0.18
host	localhost:7071
accept	/*
content-length	847

Сообщение PAREs

```

1- <html>
2- <head>
3-   <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
4-   <meta content="text/html; charset=UTF-8" http-equiv="Content-Type" />
5-   <title>Return to Merchant's site</title>
6-   <script>
7-     function onLoadHandler() {
8-       document["PAREsForm"].submit();
9-     }
10-   </script>
11- </head>
12- <body onload="onLoadHandler()">
13-   <br />
14-   <br /> Processing...
15-   <form name="PAREsForm" method="post" action="https://Pit.3dsecure.net/VbVTestSuiteService/pi
16-   <input name="PaRes" type="hidden" value="eJyWlUoPasos9/Pp5JM0efz0I1Aq7DNPcURFBLsrLCUrLRQ
17-   <input name="MD" type="hidden" value="transactionID=17d7ccfc-745b-455e-8ee4-78971ae27b44&
18- </form>
19- </body>
20- </html>

```

Статус 200 - OK

Наименование	Значение
Content-Type	text/html

Сообщения PAREq и PAREs участвуют в эмуляции сервера ACS. Входящее сообщение PAREq передается в формате ключ-значение в сжатом закодированном виде (Deflate + Base64). Его необходимо раскодировать и разжать, чтобы получить PAREq в формате XML. По полю acctID выбирается сообщение из журнала, и связанное с ним правило ответа используется для формирования сообщения PAREs. Добавление подписи, а также сжатие и

кодирование PAREs осуществляется с помощью функции pagesDirect из модуля IPA. В ответе возвращается HTML страница с переадресацией на TermUrl.

13.13 3DS2 Responder

3DS2 Responder эмулирует эмитента (ACS) в виде обработки цепочки сообщений. Перед тем как ответить на входящий запрос CReq, осуществляется отправка дополнительного сообщения RReq, которое содержит результат аутентификации и передается эквайеру (3DS Server). После получения ответного сообщения RRes от эквайера (3DS Server) формируется сообщение CRes, которое отправляется как синхронный ответ на запрос CReq, и цепочка сообщений заканчивается.

13.13.1 Настройка правила генерации ответа 3DS2

Чтобы настроить правило генерации ответа 3DS2:

1. Откройте форму **Правила ответа 3DS2 (Конфигурация → Правила ответа 3DS2)**.
2. Добавьте запись в форму. Появится всплывающее окно, изображенное на рисунке 120.

Рисунок 120 Форма «Правило ответа»

3. Введите параметры, описанные в таблице 13-8.
4. Нажмите **Сохранить**.

Таблица 13-8 Параметры правила формирования ответного сообщения 3DS2

Параметр	Описание
Детали	Детали
Код	Код правила ответа (до 32 символов). Уникальный среди правил данного респондера
Наименование	Наименование правила ответа (до 256 символов)

Параметр	Описание
Описание	Описание правила ответа. Справочная информация
Условия — параметры выбора правила для входящего сообщения	
Начальный BIN	Наименьшее значение диапазона BIN (обязательный). От 13 до 19 символов
Конечный BIN	Наибольшее значение диапазона BIN (обязательный). От 13 до 19 символов
Активный	Параметр указывает, используется ли правило ответа в поиске. Чтобы исключить правило ответа из поиска, необходимо снять флаг
Диалект	Наименование диалекта, которому соответствует обрабатываемое сообщение. Например, Visa. Необходимо для вычисления значения ECI
DS канал(AReq)	Код HTTP-канала с типом подключения Сервер и автоматическим режимом ответа, на который необходимо отправить сообщение AReq для его дальнейшей обработки респондером. Входящее и исходящее сообщение — в формате JSON. Пример конфигурации канала приведен на рисунке 13.13
Данные ARes — параметры группы используются для формирования ответного сообщения ARes на запрос AReq	
Статус транзакции	Статус ответа, в зависимости от которого принимается решение отправлять CReq или нет. Возможные значения: <ul style="list-style-type: none"> • Y - Успешная аутентификация, • N - Неудачная аутентификация, • U - Аутентификация не может быть выполнена, • A - Произведена попытка аутентификации, • C - Необходима аутентификация CReq/CRes, • R - Аутентификация отклонена. CReq отправляется только при статусе C - Необходима аутентификация CReq/CRes
DS reference number	Уникальный идентификатор DS, назначенный EMV (максимальный размер 32 символа)
ACS reference number	Уникальный идентификатор ACS, назначенный EMV (максимальный размер 32 символа)
ACS operator ID	Идентификатор ACS, назначенный DS (максимальный размер 32 символа)

Параметр	Описание
ACS challenge mandated	Индикатор указывает, необходим ли challenge из-за местных/региональных ограничений или других случаев. Возможные значения: <ul style="list-style-type: none"> • Y - CReq/CRes обязателен, • N - CReq/CRes не обязателен
Тип аутентификации	Параметр указывает, какой тип аутентификации владельца карты будет использовать эмитент. Возможные значения: <ul style="list-style-type: none"> • 01 - Static, • 02 - Dynamic, • 03 - OOB
Канал ACS(CReq)	HTTP-канал с типом подключения Сервер и автоматическим режимом ответа, на который необходимо отправить сообщение CReq для его дальнейшей обработки респондером. Входящее сообщение — в формате JSON, исходящее — в формате HTML
Произвольный ACS URL	Флаг, который позволяет указать ACS канал как "сырой" URL
ACS URL	URL ACS-сервера. Отображается, если установлен флаг Произвольный ACS URL
Данные CRes — параметры группы используются для формирования ответного сообщения CRes на запрос CReq	
Статус транзакции CRes	Статус CRes, является финальным статусом транзакции. Возможные значения: <ul style="list-style-type: none"> • Y - Успешная аутентификация, • N - Неудачная аутентификация
Анализировать RRes	Параметр позволяет учитывать содержимое сообщения RRes для определения статуса транзакции в CRes
Статус транзакции RReq	Статус RReq. Отражает результат аутентификации, который передается от эмитента (ACS) к эквайеру (3DS Server). Возможные значения: <ul style="list-style-type: none"> • Y - Успешная аутентификация, • N - Неудачная аутентификация, • U - Аутентификация не может быть выполнена, • A - Произведена попытка аутентификации, • R - Аутентификация отклонена

Параметр	Описание
Канал 3DS server(RReq)	<p>HTTP-канал с типом подключения Клиент, используется при отправке RReq.</p> <p>Входящее и исходящее сообщение — в формате JSON.</p> <p>Необходимо в настройках канала для окружения оставить пустым поле Шаблон URL, чтобы респондер самостоятельно его определил, и заполнить настройки SSL</p>
Причина статуса транзакции RReq	Код причины негативного статуса транзакции
<p>Данные CAVV — параметры группы используются для генерации CAVV. CAVV генерируется при значениях поля Статус транзакции Y - Успешная аутентификация, A - Произведена попытка аутентификации</p>	
CAV ключ	Ключ, используемый при генерации CAVV. HEX-строка длиной 32 символа
Индикатор ключа	Уникальный индикатор ключа внутри ACS-сервера
<p>Данные Merchant Callback — параметры группы используются в формировании ответного сообщения на POST-запрос</p>	
Канал merchant callback	Код HTTP-канала с типом подключения Сервер и автоматическим режимом ответа, который принимает POST-запрос и отвечает сообщением с пустым телом
Статус	Статус ответного сообщения

13.13.2 Примеры ARes

Challenge Flow — процесс 3D Secure аутентификации с проверкой держателя карты. Данный набор полей отображается при значении поля **Статус транзакции** C - Необходима аутентификация CReq/CRes.

▼ Данные ARes

Статус транзакции	<input style="width: 90%;" type="text" value="C - Необходима аутентификация..."/>		
DS reference number	<input style="width: 90%;" type="text" value="ds_ref_number"/>		
ACS reference number	<input style="width: 45%;" type="text" value="acs_ref_number"/>	ACS operator ID	<input style="width: 45%;" type="text" value="operator_id"/>
ACS challenge mandated	<input style="width: 45%;" type="text" value="Y - CReq/CRes обязателен"/>	Тип аутентификации	<input style="width: 45%;" type="text" value="01 - Static"/>
Канал ACS(CReq)	<input style="width: 90%;" type="text" value="3ds2_acs_server"/>		Произвольный ACS URL <input type="checkbox"/>

Пример 13-7 Пример сообщения ARes для Challenge Flow

```

{
  "messageType" : "ARes",
  "messageVersion" : "2.1.0",
  "dsTransID" : "04f3512a-e838-48b0-baa2-3a077c4ea4e8",
  "threeDSSTransID" : "8a880dc0-d2d2-4067-bcb1-b08d1690b26e",
  "acsTransID" : "6a39f44a-fc26-4835-ae23-448a4503c353",
  "acsURL" : "http://localhost:7080/ext/3ds2_acs/test.3ds2_acs_server",
  "acsReferenceNumber" : "3DS_LOA_ACS_PPFU_020100_00009",
  "acsOperatorID" : "AcsOpId 4138359541",
  "dsReferenceNumber" : "DS_LOA_DIS_PPFU_020100_00010",
  "transStatus" : "C",
  "authenticationType" : "01",
  "acsChallengeMandated" : "Y"
}
    
```

В таблице 13-9 содержится описание полей сообщения ARes для Challenge Flow.

Таблица 13-9 Поля сообщения ARes для Challenge Flow

Поле	Описание
messageType	Константа ARes
messageVersion	Копируется из AReq
dsTransID	Уникальный идентификатор транзакции, установленный DS. Генерируется распондером в виде UUID
threeDSSTransID	Копируется из AReq
acsTransID	Уникальный идентификатор транзакции, установленный ACS. Генерируется распондером в виде UUID
acsURL	Настраивается на форме правила ответа. Составляется на основе данных канала, если заполнено поле Канал ACS(CReq) или берется из поля ACS URL , если установлен флажок Произвольный ACS URL

Поле	Описание
acsReferenceNumber	Значение поля ACS reference number на форме правила ответа
acsOperatorID	Значение поля ACS operator ID на форме правила ответа
dsReferenceNumber	Значение поля DS reference number на форме правила ответа
transStatus	Значение поля Статус транзакции на форме правила ответа
authenticationType	Значение поля Тип аутентификации на форме правила ответа
acsChallengeMandated	Значение поля ACS challenge mandated на форме правила ответа

Frictionless flow

Данный набор полей отображается при значениях поля **Статус транзакции** Y - Успешная аутентификация и A - Произведена попытка аутентификации.

▼ Данные ARes

Статус транзакции	<input type="text" value="Y - Успешная аутентификация"/>
DS reference number	<input type="text" value="ds_ref_number"/>
ACS reference number	<input type="text" value="123"/>
ACS operator ID	<input type="text" value="operator_id"/>

Пример 13-8 Пример положительного ARes для Frictionless Flow

```

{
  "messageType" : "ARes",
  "messageVersion" : "2.1.0",
  "dsTransID" : "14e695a1-bacd-4866-a9ba-bb9cd0de41b2",
  "threeDSSTransID" : "8a880dc0-d2d2-4067-bcb1-b08d1690b26e",
  "acsTransID" : "8e235c19-e1e2-4815-9ca4-04f48d19b9ac",
  "acsReferenceNumber" : "DS_LOA_DIS_PPFU_020100_00010",
  "acsOperatorID" : "AcsOpId 4138359541",
  "dsReferenceNumber" : "DS_LOA_DIS_PPFU_020100_00010",
  "transStatus" : "Y",
  "authenticationValue" : "AAABAWB1IwAAAAABVXUjAAAAAA=",
  "eci" : "05"
}

```

В таблице 13-9 содержится описание полей сообщения ARes для Frictionless Flow.

Таблица 13-10 Поля положительного ARes для Frictionless Flow

Поле	Описание
authenticationValue	Генерируется респондером на основе данных из группы Данные CAVV и acctNumber

Поле	Описание
eci	<p>Генерируется респондером в зависимости от диалекта и статуса транзакции.</p> <p>Visa:</p> <ul style="list-style-type: none"> • Y - Успешная аутентификация — 05 • A - Произведена попытка аутентификации - 06 <p>Mastercard, MIR:</p> <ul style="list-style-type: none"> • Y - Успешная аутентификация — 02 • A - Произведена попытка аутентификации - 01

Данный набор полей отображается при значениях поля **Статус транзакции** N - Неудачная аутентификация, U - Аутентификация не может быть выполнена, R - Аутентификация отклонена.

▼ Данные ARes

Статус транзакции	N - Неудачная аутентификация	Причина статуса транзакции	02 - Unknown device
DS reference number	ds_ref_number		
ACS reference number	acs_ref_number	ACS operator ID	operator_id

Пример 13-9 Пример негативного ARes для Frictionless Flow

```

{
  "messageType" : "ARes",
  "messageVersion" : "2.1.0",
  "dsTransID" : "14e695a1-bacd-4866-a9ba-bb9cd0de41b2",
  "threeDSSTransID" : "8a880dc0-d2d2-4067-bcb1-b08d1690b26e",
  "acsTransID" : "8e235c19-e1e2-4815-9ca4-04f48d19b9ac",
  "acsReferenceNumber" : "DS_LOA_DIS_PPFU_020100_00010",
  "acsOperatorID" : "AcsOpId 4138359541",
  "dsReferenceNumber" : "DS_LOA_DIS_PPFU_020100_00010",
  "transStatus" : "N",
  "transStatusReason" : "02"
}
    
```

Таблица 13-11 Поля негативного ARes для Frictionless Flow

Поле	Описание
transStatusReason	Значение поля Причина статуса транзакции на форме правила ответа

13.13.3 Примеры сообщений RReq

Пример 13-10 Пример обычного RReq

```

{
  "messageType" : "RReq",
}
    
```

```

"messageVersion" : "2.1.0",
"threeDSServerTransID" : "8a880dc0-d2d2-4067-bcb1-b08d1690b26e",
"acsTransID" : "3b590756-dc07-49ac-a177-e86be89d9acf",
"authenticationType" : "01",
"authenticationValue" : "AAABAWB1IwAAAAABVXUjAAAAAA=",
"dsTransID" : "50c4503d-60be-441d-843b-678ca98d797c",
"eci" : "05",
"interactionCounter" : "01",
"messageCategory" : "01",
"transStatus" : "Y"
}
    
```

Таблица 13-12 Поля обычного RReq

Поле	Описание
messageType	Константа RReq
messageVersion	Копируется из CReq
threeDSServerTransID	Копируется из CReq
acsTransID	Копируется из CReq
authenticationType	Копируется из AReq
authenticationValue	Генерируется респондером на основе данных из группы Данные CAVV и pan
dsTransID	Копируется из AReq
eci	Генерируется респондером в зависимости от диалекта и статуса транзакции. Visa: <ul style="list-style-type: none"> • Y - Успешная аутентификация — 05 • A - Произведена попытка аутентификации - 06 Mastercard, MIR: <ul style="list-style-type: none"> • Y - Успешная аутентификация — 02 • A - Произведена попытка аутентификации - 01
interactionCounter	Константа 01
messageCategory	Копируется из AReq
transStatus	Значение поля Статус транзакции RReq на форме правила ответа

Пример 13-11 Пример специфичного RReq

```

{
"messageType" : "RReq",
"messageVersion" : "2.1.0",
"threeDSServerTransID" : "8a880dc0-d2d2-4067-bcb1-b08d1690b26e",
"acsTransID" : "3b590756-dc07-49ac-a177-e86be89d9acf",
"authenticationType" : "01",
}
    
```

```

"dsTransID" : "50c4503d-60be-441d-843b-678ca98d797c",
"interactionCounter" : "01",
"messageCategory" : "01",
"transStatus" : "N",
"transStatusReason" : "01"
}
    
```

Таблица 13-13 Поля специфического RReq

Поле	Описание
transStatusReason	Значение поля Причина статуса транзакции RReq на форме правила ответа

13.13.4 Пример CRes

Пример 13-12 Пример специфического CRes

```

{
  "messageType" : "CRes",
  "threeDSServerTransID" : "8a880dc0-d2d2-4067-bcb1-b08d1690b26e",
  "acsTransID" : "3b590756-dc07-49ac-a177-e86be89d9acf",
  "acsTransID" : "50c4503d-60be-441d-843b-678ca98d797c",
  "messageVersion" : "2.1.0",
  "transStatus" : "N"
}
    
```

Таблица 13-14 Поля CRes

Поле	Описание
messageType	Константа CRes
threeDSServerTransID	Копируется из CReq
acsTransID	Копируется из CReq
messageVersion	Копируется из CReq
transStatus	Значение поля Статус транзакции CRes на форме правила ответа. Если активна функция Анализировать RRes , то определяется в зависимости от значения поля resultsStatus сообщения RRes: <ul style="list-style-type: none"> • 01 – Y - Успешная аутентификация • 02, 03 – N - Неудачная аутентификация

13.13.5 Просмотр журнала сообщений 3DS2

Журнал сообщений 3DS2 доступен для просмотра через меню (**Тестирование** → **Журнал сообщений 3DS2**).

Чтобы открыть журнал сообщений:

1. Выберите меню **Тестирование** → **Журнал сообщений 3DS2**) (см. рисунок 121).

Журнал сообщений 3DS2

Дата начала	Дата окончания	Окружение	Правило ответа	PAN	ACS	Тип	Статус
2022-10-12 14:06:38	2022-10-12 14:06:38	test	Frictionless_N			Merchant Callback	Завершено
2022-10-12 14:06:37	2022-10-12 14:06:37	test	Frictionless_N			Merchant Callback	Завершено
2022-10-12 14:06:37	2022-10-12 14:06:37	test	Frictionless_N			Merchant Callback	Завершено
2022-10-12 14:06:37	2022-10-12 14:06:37	test	Frictionless_Y	8944988785642183	b2be9e56-91a2-4eb4-9136-a1f50fa6d3e2	3DS2	Завершено
2022-10-12 14:06:37	2022-10-12 14:06:37	test	Frictionless_N	8944988785642185	7dca06bc-09d2-4b99-bc3b-a1ff6d11b9b1	3DS2	Завершено
2022-10-12 14:06:37	2022-10-12 14:06:38	test	Challenge_N_CRes	8944988785642186	f7716087-c507-409a-9eae-46b4a84a0344	3DS2	Завершено
2022-10-12 14:06:37	2022-10-12 14:08:38	test	Challenge_Y	8944988785642184	bc72714-d223-4143-8644-57ed6992e3c5	3DS2	Завершено
2022-10-12 14:06:37	2022-10-12 14:06:38	test	Challenge_N_RReq	8944988785642187	1ec00335-ecae-4b22-9003-3efaf6e56a10	3DS2	Завершено
2022-10-05 15:08:11	2022-10-05 15:08:11	test	Frictionless_N			Merchant Callback	Завершено
2022-10-05 15:08:08	2022-10-05 15:08:11	test	Challenge_N_RReq	8944988785642187	04c7501d-8fe5-4a20-890e-64f6c633da05	3DS2	Завершено
2022-10-04 18:50:45	2022-10-04 18:50:45	test	Frictionless_N			Merchant Callback	Завершено
2022-10-04 18:50:41	2022-10-04 18:50:45	test	Challenge_Y	8944988785642184	e450bc4c-96ab-4716-9ba4-023119f52191	3DS2	Завершено
2022-10-04 15:33:43	2022-10-04 15:33:43	test	Frictionless_N			Merchant Callback	Завершено

Рисунок 121 Журнал сообщений 3DS2

2. Найдите сообщение в списке и перейдите по ссылке в колонке **Дата начала**.

В журнале сообщений отображаются детали правила, входящее сообщение и сформированный респондером ответ (исходящее сообщение) в формате xml и json.

Детали

Правило ответа	<input type="text" value="challenge"/>	Дата начала	<input type="text" value="2022-09-22 00:09:48"/>	Дата окончания	<input type="text" value="2022-09-22 00:09:51"/>
Начальный BIN	<input type="text" value="8944988785642183"/>	Конечный BIN	<input type="text" value="8944988785642183"/>	Тип	<input type="text" value="3DS2"/>
Окружение	<input type="text" value="test"/>	Статус	<input type="text" value="Завершено"/>		

Таблица 13-15 Детали журнала сообщений 3DS2

Поле	Описание
Правило ответа	Правило ответа 3DS2, на основе которого формировались ответные сообщения
Дата начала	Дата начала обработки первого сообщения (AReq)
Дата окончания	Дата окончания формирования последнего сообщения (может быть как ARes, так и CRes)
Начальный BIN	Начальный BIN, который указан на форме правила ответа
Конечный BIN	Конечный BIN, который указан на форме правила ответа
Тип	Тип сообщения из журнала. Возможные значения: 3DS2, Merchant Callback
Окружение	Окружение серверного HTTP-канала, на котором обрабатывались сообщения

Поле	Описание
Статус	Текущий статус сообщения. Возможные значения: <ul style="list-style-type: none"> • Ожидает — AReq получен, сформирован ARes. Ожидается CReq • Ошибка — возникла непредвиденная ошибка. Можно проанализировать по логам, привязанным к конкретному сообщению из журнала. • Успешно — вся цепочка сообщений 3DS2 успешно обработана • Выполняется — респондер обрабатывает входящее сообщение

Сообщения AReq, ARes

Сообщение AReq

```

1- {
2  "threeDSRequestorID": "az0123456789",
3  "threeDSRequestorName": "Example Requestor name",
4  "notificationURL": "http://localhost:7082/ext/3ds2_requestor/test.3ds2_3dsRequestor_server",
5  "acquirerBIN": "868491",
6  "acquirerMerchantID": "mGm6AJZ1YotkJm0k0fx",
7  "addMatch": "N",
8  "cardExpiryDate": "1910",
9  "acctNumber": "8944988785642183",
10 "billAddrCity": "Bill City Name",
11 "billAddrCountry": "840",
12 "billAddrLine1": "Bill Address Line 1",
13 "billAddrLine2": "Bill Address Line 2",
14 "billAddrLine3": "Bill Address Line 3",
15 "billAddrPostCode": "Bill Post Code",
16 "billAddrState": "CO",
17 "email": "example@example.com",
18 "homePhone": {
19   "cc": "123",
20   "subscriber": "123456789"
21 },
22 "mobilePhone": {
23   "cc": "123",
24   "subscriber": "123456789"
25 },
26 "cardholderName": "Cardholder Name",
27 "shipAddrCity": "Ship City Name",
28 "shipAddrCountry": "840",
29 "shipAddrLine1": "Ship Address Line 1",
30 "shipAddrLine2": "Ship Address Line 2",

```

URL: http://127.0.0.1:7073/ext/3ds2_ds/test.3ds2_ds_server

Метод: POST

Наименование	Значение
Content-Type	application/json
user-agent	ReactorNetty/1.0.18
host	localhost:7073
accept	/*
content-length	4939

Сообщение ARes

```

1- {
2  "messageType": "ARes",
3  "messageVersion": "2.1.0",
4  "dsTransID": "83517a38-f722-4542-9312-4ee09db2075a",
5  "threeDSServerTransID": "8a880dc0-d2d2-4067-bcb1-b08d1690b26e",
6  "acsTransID": "bc34ca70-fe3b-4b1d-982c-0437869270e6",
7  "acsURL": "http://localhost:7080/ext/3ds2_acs/test.3ds2_acs_server",
8  "acsReferenceNumber": "acs_ref_number",
9  "acsOperatorID": "operator_id",
10 "dsReferenceNumber": "ds_ref_number",
11 "transStatus": "C",
12 "authenticationType": "01",
13 "acsChallengeMandated": "Y"
14 }

```

Статус: 200 - OK

Наименование	Значение
Content-Type	application/json

Эмуляция DS. Входящее сообщение AReq передается в формате JSON. Подбор правила ответа осуществляется по полю acctNumber входящего сообщения CReq среди активных каналов.

Для Challenge Flow формируется сообщение с полем acsURL, на который в дальнейшем необходимо отправить CReq для успешного завершения обработки транзакции. Также респондер генерирует уникальное значение acsTransID, которое передается в CReq для формирования цепочки сообщений.

Для Frictionless Flow формируется обычное сообщение с полями authenticationValue и esi, которые являются подтверждением успешной аутентификации. В случае специфичного сообщения эти поля отсутствуют, но добавляется поле transStatusReason, которое отражает причину неуспешной аутентификации. В данном случае отправка дополнительного сообщения CReq не нужна.

Сообщения CReq, CRes

Сообщение CReq

```

1 - {
2   "ACSURL" : "https://3ds.ru/acs/brw/challenge?id=b0322961-fa7f-45d6-9a9d-04c11ad1507b",
3   "CREQ" : "eyJhdGdyZWVlbnRlcmlRyW5zSU01A1A0G40DBkYzAtZDJkMl00MDY3LWJjYjEYjA4ZDE2OTBhMjZl",
4   "THREEDSSSESSIONDATA" : "YTIxMTQ2NmMhIjZC000GZ1NFbGELMlQ1ZjMyMzUwMmRn",
5   "TERMURL" : "https://egwtest.ru/mp11/cgi-bin/cgi_link",
6   "ACTION" : "S",
7   "THREEDSSSERVERTRANSID" : "a2114673-6b7d-48c3-af0a-1d5f323e02da"
8 }
        
```

URL:

Метод:

Наименование	Значение
Content-Type	application/json
user-agent	ReactorNetty/1.0.18
host	localhost:7080
accept	*/*
content-length	594

Синхронный ответ на CReq

```

1 - <html>
2 - <head>
3   <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
4   <title>Return to Merchant's site</title>
5 - <script>
6     function onLoadHandler() {
7       document["CResForm"].submit();
8     }
9   </script>
10  </head>
11  <body onload="onLoadHandler();">
12  <br />
13  <br />
14  <br />
15  <form name="CResForm" method="post" action="http://localhost:7082/ext/3ds2_requestor/test.3d
16  <input name="notificationURL" type="hidden" value="http://localhost:7082/ext/3ds2_requestor
17  <input name="cres" type="hidden" value="eyJzZmZlYmVlbnRlcmlRyW5zSU01A1A0G40DBkYzAtZDJkMl
18  <input name="threeDSsessionData" type="hidden" value="YTIxMTQ2NmMhIjZC000GZ1NFbGELMlQ1ZjMyMzUwMmRn
19  </form>
20  </body>
21  </html>
        
```

Статус:

Наименование	Значение
Content-Type	text/html

Эмуляция ACS. Входящее сообщение CReq передается в формате JSON в закодированном виде (Base64 URL). Его необходимо раскодировать, чтобы получить CReq в формате JSON. По полю acsTransID выбирается сообщение из журнала, и связанное с ним правило ответа используется формирования CRes. Полученный CRes кодируется в Base64 URL. В ответе возвращается HTML страница с переадресацией на notificationURL, который берется из сообщения AReq. Однако перед отправкой ответа на CReq необходимо сформировать дополнительное сообщение RReq, отправить его на 3DS Server и получить ответ.

Сообщения RReq, RRes

Сообщение RReq

```

1 - {
2   "messageType" : "RReq",
3   "messageVersion" : "2.1.0",
4   "threeDSserverTransID" : "8a880dc0-d2d2-4067-bcb1-b08d1690b26e",
5   "acsTransID" : "bc34ca70-fe3b-4b1d-982c-0437869270e6",
6   "authenticationType" : "01",
7   "authenticationValue" : "AAM8AcycUAAAAAAAAAABTJQAAAAAAAA",
8   "dsTransID" : "83517a38-f722-4542-9312-4ee09db2075a",
9   "eci" : "05",
10  "interactionCounter" : "01",
11  "messageCategory" : "01",
12  "transStatus" : "Y"
13 }
        
```

URL:

Метод:

Наименование	Значение
Content-Type	application/json

Сообщение RRes

```

1 - {
2   "threeDSserverTransID" : "8a880dc0-d2d2-4067-bcb1-b08d1690b26e",
3   "acsTransID" : "d7c1ee99-9478-44a6-b1f2-391e29c6b340",
4   "dsTransID" : "1jpe0dc0-i9t2-4067-bcb1-nmt866956sgd",
5   "messageType" : "RRes",
6   "messageVersion" : "2.1.0",
7   "resultStatus" : "02"
8 }
        
```

Статус:

Наименование	Значение
Content-Type	application/json
content-length	264

Эмуляция ACS и DS. Исходящее сообщение передается в формате JSON. В отличие от сообщений AReq и CReq респондер эмулирует не ответ на полученное сообщение, а формирование и отправку исходящего сообщения. В ответ приходит сообщение RRes в формате JSON от реального эквайера (3DS Server). Отправка осуществляется на URL, который указан в поле threeDSserverURL сообщения AReq с сертификатом канала,

который указан в группе **Данные CRes** в поле **Канал 3DS server(RReq)**. После получения ответного сообщения RRes или таймаута формируется CRes.

Сообщения Merchant Callback

Эмуляция Merchant. Входящее сообщение может быть в любом формате. В ответе возвращается сообщение с пустым телом и статусом с формы правила ответа.

Приложение А. Справочник контекстных функций

Приложение SOLAR Testing Toolkit поддерживает predefined список контекстных функций, а также функции плейсхолдеров, реализованные в платформе SOLAR.

Синтаксис обращения к контекстной функции: `${function}`.

А.1 Общие контекстные функции платформы SOLAR

А.1.1 add

```
add(value1, value2)
```

Функция	Описание
<code>add(value1, value2)</code>	Арифметическая операция суммирования. Возвращает сумму значений параметров

А.1.2 caseOf

```
caseOf(String condition, String valueCondition1, String value1, String valueCondition2, String value2, .... String valueConditionN, String valueN)
```

Функция	Описание
<code>caseOf(condition, valueCondition1, value1, valueCondition2, value2, ..., valueConditionN, valueN)</code>	Проверяет условие, содержащееся в первом параметре, и возвращает значение, которое ему соответствует

Пример А-1 Использование функции caseOf

```
${caseOf('var2', 'var1', 'value1', 'var2', 'value2', 'var3', 'value3', 'var4', 'value4')}
```

Результатом выполнения функции является `value2`.

А.1.3 concat

```
concat(value1, value2)
```

Функция	Описание
<code>concat(value1, value2)</code>	Конкатенация строковых литералов

Пусть `var1 = null` (не задана), `var2 = '7'`.

Пример А-2 Использование функции concat

```
${concat('abc', 'qwe')} => abcqwe  
${concat('%07d:', var2)} => %07d:7  
${concat('abc', var1)} => abc
```

А.1.4 divide

```
divide(value1, value2)
```

Функция	Описание
<code>divide(value1, value2)</code>	Арифметическая операция деления. Возвращает частное от деления значения <code>value1</code> на <code>value2</code>

A.1.5 lowerCase

```
lowerCase(String value)
```

Функция	Описание
<code>lowerCase(value)</code>	Возвращает строку, приведенную к нижнему регистру

Пример A-3 Использование функции lowerCase

```
#{lowerCase('someExpression')}=someexpression
```

A.1.6 multiply

```
multiply(value1, value2)
```

Функция	Описание
<code>multiply(value1, value2)</code>	Арифметическая операция умножения. Возвращает произведение значений параметров

A.1.7 subtract

```
subtract(value1, value2)
```

Функция	Описание
<code>subtract(value1, value2)</code>	Арифметическая операция вычитания. Возвращает разность значений параметров

A.1.8 upperCase

```
upperCase(String value)
```

Функция	Описание
<code>upperCase(value)</code>	Возвращает строку, приведенную к верхнему регистру

Пример A-4 Использование функции upperCase

```
#{upperCase('someExpression')} => SOMEEXPRESSION
```

A.1.9 abs

```
abs(value)
```

Функция	Описание
<code>abs(value)</code>	Возвращает абсолютную величину значения параметра

Пример А-5 Использование функции abs

```
`${abs (value)}`
```

Результатом выполнения функции `abs` является 5, если `value` равно 5 или -5.

Обратите внимание, что запись унарных операций не поддерживается. Вызов ``${abs (-5)}`` является некорректным. Для передачи отрицательного числа используйте переменную.

А.1.10 addDays

```
addDays (String date, Integer days)
```

Функция	Описание
<code>addDays(date, days)</code>	Прибавляет к дате указанное количество дней

Пример А-6 Использование функции addDays

```
`${addDays (dateTime ('2021-01-31T05:00'), 10)}` = `${dateTime ('2021-02-10T05:00')}`
```

А.1.11 dateTime

```
dateTime ()
dateTime (String dateTimeValue)
```

Функция	Описание
<code>dateTime</code>	Возвращает текущую дату, если вызывается без параметров, или конвертирует строку в дату

Пример А-7 Использование функции dateTime

```
`${dateTime ()}`
`${dateTime ('2021-01-01T00:00')}`
```

А.1.12 format

```
format (value, parameters)
```

- `value` — данные для форматирования,
- `parameters` — параметры форматирования.

Параметры зависят от типа данных. Например, для чисел — округление, ведущие нули и т.д., для дат — форматирование в соответствии с ISO.

Таблица А-1 Параметры форматирования чисел

Параметр	Описание	Пример
<code>%d</code>	Целое число	«5»
<code>+%d</code>	Целое число с указанием знака	«+5»
<code>%3d</code>	Целое число длиной 3 символа	«_ _5»

Параметр	Описание	Пример
%03d	Целое число длиной 3 символа с заполнением пустых мест нулями	«005»
%f	Вещественное число с количеством знаков после разделителя по умолчанию – 6	«3.500000»
%.1f	Вещественное число с одним знаком после разделителя	«3.5»
%5.1f	Вещественное число длиной 5 символов с одним знаком после разделителя	« _3.5»

```
${format(value, parameters)}
```

A.1.13 listOf

```
listOf(String value1, String value2, ... String value N)
```

Функция	Описание
listOf(value1, value2, ... value N)	Возвращает список своих аргументов в строковом формате. Рекомендуется использовать при необходимости передать на сервер аргумент в виде списка

Пример A-8 Использование функции listOf

```
${listOf('a', 'b')}
```

Результатом выполнения функции listOf является список из элементов "a", "b".

A.1.14 padCenter

```
padCenter(String text, Integer length, String padChar)
```

Функция	Описание
padCenter(text, length, padChar)	Центрирует текст в строке фиксированной длины length, заполняя результирующую строку слева и справа символом padChar (по умолчанию его значение — пробел). Если не удастся разместить текст строго по центру, справа добавляется на один символ больше.

```
${padCenter('ab', 4, '_')} => _ab_
${padCenter('ab', 5, '_')} => _ab__
${padCenter('abcd', 2, '_')} => abcd
${padCenter(null, 4, '_')} => _____ (четыре подчеркивания)
${padCenter(null, 4, ' ')} => _____ (четыре пробела)
```

A.1.15 padLeft

```
padLeft(String text, Integer length, String padChar)
```


Функция	Описание
<code>padLeft(text, length, padChar)</code>	Возвращает строку, дополненную слева символами-заполнителями до поля указанной длины. Символ-заполнитель по умолчанию — пробел

```

${padLeft('ab', 4, '_')} => __ab
${padLeft('abcd', 2, '_')} => abcd
${padLeft(null, 4, '_')} => ____ (четыре подчеркивания)
${padLeft('', 4)} =>      (четыре пробела)

```

A.1.16 padRight

```
padRight(String text, Integer length, String padChar)
```

Функция	Описание
<code>padRight(text, length, padChar)</code>	Возвращает строку, дополненную справа символами-заполнителями до поля указанной длины. Символ-заполнитель по умолчанию — пробел

```

${padRight('ab', 4, '_')} => ab__
${padRight('abcd', 2, '_')} => abcd
${padRight(null, 4, '_')} => ____ (четыре подчеркивания)
${padRight('', 4)} =>      (четыре пробела)

```

A.1.17 repeat

```
repeat(String inputText, Integer times)
```

Функция	Описание
<code>repeat(inputText, times)</code>	Повторяет переданную строку <code>inputText</code> количество раз, равное <code>times</code>

```

${repeat('a', 3)} => aaa
${repeat('abc', 3)} => abcabcabc

```

A.1.18 substring

```
substring(String value, Integer n, Integer m)
```

Функция	Описание
<code>substring(value, n, m)</code>	Подстрока указанной строки с позиции <code>n</code> до <code>m</code> , исключая правую границу. Нумерация начинается с 0

```

${substring('0123456789', 2, 7)} => '23456'

```

A.1.19 subtractDays

```
subtractDays(date, Integer days)
```

Функция	Описание
<code>subtractDays(date, days)</code>	Вычитает из даты указанное количество дней

```
${subtractDays(dateTime('2021-01-31T05:00'), 10)} = ${dateTime('2021-01-21T05:00')}
```

A.2 Основные контекстные функции SOLAR Testing Toolkit

В данном разделе описаны основные контекстные функции, доступные в приложении.

A.2.1 base64Decode

```
base64Decode(String value)
```

Функция	Описание
<code>base64Decode</code>	Декодирование шестнадцатеричного значения по алгоритму Base64. Возвращает декодированное шестнадцатеричное значение

Пример A-9 Использование функции base64Decode

```
${base64Decode('54334A705A326C75595777675457567A6332466E5A513D3D')} => 4F726967696E616C204D657373616765
```

A.2.2 base64Encode

```
base64Encode(String value)
```

Функция	Описание
<code>base64Encode</code>	Кодирование шестнадцатеричного значения по алгоритму Base64. Возвращает закодированное шестнадцатеричное значение

Пример A-10 Использование функции base64Encode

```
${base64Encode('4F726967696E616C204D657373616765')} => 54334A705A326C75595777675457567A6332466E5A513D3D
```

A.2.3 convert

```
convert(value, String source_encoding, String target_encoding)
```

Функция	Описание
<code>convert(value, 'source_encoding', 'target_encoding')</code>	<ol style="list-style-type: none"> Преобразование шестнадцатеричных значений в строковые и наоборот, где: <ul style="list-style-type: none"> <code>value</code> — значение в исходной кодировке: числовой литерал, строковый литерал или переменная; <code>'source_encoding'</code> — исходная кодировка, <code>'target_encoding'</code> — результирующая кодировка. Конвертирование из json в xml и наоборот

Пример A-11 Использование функции convert

```

${convert(3638, 'hex', 'ascii')} => 68
${convert('F6F8', 'hex', 'ebcdic')} => 68
${convert(68, 'ebcdic', 'hex')} => F6F8
${convert(68, 'ascii', 'hex')} => 3638
${convert(myValue, 'json', 'xml')}
${convert(myValue, 'xml', 'json')}

```

A.2.4 deflateDecode

```
deflateDecode(String value)
```

Функция	Описание
deflateDecode (value)	Декодирование шестнадцатеричного значения по алгоритму Deflate. Возвращает декодированное шестнадцатеричное значение

Пример A-12 Использование функции deflateDecode

```

${deflateDecode('789CF32FCA4CCFCC4BCC51F04D2D2E4E4C4F050033BD061B')} =>
4F726967696E616C204D657373616765

```

A.2.5 deflateEncode

```
deflateEncode(String value)
```

Функция	Описание
deflateEncode (value)	Кодирование шестнадцатеричного значения по алгоритму Deflate. Возвращает закодированное шестнадцатеричное значение

```

${deflateEncode('4F726967696E616C204D657373616765')} =>
789CF32FCA4CCFCC4BCC51F04D2D2E4E4C4F050033BD061B

```

A.2.6 random

```

random()
random(int maxValue)

```

Функция	Описание
random()	Случайное значение из диапазона [0; Long.MAX_VALUE]
random(int maxValue)	Случайное значение из диапазона [0; maxValue]

A.2.7 sequence

```
sequence()
```

Функция	Описание
sequence ()	Возвращает следующее значение глобальной в рамках приложения последовательности целых чисел из диапазона [0; Long.MAX_VALUE]. Используется для генерации уникального значения
sequence (int modulus)	Возвращает остаток от деления следующего значения глобальной последовательности на значение параметра modulus

Пример A-13 Использование функции sequence

```

${sequence()}
...
${sequence()}

```

Предположим, первый вызов `sequence()` выдает значение 239. Тогда следующий вызов функции выдаст 240.

Пример A-14 Использование функции sequence с параметром

```

${sequence(1000000)}

```

Вызов функции в примере A-14 выдает целое значение из диапазона 0...999999, например 241.

Для получения шестизначного значения можно применить форматирование.

Пример A-15 Использование функции sequence с параметром и форматированием

```

${sequence(1000000):%06d}

```

`${sequence(1000000):%06d}` выдает целое шестизначное значение. Если остаток от деления значения последовательности на 1000000 меньше 6 символов, он дополняется нулями слева, например 000242.

A.2.8 sysdate

```

sysdate()

```

Функция	Описание
sysdate ()	Возвращает текущую дату. Может быть использована для генерации уникального в рамках тестового скрипта значения, например, RRN (Retrieval Reference Number) транзакции или номера XML-сообщения

Пример A-16 Использование функции sysdate

```

${sysdate():ddMMYYYY_HHmmsSSS}

```

A.2.9 unicodeDecode

```

unicodeDecode (String value)

```

Функция	Описание
unicodeDecode ()	Принимает символ, закодированный как hex, decimal, html и возвращает раскодированный символ

Пример A-17 Использование функции unicodeDecode

```
Hello${unicodeDecode('&nbsp;')}World! => Hello World!
Hello${unicodeDecode('&#xa0;')}World! => Hello World!
Hello${unicodeDecode('&#160;')}World! => Hello World!
```

A.2.10 urlDecode

```
urlDecode(String value)
```

Функция	Описание
urlDecode	Декодирование строки по алгоритму URL encode. Возвращает декодированное значение

A.2.11 urlEncode

```
urlEncode(String value)
```

Функция	Описание
urlEncode	Кодирование строки по алгоритму URL encode. Возвращает закодированное значение

A.2.12 uuid

```
uuid()
```

Функция	Описание
uuid()	Возвращает UUID-строку версии 4, например 123e4567-e89b-12d3-a456-426655440000

Пример A-18 Использование функции uuid

```
${uuid() }
```

A.2.13 xPath

```
xPath (String xmlString, String xPathExpression)
```

Функция	Описание
xPath (xmlString, xPathExpression)	<p>Применяет выражение xPath к строке XML.</p> <ul style="list-style-type: none"> xmlString — строка XML, xPathExpression — выражение xPath. <p>Может использоваться совместно с convert (myValue, 'json', 'xml') для разбора и анализа json-столбцов выборки</p>

Пример A-19 Использование функции xPath

```
${XPath('<tag>text</tag>', '/tag')} => text
```

A.3 Контекстные функции для тестирования ISO 8583

A.3.1 isoGenerateField

```
isoGenerateField(String value)
```

Функция	Описание
isoGenerateField	Генерация данных для полей ISO 8583 в виде случайной последовательности заглавных латинских букв и цифр длиной 6 символов. В текущей версии поддерживается только поле 38. Функция доступна при наличии в конфигурации приложения диалекта ISO 8583

Пример A-20 Использование функции isoGenerateField

```
${isoGenerateField(38)}  
${isoGenerateField('38')}
```

A.4 Контекстные функции для тестирования 3DS 1.0

В данном разделе описаны функции для тестирования 3DS 1.0.

A.4.1 pares

```
pares(pan, encodedPaReq, status, eci, cavvAlgorithm, cavvValue)
```

Функция	Описание
pares(pan, encodedPaReq, status, eci, cavvAlgorithm, cavvValue)	Генерация XML-представления PARES (Payer Authentication Response). <ul style="list-style-type: none"> pan — Primary Account Number, encodedPaReq — encoded Payer Authentication Request, status — статус транзакции, eci — Electronic Commerce Indicator, cavvAlgorithm — алгоритм CAVV, cavvValue — значение CAVV. <p>См. пример A-22, содержащий результат вызова функции</p>

Пример A-21 содержит значение параметра encodedPaReq, используемое в вызове функции pares.

Пример A-21 Значение параметра encodedPaReq

```
<ThreeDSecure>  
  <Message id="id_23002931_2334_4d13_bef2_001a8e56d59b">  
    <PAReq>  
      <version>1.0.2</version>  
      <Merchant>  
        <acqBIN>014565</acqBIN>  
        <merID>123321</merID>  
        <name>Test Merchant</name>
```

```

        <country>840</country>
        <url>https://example.com</url>
    </Merchant>
    <Purchase>
        <xid>DCpfekSLRT6JV0MOVmWtxQAAAAA=</xid>
        <date>20190219 14:12:07</date>
        <amount>6.0000000000</amount>
        <purchAmount>6.0000000000</purchAmount>
        <currency>840</currency>
        <exponent>0</exponent>
    </Purchase>
    <CH>
        <acctID>A0fTY+pKUTu/6hcZWZJiAA==</acctID>
        <expiry>****</expiry>
    </CH>
</PAReq>
</Message>
</ThreeDSecure>

```

Пример A-22 содержит результат вызова функции `paRes` с параметрами `pan=1234567890123456`, `encodedPaReq` из примера A-21, `status='Y'`, `eci='02'`, `cavvAlgorithm=1`, `cavvValue='AAABBEg0VhIOVniQEjRWAAAAAAA'`.

Пример A-22 Результат вызова функции `paRes`

```

<?xml version="1.0" encoding="UTF-8"?>
<ThreeDSecure>
  <Message id="id_23002931_2334_4d13_bef2_001a8e56d59b">
    <PARes id="paRes_id_23002931_2334_4d13_bef2_001a8e56d59b">
      <version>1.0.2</version>
      <Merchant>
        <acqBIN>014565</acqBIN>
        <merID>123321</merID>
      </Merchant>
      <Purchase>
        <xid>DCpfekSLRT6JV0MOVmWtxQAAAAA=</xid>
        <date>20190219 14:12:07</date>
        <purchAmount>6.0000000000</purchAmount>
        <currency>840</currency>
        <exponent>0</exponent>
      </Purchase>
      <pan>0000000000003456</pan>
      <TX>
        <time>20200903 09:53:56</time>
        <status>Y</status>
        <eci>02</eci>
        <cavvAlgorithm>1</cavvAlgorithm>
        <cavv>AAABBEg0VhIOVniQEjRWAAAAAAA=</cavv>
      </TX>
    </PARes>
    <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
      <SignedInfo>
        <CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315" />
        <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />

```

```

<Reference URI="#pares_id_23002931_2334_4d13_bef2_001a8e56d59b">
  <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
  <DigestValue>wvQIq5RNm/r3kVUBPt6aQw7jG2g=</DigestValue>
</Reference>
</SignedInfo>

<SignatureValue>qvRW902QS0scvdvBE/t/peF2aRgFophVXRgZ5mbnJXajGW+rk1M7/nqzvf3J+2CF
8Bjb3weAM5X4
pUN4rtTv2qBojqPp2ONBdnmlKdgB2sBMm6vDdCTMlWvmztGvjRCkfgH6A4kE10W8ZXJ6geZoofCX
58+jR2m+Cw1fsfnT8+SN4P+mr88CsFe4NYETK+eYBhOeOfn0Rl6txdv7UsW+zrYPdEEkXnhXneki
VuMEQ1H0hTFzklHG8/2Hrm6P0iijoiul6uFOmEM+WvuVzVzuPwuHPS+LbhyMM5NAa1D3ouIwNaXS
6oswqAW9XdMSGPubJJ8QA9XNjOtsDREQfFfsEg==</SignatureValue>
  <KeyInfo>
  <X509Data>

<X509Certificate>MIICxzCCAa+gAwIBAgIETvkRoZANBgkqhkiG9w0BAQsFADAUMRIWEAYDVQQDEw1
sb2NhbGhvc3Qw
HhcNMTg5MDAzMTYwMzUzWhcNNDYwMjE4MjYwMzUzWjAUMRIWEAYDVQQDEw1sb2NhbGhvc3QwggEi
MA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQL7Ch020s8GIWYGViT2Ho66QC6803pIz8qFNPM
DHq6zvYuFX5/H/+Cyge1xy9BluEViNalpr6yTcVavJWVvnp2EgoNNv52IUBOCLQyenOGHAF4p7Rc
nROKqX6jj1hufIKAWi/gDxaZ/V0wrwzZuqrrhGV2+zOPpar4iYphvEcMQzvgufAhpif7LFDjAHjr
e2XdnWF5H6Vs1ceNkUAbYcTSKJS81/qYF429YG3yFLbCWzDIGGEaLj1AfynuQZ4dIyBbDErLZmKw
wylszKx9GsAOI+EX51p0gNk/Cv6y/oIjWWBN3Btf4kZ1ahepQyrn1jgZ7UF77vEdz3mYl8/1iEL5
AgMBAAGjITAfMBOGA1UdDgQWBBDdu84uw2gRQxKAXYV6uvoPHGOzrjANBgkqhkiG9w0BAQsFAAOC
AQEAwi4tDk+nTQVXCZusKx4pKW6mhDjwwB0eKaJ81XI/7k+POjrtrtEzSJKF3jsEZHmRnGwUMV5Un
dwCF7GpJH3q6jyR3zucFayhbz4YdaLWT6+I9lJobURbezQAB+1TFFHR0fqZZRBUvN7pEEzWxpA6
HgF3wHtpCUt0UT1XE7Xt2E3QPzW144kbgfW09NrVlJO1UF2Hbra88CcgjCe65oG8sTOe8uyGn44
tIDVRgK0Qc1fRrWA4ZoL7XCkqBaQzr33lS9OG3NjL2dHx9E/cFWOGjqI6wYiiaE6rTH8tdT+FrWu
U8F0VnZLyvEMwp8F7vVx+cIxTnBJOvGVAMQNLUudWw==</X509Certificate>
  </X509Data>
</KeyInfo>
</Signature>
</Message>
</ThreeDSecure>

```

A.4.2 pareqEncode

```
pareqEncode(String value)
```

Функция	Описание
---------	----------

pareqEncode(value) Кодирование сообщения PAREq (Payer Authentication Request)

```

${pareqEncode('<xml>text</xml>')} => eJyzqcyjNsStJrSix0QexACz3BYs=

```

Вызов `${pareqEncode('...')}` эквивалентен `${convert(base64Encode(deflateEncode(convert('...', 'ascii', 'hex'))), 'hex', 'ascii')}`.

A.4.3 paresDecode

```
paresDecode(String value)
```

Функция	Описание
---------	----------

paresDecode(value) Декодирование закодированного значения PAREs (Payer Authentication Response)


```
${paresDecode('eJyzqcjNsStJrSix0QexACz3BYs=')} => <xml>text</xml>
```

Вызов `${paresDecode('...')}` эквивалентен `${convert(deflateDecode(base64Decode(convert('...', 'ascii', 'hex'))), 'hex', 'ascii')}`.

A.5 Контекстные функции для тестирования 3DS 2.0

A.5.1 base64UrlDecode

```
base64UrlDecode(String rawData)
base64UrlDecode(byte[] rawData)
```

Функция	Описание
base64UrlDecode	Декодирование строки или строки в виде байтов по алгоритму Base64Url. Возвращает строку. Используется для декодирования creq/cres сообщений в рамках тестирования 3DS 2.x

A.5.2 base64UrlEncode

```
base64UrlEncode(String rawData)
base64UrlEncode(byte[] rawData)
```

Функция	Описание
base64UrlEncode	Кодирование строки или строки в виде байтов по алгоритму Base64Url. Возвращает закодированную строку. Используется для кодирования creq/cres сообщений в рамках тестирования 3DS 2.x

A.5.3 generateEcdhKeyPair

```
generateEcdhKeyPair()
```

Функция `generateEcdhKeyPair` предназначена для генерации пары ключей.

Функция	Описание
generateEcdhKeyPair ()	Возвращает приватный ключ и строковые представления x, y-координат публичного ключа

A.5.4 buildEcdhSharedSecretKey

```
buildEcdhSharedSecretKey(String sdkReferenceNumber, String pubKeyX, String
pubKeyY, privateKey)
```

Функция `buildEcdhSharedSecretKey` предназначена для генерации общего секретного ключа по протоколу ECDH.

**Таблица А-2 Параметры функции
buildEcdhSharedSecretKey**

Параметр	Описание
sdkReferenceNumber	Строковое значение идентификатора SDK — библиотеки для встраивания в мобильные приложения
pubKeyX	Строковое значение координаты x публичного ключа SDK
pubKeyY	Строковое значение координаты y публичного ключа SDK
privateKey	Приватный ключ (например, сгенерированный сервером ACS по протоколу ECDH) или его строковое представление

А.5.5 buildAcsSignedContent

```
buildAcsSignedContent(String acsAppChallengeUrl, String acsEphemPubKeyX, String acsEphemPubKeyY, String sdkEphemPubKeyX, String sdkEphemPubKeyY)
```

Функция buildAcsSignedContent предназначена для генерации значения поля acsSignedContent.

**Таблица А-3 Параметры функции
buildAcsSignedContent**

Параметр	Описание
acsAppChallengeUrl	Строковое значение URL сервера ACS, на который отправляется сообщение CReq
acsEphemPubKeyX	Строковое значение координаты x публичного ключа, сгенерированного сервером ACS
acsEphemPubKeyY	Строковое значение координаты y публичного ключа, сгенерированного сервером ACS
sdkEphemPubKeyX	Строковое значение координаты x публичного ключа, сгенерированного SDK
sdkEphemPubKeyY	Строковое значение координаты y публичного ключа, сгенерированного SDK

А.5.6 parseAcsSignedContent

```
parseAcsSignedContent(String value)
```

Функция parseAcsSignedContent предназначена для декодирования acsSignedContent.

Функция	Описание
parseAcsSignedContent(value)	Возвращает декодированное значение acsSignedContent

А.5.7 encodeAppCreq

```
encodeAppCreq(String creqJson, String jweAlgorithm, cek, cekLeftPart, Integer challengeAcsCounterAtoS, String acsTxnId)
```

Функция `encodeAppCreq` предназначена для кодирования сообщения `CReq`.

Таблица А-4 Параметры функции `encodeAppCreq`

Параметр	Описание
<code>creqJson</code>	Строковое представление <code>CReq</code>
<code>jweAlgorithm</code>	Строковая константа, которая означает используемый алгоритм JWE: <ul style="list-style-type: none"> • <code>A128CBC_HS256</code> или <code>A128GCM</code>
<code>cek</code>	Значение СЕК. Байтовый массив в формате шестнадцатеричной строки
<code>cekLeftPart</code>	Значение левой части СЕК. Байтовый массив в формате шестнадцатеричной строки
<code>challengeAcsCounterAtoS</code>	Значение счетчика <code>challengeAcsCounterAtoS</code>
<code>acsTxnId</code>	ID транзакции, назначенный сервером ACS

А.5.8 `decodeAppCreq`

```
decodeAppCreq(jsonBytes, cek, cekLeftPart)
```

Функция `decodeAppCreq` предназначена для декодирования байтового представления сообщения `CReq`.

Таблица А-5 Параметры функции `decodeAppCreq`

Параметр	Описание
<code>jsonBytes</code>	Байтовый массив закодированного сообщения <code>CReq</code> в формате шестнадцатеричной строки
<code>cek</code>	Значение СЕК. Байтовый массив в формате шестнадцатеричной строки
<code>cekLeftPart</code>	Значение левой части СЕК. Байтовый массив в формате шестнадцатеричной строки
<code>algorithm</code>	Строковая константа, которая означает используемый алгоритм JWE: <ul style="list-style-type: none"> • <code>A128CBC_HS256</code> или <code>A128GCM</code>
<code>creq</code>	Объект в <code>TypedMap</code> представлении, соответствующий <code>Creq</code> -сообщению

А.6 Контекстные функции модуля криптографии

А.6.1 `svс3`

Функция `svс3` предназначена для вычисления кода `svс3`. Параметры функции описаны в таблице А-6.

Таблица А-6 Параметры функции svc3

Параметр	Описание
ivcv3	Инициализационный вектор. Шестнадцатеричное значение длиной 2 байта, например 0001
unpredictableNumber	Числовое значение для повышения криптографической стойкости шифрования. Шестнадцатеричное значение длиной 4 байта, например 00000000
transactionCounter	Значение счетчика транзакций АТС (Application Transaction Counter). Шестнадцатеричное значение длиной 2 байта, например 0001
encryptionKey	Ключ шифрования

Пример А-23 Использование функции svc3

```
§{svc3('0D14', '00000899', '005E', '462FC416E0E93D042CD0B00731AB4637')}
```

Вызов функции svc3 в примере А-23 дает результат 9089.

А.6.2 deriveIccSessionKey

Функция deriveIccSessionKey предназначена для формирования секретного сессионного ключа (Icc Session Key). Параметры функции описаны в таблице А-7.

Таблица А-7 Параметры функции deriveIccSessionKey

Параметр	Описание
dialect	VISA, MC, MIR или контекстная переменная
version	Для диалекта VISA: 10, 14, 17 или 18. Для диалектов MC и MIR: пустая строка
transactionCounter	Значение счетчика транзакций АТС (Application Transaction Counter). Шестнадцатеричное значение длиной 2 байта, например 0001
iccMasterKey	Значение главного ключа (ICC Master Key), выдаваемое функцией deriveIccMasterKey
unpredictableNumber	Числовое значение для повышения криптографической стойкости шифрования. Шестнадцатеричное значение длиной 4 байта, например 00000000

А.6.3 deriveIccMasterKey

Функция deriveIccMasterKey предназначена для формирования главного ключа (Icc Master Key). Параметры функции описаны в таблице А-8.

Таблица А-8 Параметры функции `deriveIccMasterKey`

Параметр	Описание
<code>dialect</code>	VISA, MC, MIR или контекстная переменная
<code>version</code>	Для диалекта VISA: 10, 14, 17 или 18. Для диалектов MC и MIR: пустая строка
<code>pan</code>	Номер карты (PAN)
<code>panSequenceNumber</code>	Последовательный номер карты
<code>issuerMasterKey</code>	Значение ключа Issuer Master Key, выданное финансовой организацией

А.6.4 `encryptClearPinBlockIso0`

Функция `encryptClearPinBlockIso0` предназначена для шифрования пин-блока. Параметры функции описаны в таблице А-9.

Таблица А-9 Параметры функции `encryptClearPinBlockIso0`

Параметр	Описание
<code>pin</code>	Персональный идентификационный номер (Pin). 4-12 цифр
<code>pan</code>	Номер карты (PAN). По крайней мере 14 шестнадцатеричных символов, последний из которых — проверочный
<code>paddingCharacter</code>	Шестнадцатеричный символ-заполнитель (0-9, A-F). Обычно используется F
<code>encryptionKey</code>	Ключ шифрования для алгоритма 3DES (48 шестнадцатеричных символов)

А.6.5 `isoGenerateAC`

```
isoGenerateAC(String issuerMasterKey)
isoGenerateAC(String issuerMasterKey, Boolean generateKeyByEmv)
```

Функция `isoGenerateAC` предназначена для генерации зашифрованного текста.

Функция имеет обязательный аргумент `issuerMasterKey` и извлекает остальные необходимые параметры из данных ISO-сообщения, как описано в таблице А-10.

Необязательный аргумент `generateKeyByEmv` влияет на способ генерации ключа ICC Session Key, который формируется в процессе работы функции и влияет на результат:

- `generateKeyByEmv` равен `true` — ICC Session Key генерируется по стандарту EMV 4.2,
- отсутствует или равен `false` — ICC Session Key генерируется в соответствии со стандартом текущего диалекта.

Для генерации ICC Session Key по стандарту EMV 4.2 используется только ICC Master Key, полученный из Issuer Master Key, и Transaction Counter (9F36), а для диалектов Mastercard и Mir дополнительно используется Unpredictable Number (9F37).

Таблица А-10 Служебные параметры функции isoGenerateAC

Параметр	Источник
dialect	Из кода диалекта. Код диалекта ISO должен совпадать с именем соответствующего диалекта SOLAR ISO8583
version	Для диалекта VISA: третий байт тега "9F10": '0A' — 10, '0E' — 14. Для диалектов MC и MIR: пустая строка
issuerMasterKey	Секретный ключ Issuer Master Key для формирования главного ключа ICC Master Key
terminalData	Объединение значений тегов поля 55: "9F02", "9F03", "9F1A", "95", "5F2A", "9A", "9C", "9F37". Для диалекта MC в случае отсутствия в сообщении тега 9F03 подставляется нулевое значение, например 000000000000
iccData	Объединение значений тегов поля 55: "82", "9F36", "9F10". Для VISA версий 10, 14 и 18 из тега "9F10" берется только CVR (с 4 по 7 байт включительно), VISA 17 версии — с 5 по 6 байт

В диалекте VISA данные чипа и терминала иногда передаются не в тегах поля 55, а в отдельных трехзначных полях. Если в сообщении отсутствуют теги поля 55, функция isoGenerateAC извлекает данные чипа из отдельных полей. Информацию о соответствии тегов поля 55 и отдельных полей можно найти в спецификации платежной системы VISA. В примере А-24 показаны значение параметра issuerMasterKey и содержимое полей ISO сообщения.

Пример А-24 Использование функции isoGenerateAC

```
issuerMasterKey: 0123456789ABCDEF0123456789ABCDEF
DE02: 0123456789ABCDEF0123456789ABCDEF
DE23: 00
DE55:
01005E9F34034203009F3303E0F8C8950500800480009F37046E56718C9F101206010A03A0201A0A
020000000000321B8DB39F2608${isoGenerateAC('0123456789ABCDEF0123456789ABCDEF')}
9F3602000282027C009C01009F1A0206439A031510229F02060000000366005F2A020643
DE137: 0001
```

Поле 55 будет обновлено с учетом результата вызова функции isoGenerateAC, как показано в примере ниже:

```
DE55:
01005E9F34034203009F3303E0F8C8950500800480009F37046E56718C9F101206010A03A0201A0A
020000000000321B8DB39F260801F2577A6C2645D5
9F3602000282027C009C01009F1A0206439A031510229F02060000000366005F2A020643
```

А.6.6 rawGenerateAC

Функция rawGenerateAC предназначена для генерации зашифрованного текста на основании параметров, описанных в таблице А-11.

Таблица A-11 Параметры функции rawGenerateAC

Параметр	Описание
dialect	VISA, MC, MIR или контекстная переменная
version	Для диалекта VISA: 10, 14, 17 или 18. Для диалектов MC и MIR: пустая строка
iccSessionKey	Значение сессионного ключа, которое может быть получено при помощи функции deriveIccSessionKey
terminalData	Объединение значений тегов поля 55: "9F02", "9F03", "9F1A", "95", "5F2A", "9A", "9C", "9F37". Для диалекта MC в случае отсутствия в сообщении тега 9F03 подставляется нулевое значение, например 000000000000
iccData	Icc data

Вызов функции rawGenerateAC с параметрами, указанными в примере A-25, дает результат 775297A996052288.

Пример A-25 Использование функции rawGenerateAC

```
${rawGenerateAC('VISA', '14', '16F402C1CD45C207B65000A10BD5FE34',  
'00000000100000000000000000000007100000000000007101302050030901B6A',  
'3C00005503A4A082')}
```

Приложение В. Интерфейс saveParameterSetRowBatch

Интерфейс saveParameterSetRowBatch предназначен для загрузки строк со значениями наборов параметров из файла XML. Обратите внимание, что интерфейс не используется для создания набора параметров с нуля, загрузка значений осуществляется только для существующих наборов параметров. Для того чтобы создать набор параметров, используйте графический интерфейс приложения (см. раздел 4.6.1).

Для взаимодействия с интерфейсом необходимо подготовить файл в соответствии с описанным форматом и выложить его в определенную директорию под определенным именем.

Подробнее о настройке директории и шаблоне названия файла см. в разделе В.4.

В.1 Состав заголовка XML-сообщения

Заголовок XML-сообщения расположен в корневом элементе и содержит поля, описанные в таблице В-1.

Таблица В-1 Перечень полей в заголовке XML-сообщения

XPath	Тип значения	Обязательный	Описание
/protocol		✓	Информация о протоколе
/protocol/name	String	✓	Наименование протокола. Константа "solar-ws"
/protocol/version	String	✓	Версия протокола. Текущая версия: 2.0
/messageId	String		Идентификатор сообщения. Рекомендуется использовать значение в формате UUID
/messageDate	Datetime	✓	Дата и время создания сообщения
/originator		✓	Информация о системе источника сообщения

XPath	Тип значения	Обязательный	Описание
/originator/system	String	✓	Идентификатор системы источника сообщения (например, EXTERNAL)
/receiver			Информация о системе получателя сообщения
/receiver/system	String	✓	Идентификатор системы получателя сообщения

B.2 Состав тела сообщения saveParameterSetRowBatchRequest

Тело сообщения расположено в элементе saveParameterSetRowBatchRequest/body и содержит поля, перечисленные в таблице B-2.

Таблица B-2 Перечень полей тела сообщения saveParameterSetRowBatchRequest

XPath	Тип значения	Обязательный	Описание
/parameterSetRows		✓	Информация о строках наборов параметров
/parameterSetRows/parameterSetRow			Информация об одной строке набора параметров
/parameterSetRows/parameterSetRow/refNumber	String	✓	Идентификационный номер строки набора параметров
/parameterSetRows/parameterSetRow/code	String	✓	Код строки набора параметров
/parameterSetRows/parameterSetRow/name	String		Наименование строки набора параметров
/parameterSetRows/parameterSetRow/parameterSetRef		✓	Идентификационная информация набора параметров

XPath	Тип значения	Обязательный	Описание
/parameterSetRows/parameterSetRow /parameterSetRef/parameters			Параметры, идентифицирующие набор параметров
/parameterSetRows/parameterSetRow /parameterSetRef/parameters/code	String	✓	Код набора параметров
/parameterSetRows/parameterSetRow /parameters			Информация о параметрах, из которых состоит строка
/parameterSetRows/parameterSetRow /parameters/parameter			Информация об одном параметре строки. Повторяющийся элемент. Имя параметра находится в xml-атрибуте <code>variable</code> , значение — в xml-атрибуте <code>expression</code> .

В.3 Структура файла `saveParameterSetRowBatchRequest`

Протокол поддерживает загрузку XML-файла заданного формата (см. пример В-1).

Корневой элемент `saveParameterSetRowBatchRequest` содержит элементы `header` (заголовок сообщения) и `body` (тело сообщения).

Пример В-1 Пример файла `saveParameterSetRowBatchRequest.xml`

```
<saveParameterSetRowBatchRequest>
  <header>
    <protocol>
      <name>solar-ws</name>
      <version>2.0</version>
    </protocol>
    <messageId>2ca7f8f0-5c92-4619-b2f5-592ca75ad6de</messageId>
    <messageDate>2015-12-24T13:53:55</messageDate>
    <originator>
      <system>EXTERNAL</system>
    </originator>
  </header>
```

```
<body>
  <parameterSetRows>
    <parameterSetRow>
      <refNumber>ps_row1</refNumber>
      <code>row1</code>
      <parameterSetRef>
        <parameters>
          <code>ps</code>
        </parameters>
      </parameterSetRef>
      <parameters>
        <parameter variable="a" expression="{random(123)}"/>
        <parameter variable="b" expression="myValue"/>
      </parameters>
    </parameterSetRow>
  </parameterSetRows>
</body>
</saveParameterSetRowBatchRequest>
```

B.4 Настройка взаимодействия с интерфейсом saveParameterSetRowBatch

Чтобы настроить расположение директории, из которой приложение будет загружать файлы со значениями параметров, нужно определить значение переменной `slq/proof/saveParameterSetRowFiles/baseDir` в конфигурационном файле `context.xml`, расположенном в директории `./config` в рабочей директории приложения, в следующем формате:

```
<entry key="<variable_name>" value="<path_to_folder>"/>
```

Пример B-2 Настройка директории для загрузки файлов с набором параметров

```
<entry key="slq/proof/saveParameterSetRowFiles/baseDir" value="."/>
```

Если указать путь как в примере B-2 (`value="."`), будет использоваться директория `<APP_DIR>/data/incoming`, где `<APP_DIR>` — рабочая директория приложения.

Название файла должно подходить под маску `^saveParameterSetRow(.)*\\.xml$`, например, `saveParameterSetRow_001.xml`.

История изменений документа

Дата	Версия ПО	Описание
02.11.2022	3.1.0	Обновлены: <ul style="list-style-type: none">• Нагрузочное тестирование: использование наборов параметров возможно только на основных шагах сценария.• Наборы параметров: уточнена возможность обновления существующего набора параметров.• Клиринг: поддержка новых клиринговых диалектов, обновлены примеры, добавлено описание зависимостей в rom.xml
29.06.2023	3.1.0	Обновлены: <ul style="list-style-type: none">• Работа с диалектом «Сообщение операционной системе»: указана рабочая директория выполняемого скрипта SOLAR_DATA_DIR и описана настройка прав доступа к скрипту для текущего пользователя приложения
10.11.2023	3.1.0	Обновлены: <ul style="list-style-type: none">• Исправлены опечатки в примерах вызова функций