Нечкин Сергей Сергеевич

ssnechkin@gmail.com

Аннотация

Приложение, позволяющее быстро разрабатывать и выполнять сценарии тестирования без знаний языка программирования

RSAutotest

Автоматизация тестирования

Оглавление

[Введение 1](#_Toc520389726)

[СТРУКТУРА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ 1](#_Toc520389727)

[Редактор. Описание UI 2](#_Toc520389728)

[ЗАПУСК ПРИЛОЖЕНИЯ 2](#_Toc520389729)

[РЕДАКТОР ТЕСТОВ 3](#_Toc520389730)

[Создание файла с наборами тестов 4](#_Toc520389731)

[Добавление уже созданного ранее файла с тестами в редактор 5](#_Toc520389732)

[Удаление файла с тестами из редактора 5](#_Toc520389733)

[Управление запуском тестов 5](#_Toc520389734)

[Шаги сценария 8](#_Toc520389735)

[Связанные тесты 9](#_Toc520389736)

[ОТЧЁТЫ 9](#_Toc520389737)

# Введение

Приложение RSAutotes предназначено для автоматизации тестирования программного обеспечения. Разработано для людей, не имеющих знаний языка программирования. Позволяет разрабатывать и исполнять сценарии тестирования, написанные фразами на русском языке. Тесты выполняются в многопоточном режиме (параллельно).

Приложение позволяет выполнять тестирование:

* WEB-приложений
* Приложений PL/SQL на уровне SQL-запросов
* Интеграцию систем по средствам запросов SOAP, POST, GET
* Очередь IBM MQ
* FTP – загрузка скачивание, удаление файлов
* Манипуляции в командной строке

Список возможных вариантов тестирования состоит из встроенных в приложение модулей. Модули содержат типовые шаги (шаги должны быть указаны в коде приложения), которые могут быть использованы для создания тестовых сценариев. Имеется возможность расширять количество модулей в зависимости от потребности в добавлении новых шагов для тестирования системы.

В перспективе развития приложения RSAutotes планируется вынести встроенные модули в отдельные файлы-плагины, содержащие шаги тестовых сценариев, в разбивке по определённой тематике (т.е. по группам). Например, файл-плагин с шагами для тестирования LDAP.

Приложение разработано на языке Java, имеет встроенный WEB-редактор тестов, разработанный на JavaScript.

# СТРУКТУРА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

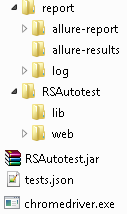
Приложение состоит из файла приложения RSAutotest.jar и директории RSAutotest.

Директория «RSAutotest» содержит:

* Web-редактор тестов (директория «web»);
* Java-библиотеки для обеспечения работы приложения «RSAutotest.jar» (директория «lib»);
* Файлы с настройками для модулей программы.

Приложение генерирует директорию «RSAutotest» при первом запуске.

После выполнения тестов в директории файла «RSAutotest.jar» приложение создает директорию «report», которая содержит файлы событий (директория «log») и отчёты о выполненных тестах (директории «allure-report» и «allure-results»).



Для работы встроенного модуля тестирования WEB-приложений необходимо добавить в директорию файла «RSAutotest.jar» драйвер браузера, в котором будут выполняться тесты. Драйвер браузера следует предварительно скачать. Например, «chromedriver.exe». Тесты будут храниться в файлах с расширением json или yaml.

# Редактор. Описание UI

В левой части редактора отображается дерево наборов и тестов. Примеры иконок, маркирующих объекты левой области редактора:

* C:\RSAutotest-2\target\RSAutotest\web\assets\images\suite.png Набор тестов
* C:\RSAutotest-2\target\RSAutotest\web\assets\images\test.pngТест

В правой части редактора отображается дерево сценария. Примеры иконок, маркирующих объекты правой области редактора:

* C:\RSAutotest-2\target\RSAutotest\web\assets\images\step.pngШаг сценария
* C:\RSAutotest-2\target\RSAutotest\web\assets\images\node.pngВетка сценария, содержащая шаги
* C:\RSAutotest-2\target\RSAutotest\web\assets\images\for.pngНачало или конец цикла выполняемых шагов
* C:\RSAutotest-2\target\RSAutotest\web\assets\images\data.pngДанные для теста

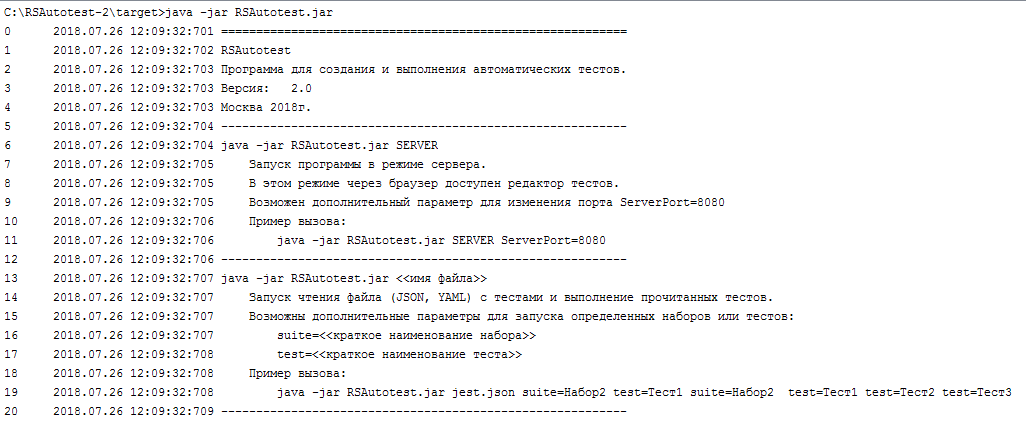
Объекты, имеющие флаг «Не выполнять», маркируются серыми иконками.

# ЗАПУСК ПРИЛОЖЕНИЯ

Предварительно в операционной системе должна быть установлена JAVA JRE начиная с версии 8. Для запуска приложения необходимо выполнить в командной строке Windows или Unix команду:

«java -jar RSAutotest.jar»

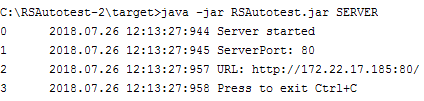
В консоли будут выведены примеры запуска тестов или редактора:



Чтобы запустить WEB-редактор, следует выполнить команду (указана в примерах запуска):

«java -jar RSAutotest.jar SERVER»

Приложение укажет в консоли URL-адрес WEB-редактора:



Чтобы запустить файл с тестами, следует выполнить команду (указана в примерах запуска):

«java -jar RSAutotest.jar имя\_файла.json»

По умолчанию приложение выполняет все тесты в указанном в команде файле.

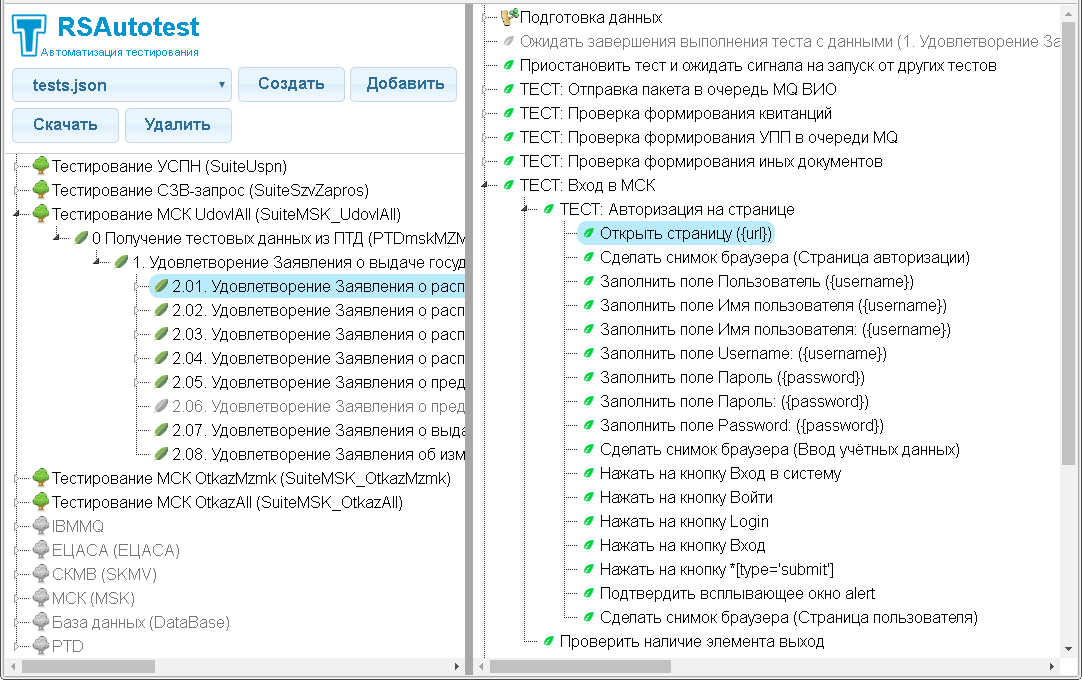
Для запуска определённых тестов или наборов тестов из файла необходимо указать команду с кратким наименованием набора тестов и/или теста:

«java -jar RSAutotest.jar имя\_файла.json suite=краткое\_имя\_набора test=краткое\_имя\_теста»

(при создании теста пользователь имеет возможность указать полное и краткое наименования теста или набора тестов).

# РЕДАКТОР ТЕСТОВ

Редактор тестов работает в браузере. После перехода по ссылке редактора в окне браузера отображается страница, которая содержит дерево наборов с тестами, дерево со сценариями тестов и управляющие элементы (кнопки «Создать», «Добавить», etc.).

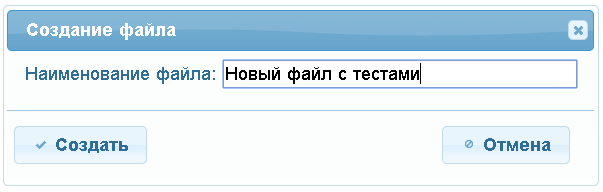


Пример загруженного редактора тестов

При запуске редактора тестов c аргументом SERVER (команда «java -jar RSAutotest.jar SERVER») в память программы загружаются файлы с тестами, которые хранятся в директории «edit\_web\_files» вместе с файлом «RSAutotest.jar». Загруженные файлы будут доступны для работы в редакторе.

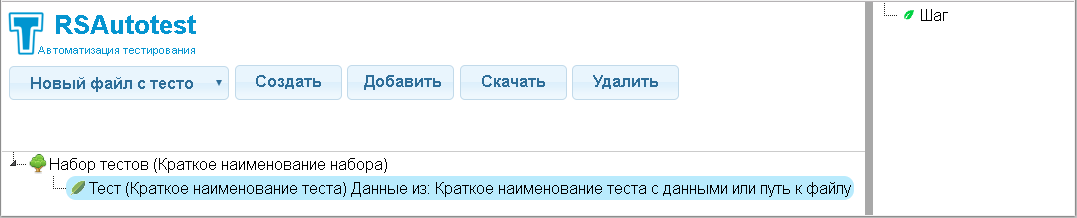
## Создание файла с наборами тестов

Нажмите на кнопку «**Создать**» для создания файла с наборами тестов. В открывшемся окне укажите наименование нового файла с тестами и нажмите на кнопку «Создать» для создания файла или «Отмена» для отмены действия:



Окно создания нового файла

После нажатия кнопки «Создать» создаётся файл в директории «edit\_web\_files», там же, где находится файл «RSAutotest.jar». В окне браузера будет указан пример дерева набора с тестом и одним шагом сценария.



Пример содержимого созданного файла

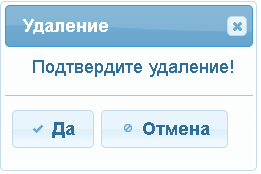
Наименование созданного файла добавляется в выпадающий список «Новый файл с тестом».

## Добавление уже созданного ранее файла с тестами в редактор

Нажмите кнопку «**Добавить**» для выбора существующего файла с тестами. В открывшемся окне выберите необходимый файл и нажмите кнопку «ОК». Наименование будет добавлено в выпадающий список «Новый файл с тестом», а набор и тесты, которые содержатся в этом файле, будут отображены в дереве наборов с тестами. Файл будет сохранен в директорию «edit\_web\_files».

## Удаление файла с тестами из редактора

Выберите в выпадающем списке файл с тестами, подлежащий удалению. Нажмите на кнопку «**Удалить**», а затем в открывшемся окне кнопку «Да» для подтверждения удаления файла из списка.

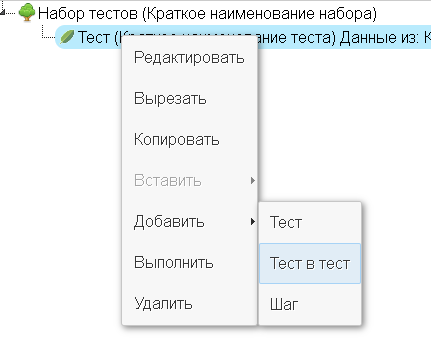


Окно удаления файла из редактора

После подтверждения файл удаляется из выпадающего списка и из директории «edit\_web\_files».

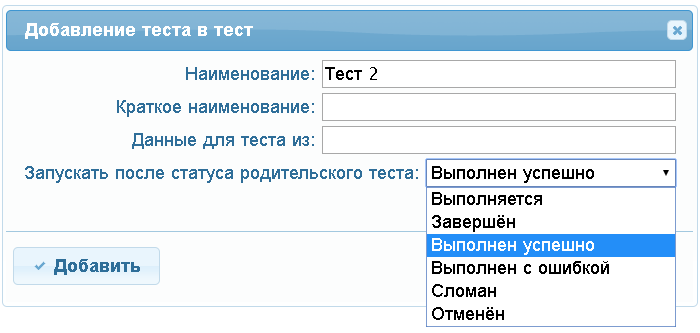
## Управление запуском тестов

По умолчанию в наборе тестов тесты запускаются параллельно. Для последовательного запуска тестов необходимо расположить тесты как ветки родительского теста в дереве. Для этого следует воспользоваться пунктом меню «Добавить» => «Тест в тест».

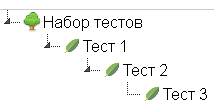


Пример пункта меню

В окне создания нового теста есть возможность выбрать статус родительского теста, при котором будет запущен дочерний тест.

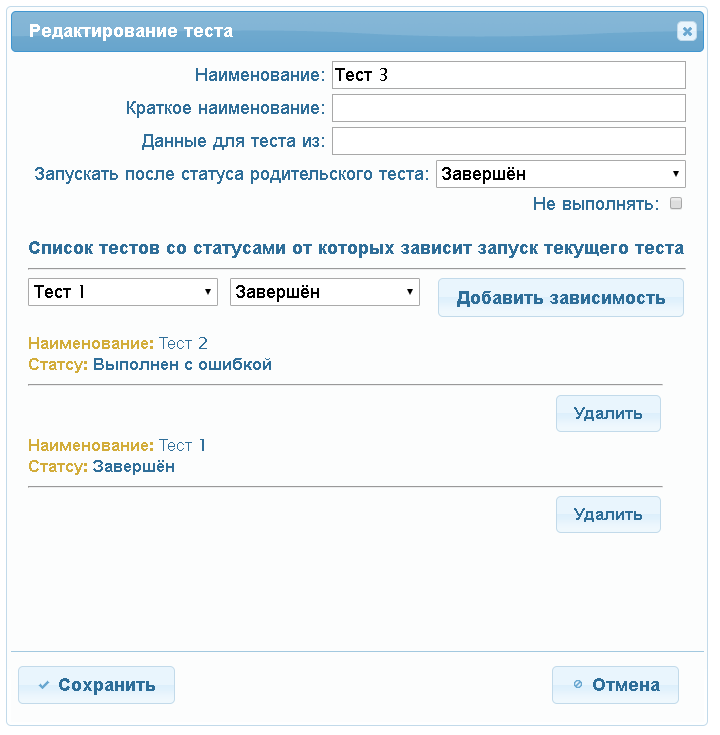


Пример окна для добавления теста в тест



Пример дерева с последовательным запуском тестов

Также предусмотрена установка зависимости запуска от нескольких тестов. В окне редактирования теста доступен блок для выбора родительских тестов и их статусов, при которых будет выполнен редактируемый тест.



Настройка условий запуска теста

Предусмотрена возможность приостановки и возобновления работы теста во время выполнения. Для приостановки теста в сценарий тестирования должен быть добавлен один из шагов сценария из группы шагов «Приостановка теста»:

* «Ожидать завершения выполнения теста с данными»
* «Ожидать положительного завершения выполнения теста с данными»
* «Ожидать отрицательного завершения выполнения теста с данными»
* «Приостановить тест и ожидать сигнала на запуск от других тестов»
* «Ожидание в миллисекундах»

Работа теста будет продолжена после завершения указанного теста в шаге. Для шага «Приостановить тест и ожидать сигнала на запуск от других тестов» любой из тестов в наборе должен содержать шаг «Разбудить тест с данными» с указанием наименования теста. После выполнения данного шага будет возобновлена работа теста. Следует соблюдать последовательность выполнения шагов: шаг «Приостановить тест и ожидать сигнала на запуск от других тестов» должен быть выполнен перед выполнением шага «Разбудить тест с данными».

## Шаги сценария

Для добавления шага в новый тест реализован пункт меню у теста «Добавить» => «Шаг». Добавление шага также возможно через пункт меню добавленного шага.

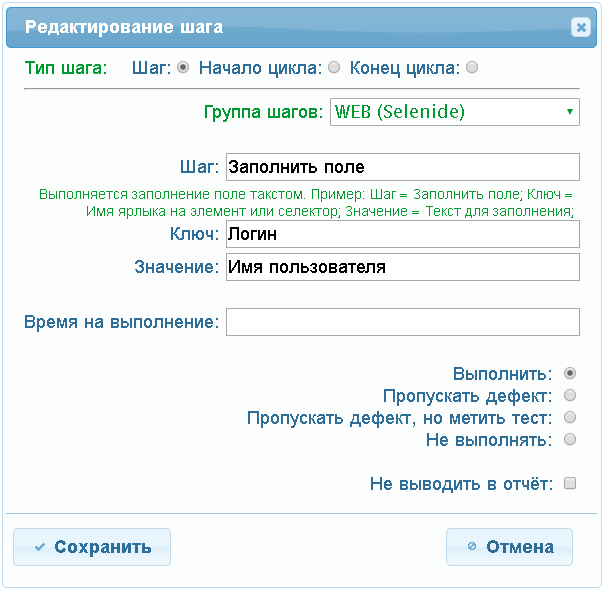
Шаг сценария представляет собой фразу, смысл которой обозначает выполняемое действие во время работы теста. Шаг может содержать аргументы – «Ключ» и «Значение».

Пример шага для перехода по ссылке в браузере: «Открыть страницу (http://yandex.ru)», где «http://yandex.ru» - это аргумент Значение.

Пример шага для заполнения поля на странице в браузере: «Заполнить поле Логин (Имя пользователя)», где «Логин» — это аргумент Ключ, а «Имя пользователя» - аргумент Значение.

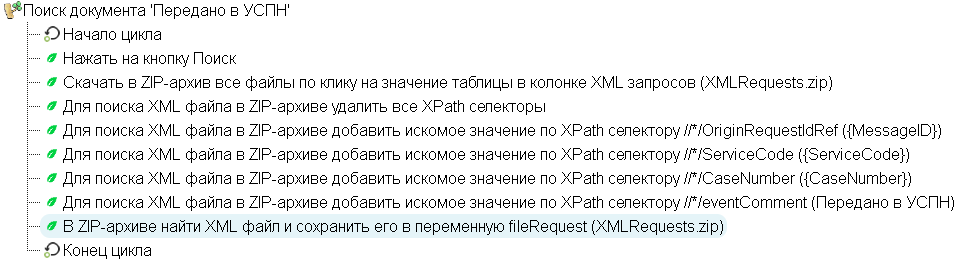
Пример шага для нажатия на кнопку в браузере: «Нажать на кнопку Войти», где «Войти» - это аргумент Ключ.

В редакторе шаги разделены на группы соответствующего модуля/плагина программы. Список групп и шагов доступен в окнах добавления/редактирования шага. При изменении или добавлении шага в окне отображается описание шага с примером заполнения ключа и значения.



Пример описания шага в окне редактирования шага

В приложении предусмотрена возможность выполнения нескольких шагов под циклом, для этого в окне добавления/редактирования шага нужно выбрать «Тип шага» «Начало цикла» и указать время на выполнение шагов или количество итераций цикла. Группа шагов цикла должны быть закрыта шагом с типом «Конец цикла».



Пример группы шагов, выполняющихся под циклом

В шагах возможно генерировать дату, для этого в аргументе «Ключ» или «Значение» нужно написать текст вида:

Пример: Шаг = Заполнить поле; Ключ = Имя\_поля; Значение = Datetime: +2h"

При выполнении будет сгенерирована строка: Значение = «текущая дата, увеличенная на 2 часа»

Заменяет CurrentDateTime на текущую дату. Формат может указываться в скобках CurrentDateTime(dd/MM/yyyy H:mm:ss)

Пример: Значение = CurrentDateTime(dd.MM.yyyy)

При выполнении будет сгенерирована строка Значение = 26.07.2018

При выполнении теста между шагами можно передавать данные через переменные, созданные в шагах. В шаге имена переменных нужно указывать в фигурных скобках. При выполнении программы переменные будут заменены содержимым переменной.

Пример шагов с переменными:

* Извлечь из ZIP-файла файл msk\RefMzop.json (Global\_PTD.zip)
* Из JSON файла сохранить значение url (msk\RefMzop.json)
* Открыть страницу ({url})
* Печать значения переменной url

## Связанные тесты

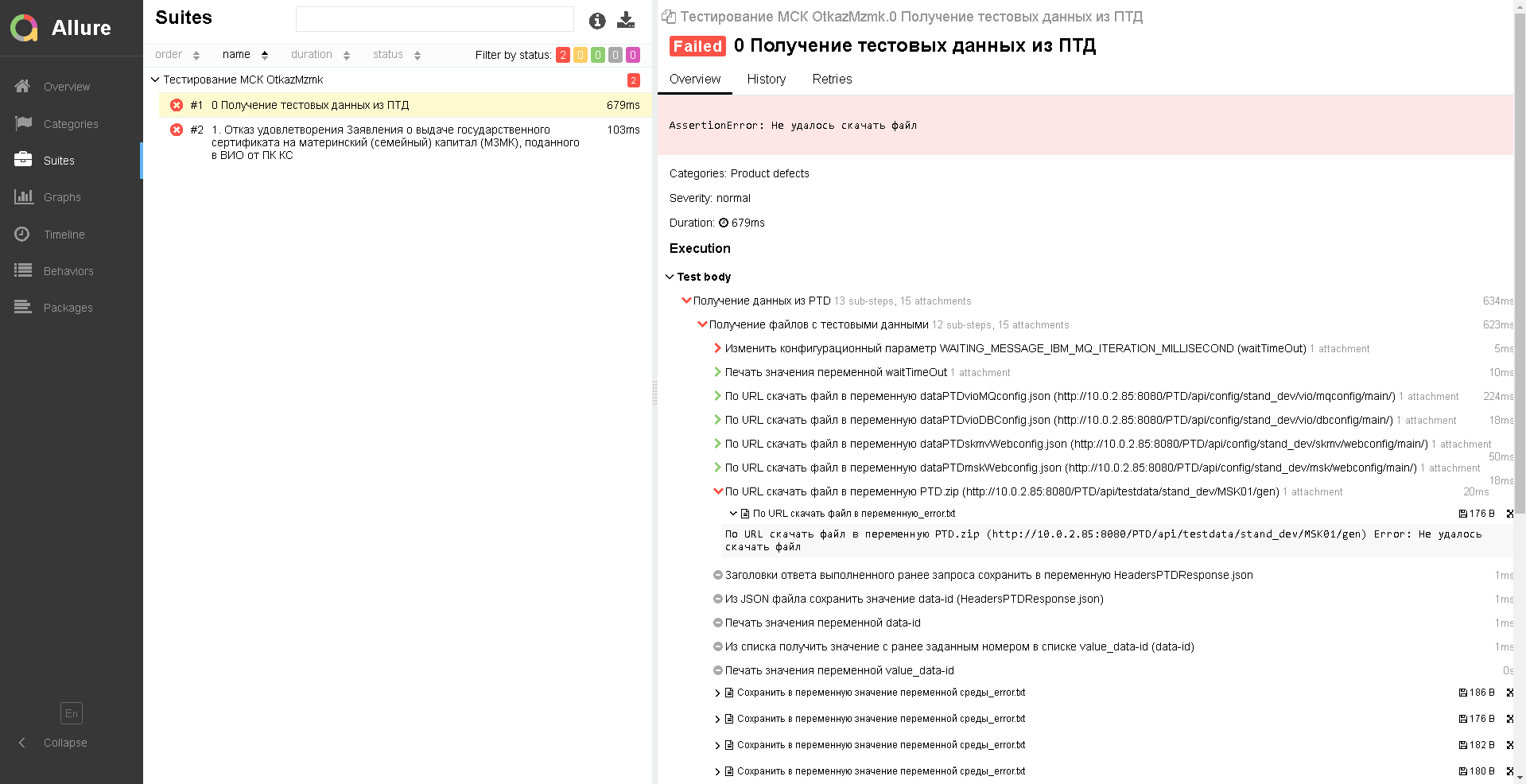
В шагах теста возможно добавлять шаги из других тестов в файле. Для этого в окнах добавления/редактирования шага в поле «Шаг» нужно указать краткое наименование теста.

В редакторе связанные шаги отобразятся в ветке с префиксом «ТЕСТ». Внутри ветки будут отображены шаги из связанного теста. Редактирование шагов приведёт к изменению шагов в связанном тесте.

Предусмотрена возможность загрузки шагов тестов из другого файла. Для этого в шаге нужно указать относительный или полный путь к внешнему файлу. Связанные шаги из теста внешнего файла не будут помечены префиксом «ТЕСТ» (поскольку файл является внешним), но будут выполнены в процессе работы теста.

# ОТЧЁТЫ

В приложении реализован модуль генерации отчётов Allure и IBM MQ. Возможно расширение приложения новыми модулями(плагинами) для генерации отчётов.



Пример сформированного Allure отчёта

http://172.22.17.185:53496/index.html#suites/e54725b901b34320b57fdbaef5a33cee/e923c8c5193482e7