



AppLine
Innovation

ООО «ИБС АППЛАЙН ИННОВАЦИИ»
ОГРН 5177746371237, ИНН/ КПП 7731394794/ 773101001

Россия, 121205, г.Москва, Территория Сколково Инновационного
Центра, Большой б-р, дом 42, строение 1, эт. 1 пом. 335 Р.М.08
телефон/факс: +7 (495) 967 80 80
info@apline.ru, www.apline-innovation.ru

QUAL IT

РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

Москва, 2023



ООО «ИБС АППЛАЙН ИННОВАЦИИ»
ОГРН 5177746371237, ИНН/ КПП 7731394794/ 773101001

Россия, 121205, г.Москва, Территория Сколково Инновационного
Центра, Большой б-р, дом 42, строение 1, эт. 1 пом. 335 Р.М.08
телефон/факс: +7 (495) 967 80 80
info@appline.ru, www.appline-innovation.ru

Оглавление

1. Назначение программы	3
2. Функциональные характеристики	3
2.1. Настройка программы	3
2.2. Разработка сценариев Cucumber	4
2.3. Вспомогательные методы	5



1. Назначение программы

Test IT - система управления тестированием. Система Test IT предназначена для хранения тестовой документации, планирования и выполнения работ по тестированию, сбора результатов ручных и автоматизированных тестов, анализа их результатов и построения отчетности.

Chameleon – фреймворк используемый для автоматизации тестирования программного обеспечения. Подключается к проектам, разрабатываемым на языке Java, с помощью средства автоматизации сборки.

Назначением модуля QualIT является отправка во время выполнения автотестов, написанных с использованием Chameleon, информации по пройденным шагам, в систему TestIT, для возможности использования функционала по сбору и анализу результатов выполнения автоматизированных тестов и построения отчетности в веб-интерфейсе TestIT.

2. Функциональные характеристики

2.1. Настройка программы

Для функционирования модуля необходимо выставить настройки. Для настройки доступны следующие параметры:

- url – URL сервера Test IT (**обязательный параметр**)
- privateToken – секретный ключ в профиле Test IT (**обязательный параметр**)
- projectId – id рабочего проекта в системе Test IT (**обязательный параметр**)
- configurationId – id конфигурации в рабочем проекте в системе Test IT (**обязательный параметр**)
- testRunId - id TestRun в который необходимо залить результаты (можно не задавать, если хотите создать новый)
- testRunName - имя TestRun (если не задать, то имя TestRun будет равно текущей даты запуска)
- проху - проху для сервера Test IT (можно не задавать)
- enabled - параметр для отправки результатов в Test IT: true - если хотите отправлять результаты в Test IT, false - если не хотите (можно не задавать, по умолчанию true)
- configFile - путь до конфигурационного файла json, в котором можно задать все настройки выше (можно не задавать - в таком случае будет искаться конфигурационный файл с именем по умолчанию)

Для настройки интеграции удобно использовать конфигурационный файл, его необходимо положить в каталог resources проекта.

Пример конфигурационного файла:

Листинг 2. Конфигурационный файл интеграции с TestIT

```
{
  "url": "URL сервера Test IT",
  "privateToken": "Секретный ключ в профиле Test IT",
  "projectId": "Id рабочего проекта в системе Test IT",
  "configurationId": "Id конфигурации в рабочем проекте в системе Test IT",
  "testRunId": "Id TestRun, в который необходимо залить результаты",
  "testRunName": "Имя TestRun",
  "проху": "Проху для сервера Test IT"
```



}

Параметры интеграции с TestIT также могут быть заданы с помощью параметров запуска, они имеют более высокий приоритет, перед параметрами из конфигурационного файла.

Пример задания параметров запуска:

```
Листинг 3. Настройка интеграции с TestIT через передачу параметров в  
параметры запуска  
-Dtestit.projectId=123 -Dtestit.configurationId=321 -  
Dtestit.privateToken=token -Dtestit.url=url -Dtestit.proxy=proxy  
-Dtestit.testRunName=name -Dtestit.testRunId=111 -Dtestit.enabled=true
```

В переменных окружения можно указать параметры аналогичным образом, но без ключа -D.

```
Листинг 4. Настройка интеграции с TestIT через переменные окружения  
testit.configurationId=321  
testit.configFile=config/dev_config.json
```

2.2. Разработка сценариев Cucumber

Модуль интеграции с Test IT для Cucumber позволяет составлять тестовые сценарии на языке Gherkin в feature файлах. В сценариях можно задавать метаданные через тэги:

- @ExternalId - ID автотеста в рабочем проекте системы Test IT (обязательный параметр).
- @WorkItemId - привязка автотеста к ручному тест-кейсу.
- @DisplayName - название автотеста в списке автотестов.
- @Description - описание в карточке автотеста.
- @Link - ссылки в карточке автотеста.
- @Label - тэги в карточке автотеста. Задаются через символ @ без знака равно.
- Сценарий - название в карточке автотеста. Задается не через тег, а в поле "Сценарий" в feature файле.

Пример тестового сценария:

Листинг 5. Тестовый сценарий Cucumber с тегами от модуля QualIT для интеграции с TestIT

Функция: Тестовая функция

```
@ExternalId=0001  
@Description=Очень_важный_тест  
@WorkItemId=1234  
@DisplayName=TestScript  
@Link=https://testit.ru @Link=https://ibs.ru  
@testLabel @newLabel @topLabel  
Сценарий: Тестовый сценарий  
  
* Авторизация  
  
* Проверка  
  
* Выход
```



При запуске тестов все собранные метаданные об автотестах и их запуске сохраняются, собираются и обрабатываются в формат пригодный для отправки через API Client. После сбора и преобразования данных они отправляются в Test IT.

2.3. Вспомогательные методы

Модуль также предоставляет пользователю вспомогательные методы для отправки данных в результаты автотестов:

- createAttachments() - массив, от 1 до N. Прикрепляемые к результату файлы.
- addLinks() - массив, от 1 до N. Ссылки в результатах автотеста.
- addMessage() - string, который будет дописываться, а в конце теста заливать собранное информативное сообщение об автотесте (в результатах автотеста), добавляя errorMessage (не stack trace) от тестового фреймворка.

Для использования метода createAttachments() необходимо импортировать класс AttachmentSteps и объявить его переменную через аннотацию @Autowired.

Для использования методов addLinks() и addMessage() необходимо импортировать класс TestITSteps и также объявить его переменную через аннотацию @Autowired.

Листинг 6. Настройка использования вспомогательных функций в Java-коде

```
import ru.ibsqa.chameleon.steps.AttachmentSteps;
import TestITSteps;

@Component
public class TestSteps {
    @Autowired
    private TestITSteps testITSteps;
    @Autowired
    private AttachmentSteps attachmentSteps;
}
```

Далее можно пользоваться методами, которые предоставляют классы TestITSteps и AttachmentSteps. Например:

Листинг 7. Использование вспомогательных методов в Java-коде

```
attachmentSteps.createAttachment(new File("путь к файлу"), "mimeType")

attachmentSteps.createAttachment("Имя_Файла.txt", "Текст, который будет в
файле", "text/plain");

testITSteps.addLink("https://www.ibs.ru/", "Title", "Description",
LinkType.RELATED);

testITSteps.addMessage("Шаг прошел успешно.");
```

В Cucumber сценариях данные вспомогательные методы можно использовать как в примере ниже:

Листинг 8. Использование вспомогательных методов в Cucumber-сценарии

- * добавить к результату теста файл "путь к файлу" с типом "image/jpeg"
- * добавить к результату теста сообщение "Тест завершился успешно!"