



# Платформа автоматизированного тестирования мобильных приложений “*bft-aT*”

## ИНСТРУКЦИЯ ПО УСТАНОВКЕ И ЭКСПЛУАТАЦИИ

### Оглавление

|                                        |   |
|----------------------------------------|---|
| 1. Изменения.....                      | 2 |
| 2. Требования к окружению .....        | 2 |
| 3. Подготовка к выполнению .....       | 2 |
| 4. Запуск на выполнение.....           | 3 |
| 5. Анализ результатов .....            | 3 |
| 6. Изменение параметров окружения..... | 4 |
| 7. Изменение тестовых данных.....      | 5 |
| 8. Пример Jenkins pipeline.....        | 6 |
| 9. Контакты.....                       | 7 |



## 1. Изменения

- Версия 2.0
- Тест 1\_2 2

## 2. Требования к окружению

1. ОС: Linux
2. Включена поддержка виртуализации (рекомендуется, чтобы данная поддержка была на аппаратном уровне): Для проверки можно использовать набор команд:  

```
sudo apt install cpu-checker kvm-ok
```
3. Docker версии не ниже 20.10
4. JDK версии 11
5. Установлены Docker образы (docker pull , требуется доступ в сеть Интернет, или скопировать эти образы вручную в соответствии с официальной документацией Docker):
  - a. Budtmo Android 11 (образ «budtmo/docker-android-x86-11.0»)
  - b. Budtmo Android 12 (образ «budtmo/docker-android-x86-12.0»)
6. Пользователь, от имени которого выполняются тесты, имеет права администратора
7. Не менее 10 ГБ свободного места
8. ОЗУ, не менее 16 ГБ
9. Версия тестируемого приложения (APK файл)

## 3. Подготовка к выполнению

1. Распаковать архив на выбранном ПК (например, директория «/user1-tests»).
2. Изменить параметры тестового стенда в соответствии с инструкцией ниже.
3. Изменить информацию о тестовых данных в соответствии с инструкцией ниже.



## 4. Запуск на выполнение

1. Запустить эмулятор командой:

```
emulator @Pixel_5_API_30
```

2. Запустить тест(ы) на выполнение командой:

для запуска всех тестов без выгрузки результатов в TestIt:

```
sh %WORK_DIR%/launch-tests-win.sh %WORK_DIR% false
```

для запуска тестов входящих в тест план с соответствующим номером (в TestIt):

```
sh %WORK_DIR%/launch-tests-win.sh %WORK_DIR% true <номер  
тест плана>
```

где %WORK\_DIR% – путь к директории проекта,

<номер тест плана> - номер тест плана в TestIt в соответствии с которым будут выполнены только тесты входящие в данный тест план.

Важно!!! Для запуска тестов по номеру тест плана в файле conf/Parameters.json необходимо указать testIt.url, testIt.providerId, testIt.username, testIt.password

## 5. Анализ результатов

По результатам выполнения формируется отчет, который доступен к просмотру с помощью команды:

```
allure serve %WORK_DIR%/build/allure-results
```

Также в случае включенной интеграции с TestIt информация о статусе выполнения будет записана в TestIt.



## 6. Изменение параметров окружения

Параметры окружения описываются в файле «conf/Parameters.json» и представляют собой JSON структуру:

- retryCount – количество перезапусков упавшего теста;
- useDocker – используется ли Docker для запуска тестов, true или false;
- apkPath – путь до тестируемого приложения с именем и расширением файла;
- deviceName – имя девайса для запуска тестов (в случае если Docker не используется);

chromeDriverExecutablePath – путь до вебдрайвера браузера chrome с именем и расширением файла;

envUrl – URL тестового сервера;

sessionTimeout – таймаут бездействия пользователя в тестируемом приложении, секунды;

pushCodeTimeout – таймаут на ввод пуш-кода в тестируемом приложении, секунды;

implicitlyWaitTimeout – таймаут ожидания элементов, секунды;

useMultiFactorAuthorization – используется ли на тестовом сервере двухфакторная авторизация, true или false;

existingLogin – логин, который занят в системе;

pushCode.validValue – корректное значение push кода, передаваемого в смс сообщениях пользователю;

pushCode.invalidValue – некорректное значение push кода, которое необходимо вводить для негативных тестов.

testIt.url – url адрес на который отправляются API запросы;

testIt.providerId - секретный ключ API необходимый для авторизации, инструкция по работе с ним на официальном сайте с документацией TestIt:

<https://docs.testit.software/user-guide/swagger-api-v2.0-experimental>;

testIt.username – логин в системе TestIt;

testIt.password – пароль в системе TestIt.

Прочие параметры остаются без изменения.

!Рекомендуется создать локальную копию измененного файла и впоследствии вносить изменения в нее при необходимости, а для выполнения выполнять полную замену файла из пакета доставки своей локальной копией.



## 7. Изменение тестовых данных

Информация о тестовых данных содержится в файле «conf/TestMapping.json» и представляет собой JSON массив «tests», в котором для каждого теста указывается информация о тестовых данных которые необходимо использовать:

id – ID теста в TestIt, не редактируется;

clients – массив данных о клиентах, необходимых для выполнения данного теста, количество элементов в массиве не должно изменяться;

clients[n].login – имя пользователя клиента;

clients[n].password – пароль клиента;

clients[n].accounts – массив номеров счетов клиента, которые необходимы для данного теста. Количество элементов в массиве не должно меняться;

clients[n].accounts[m].number – номер счета;

clients[n].accounts[m].cards – массив карт, привязанных к данному счету, количество элементов массива не изменяется;

clients[n].accounts[m].cards[k].alias – ассоциативный ID карты для внутреннего использования, не редактируется;

clients[n].accounts[m].cards[k].number – номер карты;

clients[n].accounts[m].cards[k].pin – пин код карты.

Также каждый элемент может иметь поле комментария (comment) с дополнительной информацией об элементе.

При редактировании файла важно помнить:

- Количество элементов в массивах не должно изменяться;
- Порядок следования тестов в массиве не имеет значения;
- Обозначенные выше нередатируемые поля не должны изменяться;
- Не все указанные выше поля присутствуют во всех элементах. Файл содержит только те данные для каждого теста, которые необходимы для его выполнения, например:
- Если в файле отсутствует номер счета, но описаны карты, то это значит, что у клиента должен быть счет и выпущенная к нему карта (карты), но сама информация о номере счета не имеет значения для данного теста

*!Рекомендуется создать локальную копию измененного файла и впоследствии вносить изменения в нее при необходимости, а для выполнения выполнять полную замену файла из пакета доставки своей локальной копией.*

## 8. Пример Jenkins pipeline

Перед запуском тестов на выполнение в Jenkins необходимо из командной строки запустить эмулятор следующей командой

```
emulator @Pixel_5_API_30
```

```
pipeline {
  agent {label 'windows'} //запуск на slave ноде

  parameters {
    choice(name: 'TESTIT_ENABLED', choices: ['false', 'true'],
description: '') //использование TestIt
    choice(name: 'TEST_PLAN_ID', choices: ['6'], description: '')
//выбор тест плана
  }

  environment {
    WORK_DIR = 'C:\\Users\\dynamica\\Desktop\\taf-win' //путь к
рабочей директории
  }

  stages {
    stage('Run tests') { //запуск тестов на выполнение
      steps {
        script {
          if (params.TESTIT_ENABLED == 'false'){
            bat 'sh %WORK_DIR%\\launch-tests-win.sh
%WORK_DIR% %TESTIT_ENABLED%'
          } else {
            bat 'sh %WORK_DIR%\\launch-tests-win.sh
%WORK_DIR% %TESTIT_ENABLED% %TEST_PLAN_ID%'
          }
        }
      }
      post {
        always {
          echo 'Generating allure report' //генерация отчета
Allure

          script {
            allure([
              includeProperties: false,
              jdk: '',
              properties: [],
              reportBuildPolicy: 'ALWAYS',
              results: [[path: 'build/allure-results']]
            ])
          }
          echo 'Cleaning workspace'
          cleanWs()
        }
      }
    }
  }
}
```



## 9. Контакты

Вопросы по установке/ эксплуатации ПО: [service@bft-it.ru](mailto:service@bft-it.ru)