ФИЛОСОФИЯ.ИТ
РОСАТОМ

Общество с ограниченной ответственностью «Философия.ИТ»
Россия, 107023, г. Москва, ул. Измайловский Вал, д. 30
Тел.: +7 (495) 988-37-38, факс: +7 (495) 988-37-38,
e-mail: customer@fil-it.ru
ИНН 7713728490
КПП 771901001
ОГРН 1117746379145

ДОКУМЕНТАЦИЯ, СОДЕРЖАЩАЯ ИНФОРМАЦИЮ, НЕОБХОДИМУЮ ДЛЯ ЭКСПЛУАТАЦИИ ЭКЗЕМПЛЯРА ПО

УНИВЕРСАЛЬНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ АВТОМАТИЗИРОВАННОГО ТЕСТИРОВАНИЯ ЦИФРОВЫХ СИСТЕМ «ТАЙМЫР.АВТОТЕСТИРОВАНИЕ» (TAYMYR.AUTOTEST)

г. Москва

Оглавление

1. Общие положения

1.1. Полное наименование программы для ЭВМ, обозначение

Полное наименование Программы для ЭВМ: «Универсальное программное обеспечение для автоматизированного тестирования цифровых систем "Таймыр.Автотестирование" (Taymyr.Autotest)» – далее по тексту Система.

Краткое наименование (обозначение) Системы: Таймыр.Автотестирование или Taymyr.Autotest.

1.2. Назначение системы

Таймыр.Автотестирование представляет собой систему для автоматизированного тестирования цифровых платформ. Зоны охвата автоматизации: Web, API, мобильные приложения. ПО предназначено для реализации и запуска автоматизированных тестов, а также формирования отчетности по результатам запуска. ПО служит для уменьшения трудозатрат тестирования в цикле разработки.

1.3. Состав системы

Система состоит из 4 частей:

- Модуль для автоматизации тестирования Web. Предназначен для написания автотестов, которые делают проверки в Web-интерфейсе.
- Модуль для автоматизации тестирования API. Предназначен для написания автотестов, которые делают проверки API.
- Модуль для автоматизации тестирования мобильных приложений. Предназначен для написания автотестов, которые делают проверки в мобильных приложениях.
- Общий модуль с реализаций функционала, который используют другие модули.

1.4. Разработчик системы

Полное наименование: Общество с ограниченной ответственностью «Философия.ИТ».

Сокращенное наименование: ООО «Философия.ИТ».

1.5. Назначение документа

Настоящий документ входит в комплект эксплуатационной документации по Таймыр.Автотестирование и содержит информацию, необходимую для эксплуатации предоставленного экземпляра ПО.

2. Настройка Системы для запуска

2.1. Настройки Appium для создания фермы устройств для запуска Mobile-автотестов

Настройка для сервера Appium

Настройка сервера Appium производится путем редактирования конфигурационного файла ".appiumrc.json" в корне модуля mobile-tests проекта автотестирования.
Структура и возможные параметры конфигурационного файла сервера Appium описаны на странице руководства Appium.

Запуск сервера Appium

Для запуска сервера Appium необходимо перейти в директорию проекта mobile-tests проекта автотестирования и выполнить команду запуска "npx Appium".
После этого станет доступен сервис Device Farm, позволяющий работать с локально подключенными к серверу устройствами и предоставляющий функции мониторинга прохождения тестов.

Device Farm

Адрес: http://127.0.0.1:4723/device-farm.
Представляет из себя инструмент для управления группой как виртуальных, так и реальных устройств Android или iOS.
Плагин настроен на опрос подключений виртуальных/реальных устройств Android и реальных устройств iOS.
Настройка плагина производится путем редактирования конфигурационного файла файла ".appiumrc.json" в корне модуля mobile-tests проекта автотестирования.
С возможными конфигурационными параметрами можно ознакомиться в документации плагина.

На вкладке «Devices» представлены все подключенные к серверу устройства (пример на **Рисунок 1**). Все ранее подключенные устройства автоматически будут видны в Device Farm. К этим устройствам можно подключаться (Use Device) или выводить их из пула доступных устройств для тестирования (Block Device).

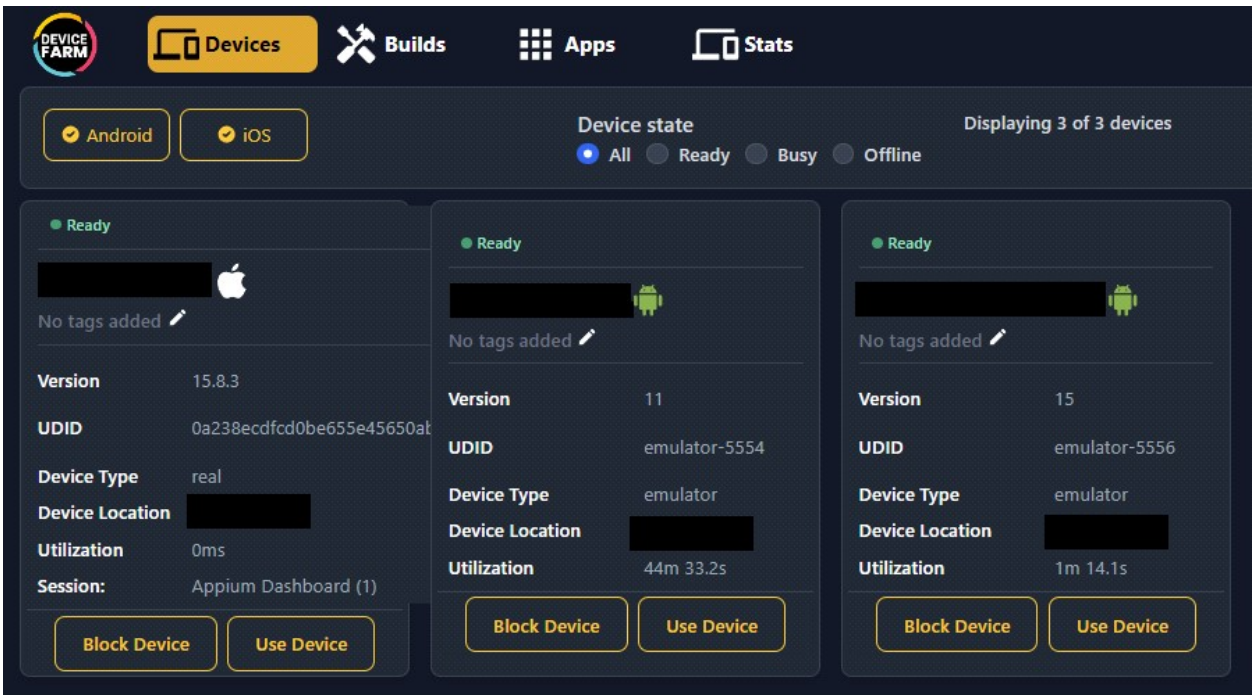

Рисунок 1

Для дальнейшего использования в автотестах этих устройств необходимы их уникальные идентификаторы UDID.

На вкладке «Apps» можно загружать и хранить все тестируемые приложения (пример на **Рисунок 2**).

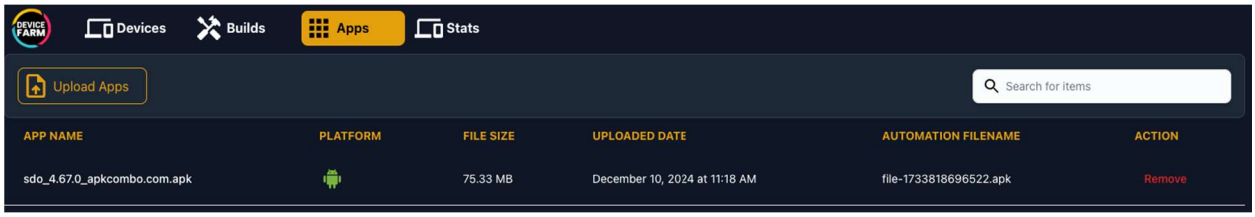

Рисунок 2

Для дальнейшего использования в автотестах этих приложений необходимы их уникальные имена из колонки «AUTOMATION FILENAME». На **Рисунок 2** можно увидеть список загруженных приложений.

Сбросить информацию обо всех сессиях можно командой:
“npx appium plugin run device-farm reset”.

2.2. Для локального запуска

Для настройки комплекса в локальном режиме запуска на рабочем месте необходимо сделать следующее:

- Открыть проект в IDE.

- Перейти в корень директории api-tests.
- Создать файл .env.
- Заполнить необходимыми параметрами файл .env для API (жирным шрифтом выделены обязательные параметры):
    - **MODULE**=type – тип автотестов (API, WEB, MOBILE).
    - **BASE_URL**=https://… – url окружения, на котором необходимо запустить тесты (например, https://digitalleague.ru/).
    - **TAGS**=@tag – тег сценариев, которые необходимо запустить (например, @test).
    - SEQUENCE_TAGS=@tag – сценарии, помеченные этими тегами, будут запускаться параллельно, независимо от значения параметра PARALLEL_COUNT (если параметр не указывать, то по умолчанию не будет ограничений по параллельному прохождению сценариев). Использование данного параметра может быть полезно, когда, например, 2 сценария изменяют одно и тоже поле в базе данных разными значениями. При параллельном прохождении сценарии могут влиять друг на друга и тесты не будут стабильными. Чтобы явно указать, что эти сценарии должны запускаться последовательно, необходимо оба сценария пометить одним и тем же тегом и указать этот тег в параметре SEQUENCE_TAGS. В случае, если есть необходимость в указании нескольких таких наборов сценариев, теги в параметре указываются через запятую (например, SEQUENCE_TAGS=@tag1,@tag2).
    - PARALLEL_COUNT=1 – количество параллелей запусков сценариев (если параметр не указывать, то по умолчанию значение будет 0 и, как следствие, все сценарии будут запущены последовательно).
    - RETRY_COUNT=1 – количество перезапусков сценариев в случае их падения с тегом @flaky (если параметр не указывать, то по умолчанию значение будет 1).

  В случае если необходим запуск с записью отчета в ReportPortal, следует добавить следующие параметры:
    - REPORT_PORTAL_API_KEY=token – токен для взаимодействия с ReportPortal (в случае если токен утерян, можно сформировать новый в настройках профиля ReportPortal).
    - REPORT_PORTAL_ENDPOINT=http://… – url ReportPortal.

- REPORT_PORTAL_PROJECT=name – название проекта для записи отчета в ReportPortal (если параметр не указывать, то по умолчанию значением будет название проекта “autotest”).
- REPORT_PORTAL_LAUNCH=name – название запуска в отчете ReportPortal (если параметр не указывать, то по умолчанию значением будет API_TEST_EXAMPLE).
- REPORT_PORTAL_DESCRIPTION=description – описание запуска, который будет отображаться в ReportPortal (если параметр не указывать, то по умолчанию значением будет “Тестовый запуск”).
- REPORT_PORTAL_MODE=name – название директории, в которую будет записан отчет в ReportPortal. Возможны 2 варианта: DEBUG или DEFAULT (если параметр не указывать, то по умолчанию будет выбран вариант DEBUG).

В случае необходимости использования, следует добавить следующие параметры:

- ELASTICSEARCH_URL=url – url elasticsearch (если нужен для тестов).
- ORACLE_PASSWORD=pswd – пароль от OracleDB (если нужен для тестов).
- ORACLE_URL=url – url OracleDB (если нужен для тестов).
- CASSANDRA_PASSWORD=pswd – пароль от CassandraDB (если нужен для тестов).
- CASSANDRA_USERNAME=username – username от CassandraDB (если нужен для тестов).
- CASSANDRA_URL=url – url от CassandraDB (если нужен для тестов).
- KAFKA_BROKER=config – конфигурация для доступа к Kafka (если нужен для тестов).
- REDIS_URL= url – url от Redis (если нужен для тестов).

- Перейти в корень директории web-tests.
- Создать файл .env.
- Заполнить необходимыми параметрами файл .env для WEB (находится в корне директории web-tests):
  - **MODULE**=type – тип автотестов (API, WEB, MOBILE).

- **BASE_URL**=https://… - url окружения, на котором необходимо запустить тесты (например, https://digitalleague.ru/)
- **TAGS**=@tag – тег сценариев, которые необходимо запустить (например, @test).
- RUN_CHROME=true – необходимость запускать только в Chrome (true/false). Значение по умолчанию – true.
- RUN_MULTI_CHROME=true – флаг, отвечающий за необходимость запустить несколько браузеров для одной сессии. Используется только для сценария “Проверка сессии после выхода”, при запуске других сценариев, параметр можно не передавать, либо передавать со значение false.
- MAX_INSTANCES-количество параллелей (количество поднятых docker-контейнеров с Chrome). Значение по умолчанию – 1.
- DRY_RUN=true – вариация запуска тестов. Варианты (true/false), где true – запускать тесты без прогона, а false – обычный запуск тестов.

В случае если необходим запуск с записью отчета в ReportPortal, следует добавить следующие параметры:

- REPORT_PORTAL_API_KEY=token – токен для взаимодействия с ReportPortal (в случае если токен утерян, можно сформировать новый в настройках профиля ReportPortal).
- REPORT_PORTAL_ENDPOIN=http://… – url ReportPortal.
- REPORT_PORTAL_LAUNCH=name – название запуска в отчете ReportPortal (если параметр не указывать, то по умолчанию значением будет “Тестовый запуск”).
- REPORT_PORTAL_MODE=name – название директории, в которую будет записан отчет в ReportPortal. Возможны 2 варианта: DEBUG или DEFAULT (если параметр не указывать, то по умолчанию будет выбран вариант DEBUG).
- REPORT_PORTAL_PROJECT=name – название проекта для записи отчета в ReportPortal (если параметр не указывать, то по умолчанию значением будет название проекта “autotest”).
- REPORT_PORTAL_DESCRIPTION=description – описание запуска, который будет отображаться в ReportPortal (если параметр не указывать, то по умолчанию значением будет “Тестовый запуск”).

В случае необходимости использования, следует добавить следующие параметры:

- ELASTICSEARCH_URL=url – url elasticsearch (если нужен для тестов).
- ORACLE_PASSWORD=pswd – пароль от OracleDB (если нужен для тестов).
- ORACLE_URL=url – url OracleDB (если нужен для тестов).
- CASSANDRA_PASSWORD=pswd – пароль от CassandraDB (если нужен для тестов).
- CASSANDRA_USERNAME=username – username от CassandraDB (если нужен для тестов).
- CASSANDRA_URL=url – url от CassandraDB (если нужен для тестов).
- KAFKA_BROKER=config – конфигурация для доступа к Kafka (если нужен для тестов).
- REDIS_URL= url – url от Redis (если нужен для тестов).

- Перейти в корень директории mobile-tests.
- Создать файл .env.
- Заполнить необходимыми параметрами файл .env для Mobile (находится в корне директории mobile-tests):
  - **MODULE**=type – тип автотестов (API, WEB, MOBILE).
  - **TAGS**=@tag – тег сценариев, которые необходимо запустить (например, @test).
  - **APPIUM_HOSTNAME**=hostname – адрес appium-сервера.
  - **MOBILE_APP_ID**=id – идентификатор мобильного приложения (например, ru.fil_it.sdo_liga).
  - DRY_RUN=true – вариация запуска тестов. Варианты (true/false). true – запускать тесты без прогона. false – обычный запуск тестов.
  - **UDID_ARRAY**=[] – массив устройств, на которых нужно запускать тесты. Например, [{"platformName": "Android","UDID": "emulator-5535","app": "example.apk"}]. Параметры UDID и app описаны в разделе “ Настройки Appium для создания фермы устройств. Device Farm” (в интерфейсе Device Farm представлены как параметры UDID и AUTOMATION FILENAME соответственно).

Данная конфигурация является автоматизацией конфигурации WebdriverIO для того, чтобы не приходилось настраивать множество параметров посредством внесения данных в файл wdio.conf.appium.ts репозитория автотестов. При этом появляется возможность конфигурации через переменные окружения. Для детального понимания всех параметров данной конфигурации ниже приведена таблица с описанием параметров WebdriverIO и комментарий с описанием параметра или с описанием алгоритма автоматизации данного параметра.

Параметр	**Платформа**	**Описание**
platformName	Android, iOS	Значение нужно задать (Android или iOS в зависимости от платформы).
appium:automationName	Android, iOS	Выбирается автоматически на основе указанного параметра platformName
appium:udid	Android, iOS	Уникальный идентификатор устройства. Можно посмотреть на вкладке «Devices» плагина Farm Devices.
appium:app	Android, iOS	Польный путь к приложению. В данном случае имя предварительно загруженного приложения в плагине Farm Devices на вкладке «Apps» из столбца «AUTOMATION FILENAME».
appium: appPackage	Android	Уникальный идентификатор приложения (подставляется значение параметра MOBILE_APP_ID, который описан выше.
appium:autoLaunch	Android, iOS	Нужно ли запускать приложение автоматически

Параметр	Платформа	Описание
		после запуска теста. Так как уже запрограммированный шаг, самостоятельно запускающий приложение, то автозапуск отключен (значение false).
df:build	Android, iOS	Имя группировки сборки на панели сборок Device Farm. Автоматически проставляется значение параметра platformName.
df:recordVideo	Android, iOS	Включить запись и сохранение видео прохождения теста. Всегда включено (значение true).
appium:bundleId	iOS	Уникальный идентификатор приложения (подставляется значение параметра MOBILE_APP_ID, который описан выше.
appium:autoAcceptAlerts	iOS	Автоматически принимать все всплывающие оповещения. Всегда включено (значение true).

Другие возможные параметры конфигурирования (их задание не автоматизировано и для применения нужно будет менять код фреймворка) можно посмотреть в официальной документации WebdriverIO, драйвера UiAutomator2 (Android), драйвера XCUITest (iOS) и плагина Device Farm.

В случае если необходим запуск с записью отчета в ReportPortal, следует добавить следующие параметры:

- REPORT_PORTAL_API_KEY=token – токен для взаимодействия с ReportPortal (в случае если токен утерян, можно сформировать новый в настройках профиля ReportPortal).
- REPORT_PORTAL_ENDPOIN=http://… – url ReportPortal.
- REPORT_PORTAL_LAUNCH=name – название запуска в отчете ReportPortal (если параметр не указывать, то по умолчанию значением будет “Тестовый запуск”).
- REPORT_PORTAL_MODE=name – название директории, в которую будет записан отчет в ReportPortal. Возможны 2 варианта: DEBUG или DEFAULT (если параметр не указывать, то по умолчанию будет выбран вариант DEBUG).
- REPORT_PORTAL_PROJECT=name – название проекта для записи отчета в ReportPortal (если параметр не указывать, то по умолчанию значением будет название проекта “autotest”).
- REPORT_PORTAL_DESCRIPTION=description – описание запуска, который будет отображаться в ReportPortal (если параметр не указывать, то по умолчанию значением будет “Тестовый запуск”).

В случае необходимости использования, следует добавить следующие параметры:

- ELASTICSEARCH_URL=url – url elasticsearch (если нужен для тестов).
- ORACLE_PASSWORD=pswd – пароль от OracleDB (если нужен для тестов).
- ORACLE_URL=url – url OracleDB (если нужен для тестов).
- CASSANDRA_PASSWORD=pswd – пароль от CassandraDB (если нужен для тестов).
- CASSANDRA_USERNAME=username – username от CassandraDB (если нужен для тестов).
- CASSANDRA_URL=url – url от CassandraDB (если нужен для тестов).
- KAFKA_BROKER=config – конфигурация для доступа к Kafka (если нужен для тестов).
- REDIS_URL= url – url от Redis (если нужен для тестов).

2.3. Для серверного запуска

Для настройки комплекса в серверном режиме запуска на сервере необходимо сделать следующее:

- Перейти в корень директории api-tests.
- Создать файл .env.
- Заполнить необходимыми параметрами файл .env для API (жирным шрифтом выделены обязательные параметры):
    - **MODULE**=type – тип автотестов (API, WEB, MOBILE).
    - **BASE_URL**=https://… – url окружения, на котором необходимо запустить тесты (например, https://digitalleague.ru/).
    - **TAGS**=@tag – тег сценариев, которые необходимо запустить (например, @test).
    - SEQUENCE_TAGS=@tag – сценарии, помеченные этими тегами, будут запускаться параллельно, независимо от значения параметра PARALLEL_COUNT (если параметр не указывать, то по умолчанию не будет ограничений по параллельному прохождению сценариев). Использование данного параметра может быть полезно, когда, например, 2 сценария изменяют одно и тоже поле в базе данных разными значениями. При параллельном прохождении сценарии могут влиять друг на друга и тесты не будут стабильными. Чтобы явно указать, что эти сценарии должно запускаться последовательно, необходимо оба сценария пометить одним и тем же тегом и указать этот тег в параметре SEQUENCE_TAGS. В случае, если есть необходимость в указании нескольких таких наборов сценариев, теги в параметре указываются через запятую (например, SEQUENCE_TAGS=@tag1,@tag2).
    - PARALLEL_COUNT=1 – количество параллелей запусков сценариев (если параметр не указывать, то по умолчанию значение будет 0 и, как следствие, все сценарии будут запущены последовательно).
    - RETRY_COUNT=1 – количество перезапусков сценариев в случае их падения с тегом @flaky (если параметр не указывать, то по умолчанию значение будет 1).

    В случае если необходим запуск с записью отчета в ReportPortal, следует добавить следующие параметры:

- REPORT_PORTAL_API_KEY=token – токен для взаимодействия с ReportPortal (в случае если токен утерян, можно сформировать новый в настройках профиля ReportPortal).
- REPORT_PORTAL_ENDPOINT=http://… – url ReportPortal.
- REPORT_PORTAL_PROJECT=name – название проекта для записи отчета в ReportPortal (если параметр не указывать, то по умолчанию значением будет название проекта “autotest”).
- REPORT_PORTAL_LAUNCH=name – название запуска в отчете ReportPortal (если параметр не указывать, то по умолчанию значением будет API_TEST_EXAMPLE).
- REPORT_PORTAL_DESCRIPTION=description – описание запуска, который будет отображаться в ReportPortal (если параметр не указывать, то по умолчанию значением будет “Тестовый запуск”).
- REPORT_PORTAL_MODE=name – название директории, в которую будет записан отчет в ReportPortal. Возможны 2 варианта: DEBUG или DEFAULT (если параметр не указывать, то по умолчанию будет выбран вариант DEBUG).

В случае необходимости использования, следует добавить следующие параметры:

- ELASTICSEARCH_URL=url – url elasticsearch (если нужен для тестов).
- ORACLE_PASSWORD=pswd – пароль от OracleDB (если нужен для тестов).
- ORACLE_URL=url – url OracleDB (если нужен для тестов).
- CASSANDRA_PASSWORD=pswd – пароль от CassandraDB (если нужен для тестов).
- CASSANDRA_USERNAME=username – username от CassandraDB (если нужен для тестов).
- CASSANDRA_URL=url – url от CassandraDB (если нужен для тестов).
- KAFKA_BROKER=config – конфигурация для доступа к Kafka (если нужен для тестов).
- REDIS_URL= url – url от Redis (если нужен для тестов).

- Перейти в корень директории web-tests.
- Создать файл .env.

- Заполнить необходимыми параметрами файл .env для WEB (находится в корне директории web-tests):
    - **MODULE**=type – тип автотестов (API, WEB, MOBILE).
    - **BASE_URL**=https://… - url окружения, на котором необходимо запустить тесты (например, https://digitalleague.ru/)
    - **TAGS**=@tag – тег сценариев, которые необходимо запустить (например, @test).
    - RUN_CHROME=true – необходимость запускать только в Chrome (true/false). Значение по умолчанию – true.
    - RUN_MULTI_CHROME=true – флаг, отвечающий за необходимость запустить несколько браузеров для одной сессии. Используется только для сценария “Проверка сессии после выхода”, при запуске других сценариев, параметр можно не передавать, либо передавать со значение false.
    - MAX_INSTANCES-количество параллелей (количество поднятых docker-контейнеров с Chrome). Значение по умолчанию – 1.
    - DRY_RUN=true – вариация запуска тестов. Варианты (true/false), где true – запускать тесты без прогона, а false – обычный запуск тестов.

    В случае если необходим запуск с записью отчета в ReportPortal, следует добавить следующие параметры:
    - REPORT_PORTAL_API_KEY=token – токен для взаимодействия с ReportPortal (в случае если токен утерян, можно сформировать новый в настройках профиля ReportPortal).
    - REPORT_PORTAL_ENDPOIN=http://… – url ReportPortal.
    - REPORT_PORTAL_LAUNCH=name – название запуска в отчете ReportPortal (если параметр не указывать, то по умолчанию значением будет “Тестовый запуск”).
    - REPORT_PORTAL_MODE=name – название директории, в которую будет записан отчет в ReportPortal. Возможны 2 варианта: DEBUG или DEFAULT (если параметр не указывать, то по умолчанию будет выбран вариант DEBUG).
    - REPORT_PORTAL_PROJECT=name – название проекта для записи отчета в ReportPortal (если параметр не указывать, то по умолчанию значением будет название проекта “autotest”).

- REPORT_PORTAL_DESCRIPTION=description – описание запуска, который будет отображаться в ReportPortal (если параметр не указывать, то по умолчанию значением будет “Тестовый запуск”).

В случае необходимости использования, следует добавить следующие параметры:

- ELASTICSEARCH_URL=url – url elasticsearch (если нужен для тестов).
- ORACLE_PASSWORD=pswd – пароль от OracleDB (если нужен для тестов).
- ORACLE_URL=url – url OracleDB (если нужен для тестов).
- CASSANDRA_PASSWORD=pswd – пароль от CassandraDB (если нужен для тестов).
- CASSANDRA_USERNAME=username – username от CassandraDB (если нужен для тестов).
- CASSANDRA_URL=url – url от CassandraDB (если нужен для тестов).
- KAFKA_BROKER=config – конфигурация для доступа к Kafka (если нужен для тестов).
- REDIS_URL= url – url от Redis (если нужен для тестов).

3. Запуск Системы

3.1. Для локального запуска

- В терминале перейти в директорию с API-автотестами (выполнить в терминале команду cd api-tests).
- В терминале выполнить команду: “npm run test”.
- В терминале перейти в директорию с WEB-автотестами (выполнить в терминале команду cd web-tests).
- В терминале выполнить команду: “npm run wdio”.
- В терминале перейти в директорию с Mobile-автотестами (выполнить в терминале команду cd mobile-tests).
- В терминале выполнить команду: “npm run wdio.appium”.

3.2. Для серверного запуска

- В терминале перейти в директорию с API-автотестами (выполнить в терминале команду cd api-tests).
- В терминале выполнить команду: “ docker run --rm -e NODE_TLS_REJECT_UNAUTHORIZED=0 -e TZ=Europe/Moscow -e LANG=ru_RU.UTF-8 -e LANGUAGE=ru_RU.UTF-8 -e LC_ALL=ru_RU.UTF-8 -e PACTUM_REQUEST_TIMEOUT=40000 --name api-tests api-tests npm run test”.
- В терминале перейти в директорию с WEB-автотестами (выполнить в терминале команду cd web-tests).
- В терминале выполнить команду: “ docker-compose run --rm -e NODE_TLS_REJECT_UNAUTHORIZED=0 cucumber”.

4. Контактная информация

В случае возникновения вопросов по настоящей инструкции желательно обратиться к следующим специалистам:

Иваньков Иван
Руководитель практики
+7 (903) 141-49-56
iivankov@fil-it.ru

Зубцов Игорь
Руководитель группы тестирования
+7 (910) 283-79-07
izubtsov@fil-it.ru