



JavaScript QA Engineer

Практический курс по автоматизации тестирования на JavaScript

Длительность курса: 120 академических часов

1 Введение в автоматизацию тестирования на JS

1 Введение в автоматизацию тестирования

Краткое содержание:

что такое тестовый фреймворк;
из каких частей состоит тестовый фреймворк;
модель ISQTB;
что такое runner;
популярные тестовые ранеры.

Домашние задания

1 Знакомство

Цель: Привет! Мы хотим познакомиться с вами, поэтому предлагаем ответить на вопросы ниже.

Составьте короткий рассказ о себе.

- 1) Ваш опыт в автотестировании, ваша роль на текущем проекте
- 2) Какие фреймворки/библиотеки использовали и какие хочется попробовать
- 3) Какую самую интересную штуку/проект/задачу тестировали?

Отправь на проверку преподавателю через кнопку "Отправить ДЗ на проверку"

2 Git на кончиках пальцев

Цели занятия:

назвать команды для работы с локальными изменениями;
назвать команды для работы с ветками;
назвать команды для работы с файлами;
назвать команды для работы с информацией.

Краткое содержание:

основные команды git;
с чем едят gitflow;
популярные графические клиенты. Git hook.

Домашние задания

1 Домашняя работа №2

Цель: Продемонстрируйте базовый уровень владения Git и сервисами github.com

Создать репозиторий на GitHub, сделать ветки master и develop, внести изменения в ветке develop, создать Pull Request на слияние develop и master.

Ссылку на Pull request отправьте в чат к ДЗ.

И пришлите в чат свою учетку github для добавления вас в закрытую группу.

3 Unit тесты и data driven testing

Цели занятия:

написать unit тест для функции;
написать параметризированный тест.

Краткое содержание:

принцип AAA при проектировании тестов;
Unit тесты (признаки unit теста, метрики количества unit тестов на проекте).

Домашние задания

1 Пробуем unit тесты своими руками

Цель: В этой ДЗ знакомимся с синтаксисом обычного теста и параметризованного теста в jest. И пробуем применить принцип AAA на практике.

Сделайте форк репозитория <https://github.com/OTUS-QA-JS-2022-06/jest2022-06> Домашние задания, а точнее папка с тестами, прячется в каталоге specs. Для ревью домашнего задания присылайте в чат ссылку с pull request в fork своего проекта.

В случае если в основном проекте будут изменения, вам нужно будет синхронизировать ваш и основной репозиторий используя git rebase. Об изменениях в коде основного проекта будет уведомления в slack.

Краткое содержание:

функции;
стрелочные функции;
экспорт функций.

Домашние задания

1 Домашняя работа №4

Цель: Знакомимся с интерполяцией и switch, экспортируем и импортируем функции

```
/**  
 * В src/homework4.js напишите функцию "Колобок"  
 * и тесты к ней здесь.  
 * Функция на вход принимает имя персонажа,  
 * например, "дедушка", а в ответ возвращает,  
 * текстовую строку.  
 * Значение текстовой строки - какой был результат  
 * взаимодействия Колобка с данным персонажем.  
 * Например, "дедушка" - "Я от дедушки ушел".  
 * В функции используйте конструкцию switch -  
 * https://learn.javascript.ru/switch  
 */
```

```
/**  
 * В src/homework4.js напишите функцию "Новый  
 * год" и тесты к ней здесь.  
 * Функция на вход принимает имя персонажа. Дед  
 * Мороз или Снегурочка.  
 * Возвращает "Снегурочка! Снегурочка!  
 * Снегурочка!" или "Дед Мороз! Дед Мороз! Дед  
 * Мороз!  
 * В функции используйте интерполяцию.  
 * https://learn.javascript.ru/string  
 *  
 * https://developer.mozilla.org/ru/docs/Web/JavaScript/  
 \* Reference/Template\_literals  
 */
```

1 Препарируем http запросы

Цели занятия:

назвать части http запроса;
отправить типичные запросы через Postman;
перечислить особенности реализации api;
перечислить виды авторизации в api.

Краткое содержание:

что такое API;
виды авторизации в api;
что такое заголовки сообщений;
какого формата бывает тело запроса;
CRUD с использованием POST, PUT, GET, DELETE. 4
типа параметров.

Домашние задания

1 Немного тестов API)

Выберите любой проект с api. Например:

- 1) <https://developers.kaiten.io/docs/api-v1.html>
- 2) Jira
- 3) <https://airportgap.dev-tester.com>

Создайте файл homework5.js и создайте не менее 5 автотестов для api.

2 Препарируем http запросы. Практика

Цели занятия:

попрактиковаться в написании автотестов,
разобрать выполнение домашнего задания

3 Основы JS. Объекты и массивы

Цели занятия:

применить техники языка, которые помогут в изучении тестовых фреймворков.

Краткое содержание:

объекты и массивы как часть ответа api;
как вытащить данные;
как обработать список.

Домашние задания

1 Объекты и массивы

Напишите функцию (и документацию к ней) и набор тестов.

Функция на вход принимает объект, в котором хранятся оценки за ДЗ нашей группы.

```
const scores = {  
  Anna: 10,  
  Olga: 1,  
  Ivan: 5,  
}
```

Функция возвращает в ответ сумму всех баллов.

4 Библиотеки для тестирования API

Цели занятия:

назвать популярные библиотеки для тестирования API; отправить типичные запросы через jest+axios.

Краткое содержание:

обзор библиотек: axios, got, shakram, supertest.

Домашние задания

1 API тесты с использованием контроллеров

Выберите любой проект с api. Например:

- 1) <https://developers.kaiten.io/docs/api-v1.html>
- 2) Jira
- 3) <https://airportgap.dev-tester.com>
- 4) <https://bookstore.toolsqa.com/swagger/>
- 5) <http://try.vikunja.io/>
- 6) <https://kaiten.io/>

Создайте файл homework6.js и создайте не менее 10 автотестов для api с использованием декомпозиции. Используйте контроллеры и конфиги.

Рекомендуемая структура проекта начиная с этого ДЗ:

```
src // исходный код проекта, опциональная папка,
если проект написан на JS и тесты хранятся в
одном репозитории с исходным кодом проекта
specs // папка с тестами
framework // тестовый фреймворк
framework/services // контроллеры, обёртка
запросов к API
framework/config // базовый урл, логин / пароль,
токен авторизации
framework/fixtures // слой с фикстурами,
генераторами данных для тестов
```
```

---



## 5 Шаблоны проектирования и техники тест-дизайна в тестировании API

### Цели занятия:

перечислить шаблоны проектирования, которые могут быть использованы при автоматизации API тестов; рассказать какие API тесты автоматизировать в первую очередь.

### Краткое содержание:

шаблоны проектирования и техники тест-дизайна.

### Домашние задания

#### 1 Миниатюрный тестовый фреймворк

Выберите API из списка ниже, при желании вы можете написать тесты на АПИ с вашей работы.

- 1) <https://developers.kaiten.io/docs/api-v1.html>
- 2) <https://airportgap.dev-tester.com>
- 3) <https://bookstore.toolsqa.com/swagger/>
- 4) <http://try.vikunja.io/>
- 5) <https://kaiten.io/>

Напишите 3 теста с использованием паттерн контроллер и конфигов.

Первый тест - положительный.

Второй тест - набор параметризованных тестов.

Третий тест - проверяет права доступа к точке без api\_key

---

## 6 Валидируем http запросы

### Цели занятия:

применить валидацию json схемы в учебном проекте.

### Краткое содержание:

как документируется API;  
RAML или Swagger вот в чем вопрос;  
что такое json схема;  
популярные библиотеки для валидации.

### Домашние задания

#### 1 Рефлексируем над API

Напишите в чат преподавателем:

- 1) Как вам ДЗ с начала курса?
  - 2) Чувствуете ли в себе силы, на новом/текущем проекте сделать автоматизацию API?
- 

## 7 Стратегия автоматизации в agile проектах и планирование покрытия

### Цели занятия:

узнать способы построения тестовой модели.

### Краткое содержание:

по-платформенно;  
по-фичево;  
на основании пользовательских сценариев;  
на основании дерева требований.

## 1 Репортеры и внешние сервисы

### Цели занятия:

перечислить виды репортеров.

### Краткое содержание:

типы репортеров;  
репортеры.

---

## 2 Контейнеры. Docker

### Цели занятия:

создать docker образ с тестовым фреймворком.

### Краткое содержание:

docker;  
контейнер;  
том;  
образ;  
dockerfile;  
создание docker образ с тестами.

---

3 Катит - не катит.  
Gitlab CI часть  
1

**Цели занятия:**

настроить свой pipeline для запуска тестов.

**Краткое содержание:**

варианты запуска job (Авто/ручная);  
запуск тестов ручной и автоматический;  
как передать имя сервера в тесты;  
отправляем сообщения в Telegram через сторонние сервисы и webhook.

**Домашние задания**

1 Домашняя работа №10

В этом ДЗ у вас есть выбор, gitlab или github.

Если вы выбрали gitlab:

1. Создайте в gitlab репозиторий.
2. Перенесите туда минитестовый фреймворк.
3. Добавьте репортинг jest-html
4. Публикацию отчета на gitlab pages
5. Информирование в телеграмм.
6. Все переменные должны быть скрыты в настройках gitlab.

Если вы выбрали github:

1. Добавьте репортинг jest-html
  2. Публикацию отчета на github pages
  3. Информирование в телеграмм.
  4. Все переменные должны быть скрыты в настройках github.
-

4 Катит - не катит.  
Gitlab CI часть  
2

**Цели занятия:**

настроить свой расширенный pipeline для запуска тестов.

**Краткое содержание:**

подключение gitlab pages;  
публикация артефактов сборки, связанные job;  
пайплайн линтер, codecoverage (или проверка безопасности), тесты.

---

5 Q&A сессия

**Цели занятия:**

Напиши - разбор вопросов студентов по прошедшим модулям

## 1 Обзор инструментов для тестирования UI

### Краткое содержание:

codeceptJS;  
webdriverIO;  
puppeteer;  
playwright;  
testCafe;  
cypress;  
разбор основных отличий в принципах работы.

---

## 2 Искусство стабильных локаторов

### Цели занятия:

находить элементы на страницах, правильно выбирая тип локатора.

### Краткое содержание:

для чего нужны локаторы;  
puppeteer;  
CSS и XPATH локаторы.

---

## 3 Основные сценария использования Playwright

### Цели занятия:

написать первый тест, используя Playwright и jest.

### Домашние задания

#### 1 Немного тестов UI

Цель: Фиксируем как можно быстро и просто писать тесты.

Выбрать сервис, UI которого будем тестировать. Этот сервис вполне может стать основой вашей проектной работы. Пишем, 5 UI функциональных тестов, особое внимание обратить на ожидаемый результат.

---

4 **Шаблоны проектирования и техники тест-дизайна в тестировании UI**

**Краткое содержание:**

page object;  
page elements.

**Домашние задания**

1 Домашняя работа №13

Цель: Возьмите любой сайт, желательно не как в ДЗ 12)  
Сделайте 5 функциональных сценариев используя Page Object

---

5 **Основы JS. Классы**

**Краткое содержание:**

подробнее про современный синтаксис ES6;  
асинхронность в JS;  
NodeJS.

---

6 **Основы JS. Классы в типовых сценариях автотестов**

7 **Behavior Driven Development и acceptance тесты**

**Цели занятия:**

написать UI тест используя codeceptJS и playwright.

**Домашние задания**

1 CodeceptJs

Цель: Напишите 5 тестов с использованием Pageobject используя чистый CodeceptJS

## 1 Выбор темы и организация проектной работы

### Цели занятия:

на основе полученных знаний, выбрать сервис и покрыть его автоматизацией;  
выбрать и обсудить тему проектной работы;  
спланировать работу над проектом;  
ознакомиться с регламентом работы над проектом.

### Краткое содержание:

правила работы над проектом и специфика проведения итоговой защиты;  
требования к результату проекта и итоговой документации.

### Домашние задания

1 Проектная работа

---

## 2 Консультация по проектам и домашним заданиям

### Цели занятия:

получить ответы на вопросы по проекту, ДЗ и по курсу.

### Краткое содержание:

вопросы по улучшению и оптимизации работы над проектом;  
затруднения при выполнении ДЗ;  
вопросы по программе.

---



**3 Защита  
проектных  
работ**

**Цели занятия:**

защитить проект и получить рекомендации экспертов.

**Краткое содержание:**

презентация проектов перед комиссией;  
вопросы и комментарии по проектам.