

# Инженер по тестированию

1

## Введение в профессию. Клиент-серверная архитектура

- Введение в профессию
- Клиент-серверная архитектура
- Хранилища в браузере
- Принцип работы авторизации

2

## Тестирование веб-приложений. Devtools. REST и SOAP

- Что такое стиль REST
- Что такое протокол SOAP
- Формат передачи данных
- Инструмент Devtools

3

## Микросервисная архитектура. Тестируем API с помощью Postman

- Монолитная и Микросервисная архитектура
- Что такое API
- Инструмент Postman
- Практика работы с SOAP UI

4

## Виды и уровни тестирования. Техники тест-дизайна

- Виды требований
- Техники тест-дизайна
- Виды тестирования
- Пирамида тестирования

5

## Виды тестовой документации. Чек-лист и тест-план

- Виды тестовой документации
- Чек-лист
- Тест-план
- Тестовая модель
- Метрики тестирования

6

## Работа с TMS Test IT: тест-кейсы, тест-сюэты и тест-раны. Создаём баг-репорт в Яндекс Трекер

- Работа с TMS на примере Test It
- Создаём баг-репорт в Яндекс Трекере
- Как найти логи и читать дашборды
- Локализация дефекта

7

## SQL: основные операторы, виды джойнов. Базы данных, транзакции, CRUD, ACID, SQL-инъекции

- Работа с реляционной БД на примере PostgreSQL
- Работа с нереляционными БД
- Трассировка запросов в Jaeger
- Роль тестировщика в Agile команде

8

## HTML и CSS. Сниффер Charles. TCP/IP и модель OSI

- Принцип работы веб-приложений
- Сниффер Charles
- TCP/IP и модель OSI
- Авторизация с JWT-токен

### Стажировка на позиции младшего тестировщика

9

## Брокеры сообщений: Apache Kafka. Основы Linux и Git

- Брокеры сообщений
- Базовые Linux команды
- Базовые Git команды
- Подключение к серверу по SSH
- Чтение логов с сервера

10

## Тестовые окружения. CI/CD. Веб-сервер. Моки

- Что происходит после ввода домена в браузере
- Какие бывают тестовые окружения
- Что такое CI/CD
- Моки: зачем их применять тестировщику
- Жизненный цикл разработки ПО

11

## Мобильные приложения. Logcat и ADB. Firebase, пуши, SDK, диплинки

- Виды мобильных приложений
- Работаем с Android Studio
- Firebase и Crashlytics

12

## Введение в автотесты. JavaScript. Cypress. Docker

- Виды автотестов
- Пишем первые автотесты на Cypress
- Работаем с Docker

### Помощь с трудоустройством

13

## Автотесты на API. Python Requests, Pytest

- Базовый синтаксис Python
- Работа с Pytest и Requests
- Пишем первый автотест на API
- Фикстура и параметризация

14

## Автотесты на веб-приложения. Python, Selenium, Allure

- Изучаем, что такое Selenium
- Пишем первый автотест
- Настраиваем отчётность в Allure Report

15

## Подготовка к собеседованиям. Составление портфолио

- Этапы собеседований
- Общие советы по поиску работы
- Популярные вопросы на собеседовании
- Подготовка портфолио

16

## С нуля создаём резюме на HeadHunter

- Составление резюме с нуля под ключ
- Советы по написанию резюме

Ещё 8 месяцев поддерживаем на этапе поиска работы

# Введение в профессию.

# Клиент-серверная архитектура

- Принципы тестирования
- HTTP
- URL
- Cookie
- ChatGPT

Продолжительность спринта 1 неделя

В спринте изучаем 4 темы:

1. Введение в профессию	<ul style="list-style-type: none"><li>Разница между QA, QC и testing</li><li>Принципы тестирования</li><li>Роль тестировщика</li></ul>
2. Клиент-серверная архитектура	<ul style="list-style-type: none"><li>Из чего состоит HTTP-запрос и ответ</li><li>Как работает защищённый протокол TLS</li><li>Из чего состоит URL</li><li>Какие бывают статусы ответа</li></ul>
3. Хранилища в браузере	<ul style="list-style-type: none"><li>Local storage, Session storage, Cookie, Cache</li><li>В чем разница, где их найти</li></ul>
4. Принцип работы авторизации	<ul style="list-style-type: none"><li>Разница между авторизацией, аутентификацией, идентификацией и валидацией</li><li>Как создаётся сессия и где хранится куки session_id</li></ul>
Практика	В домашнем задании разберёшь HTTP URL по частям и напишешь первые промпты для нейросетей

# Тестирование веб-приложений. Devtools. REST и SOAP

- REST
- SOAP
- Devtools
- JSON
- XML

Продолжительность спринта 1 неделя

В спринте изучаем 4 темы:

1. Что такое стиль REST	<ul style="list-style-type: none"><li>• Принципы REST</li><li>• Типы запросов</li><li>• Идемпотентность</li></ul>
2. Что такое протокол SOAP	<ul style="list-style-type: none"><li>• В чем отличие от REST</li><li>• Структура запроса</li><li>• Как работать с WSDL</li></ul>
3. Формат передачи данных	<ul style="list-style-type: none"><li>• Что такое JSON и XML</li><li>• Какие используют типы данных в JSON</li><li>• Как написать свой JSON</li></ul>
4. Инструмент Devtools	<ul style="list-style-type: none"><li>• Как открыть, настроить и применять Devtools на практике</li></ul>
Практика	В домашнем задании напишешь свой первый JSON и с помощью Devtools протестируешь первые задачи

# Микросервисная архитектура. Тестируем API с помощью Postman

- API
- Postman
- Swagger
- SOAP UI
- cURL

Продолжительность спринта 1 неделя

В спринте изучаем 4 темы:

1. Монолитная и микросервисная архитектура	<ul style="list-style-type: none"><li>• Плюсы и минусы у обоих подходов</li></ul>
2. Что такое API	<ul style="list-style-type: none"><li>• Где найти документацию к API</li><li>• Как тестировать API</li><li>• Универсальные проверки при работе с API</li><li>• Как работать с документацией в Swagger</li></ul>
3. Инструмент Postman	<ul style="list-style-type: none"><li>• Создаём коллекцию запросов и переменные</li><li>• Пишем автотесты по сниппетам и пре-реквесты</li><li>• Запускаем раннеры коллекции</li></ul>
4. Практика работы с SOAP UI	<ul style="list-style-type: none"><li>• Пишем запросы на создание и получение информации</li><li>• Работаем с WSDL</li></ul>
Практика	Протестируешь API тренажёра «Битва покемонов». Это микросервисная архитектура, больше 20 REST и SOAP-методов. Поработаешь с инструментами Postman и SOAP UI

# Виды и уровни тестирования.

## Техники тест-дизайна

- Классы эквивалентности
- Граничные значения
- Pairwise
- Интеграционное тестирование

Продолжительность спринта 1 неделя

В спринте изучаем 4 темы:

1. Виды требований	<ul style="list-style-type: none"><li>Функциональные и нефункциональные требования</li><li>Как проводить тестирование требований</li></ul>
2. Техники тест-дизайна	<ul style="list-style-type: none"><li>Разбиение на классы эквивалентности</li><li>Анализ граничных значений</li><li>Pairwise</li><li>Таблица принятия решений</li><li>Схема состояний и переходов</li><li>Предугадывание ошибок</li></ul>
3. Виды тестирования	<ul style="list-style-type: none"><li>Функциональные виды тестирования</li><li>Нефункциональные виды тестирования</li><li>Белый / серый / черный ящики</li></ul>
4. Пирамида тестирования	<ul style="list-style-type: none"><li>Сквозное тестирование</li><li>Интеграционное тестирование</li><li>Юнит-тестирование</li></ul>

Практика	Применишь технику тест-дизайна «Классы эквивалентности» и «Граничные значения» для тестирования подписки Премиум в «Битве покемонов», а также протестируешь документацию к функционалу применения промокода
----------	---

# Виды тестовой документации.

## Чек-лист и тест-план

- Чек-лист
- Тест-план
- Figma
- Тестовая модель

Продолжительность спринта 1 неделя

В спринте изучаем 3 темы:

1. Виды тестовой документации	<ul style="list-style-type: none"><li>Чек-лист</li><li>Тест-план</li></ul>
2. Чек-лист	<ul style="list-style-type: none"><li>Инструменты для составления чек-листов</li><li>Особенности браузерных движков</li><li>Особенности чек-листа для сайта, приложения и API</li><li>Планирование времени на тестирование</li></ul>
3. Тест-план	<ul style="list-style-type: none"><li>Этапы тестирования</li><li>Тестовая модель</li><li>Метрики тестирования</li><li>Критерии начала и завершения тестирования</li></ul>
Практика	Протестируешь UI формы для оплаты, используя Figma. Составишь чек-листы для тестирования авторизации и API-метода «поймать покемона в покебол»

# Работа с TMS Test IT: тест-кейсы, тест-сююты и тест-раны. Создаём баг-репорт в Яндекс Трекере

- TMS
- Баг-репорт
- Kibana
- Grafana
- Sentry

Продолжительность спринта 1 неделя

В спринте изучаем 4 темы:

1. Работа с TMS на примере Test IT	<ul style="list-style-type: none"><li>• Как составить тест-кейс</li><li>• Как создать test suite и test run</li><li>• Как выглядит отчёт по тестированию</li><li>• Интеграция GitLab и Test IT: запускаем автотесты</li></ul>
2. Создаём баг-репорт в Яндекс Трекере	<ul style="list-style-type: none"><li>• Как правильно назвать баг-репорт</li><li>• Разница между Priority и Severity</li></ul>
3. Как найти логи и читать дашборды	<ul style="list-style-type: none"><li>• Анализ логов в Kibana</li><li>• Изучение подробностей ошибки в Sentry</li><li>• Работаем с дашбордами в Grafana</li></ul>
4. Локализация дефекта	<ul style="list-style-type: none"><li>• Инструменты для локализации</li><li>• Признаки бага со стороны клиента и API</li><li>• Очерёдность поиска причины</li></ul>

Практика	Будешь искать искать баги, создавать баг-репорты на UI и API, научишься искать логи в Kibana и Sentry, читать дашборды в Grafana. Создашь первые тест-кейсы и тест-сююты в TMS Test IT
----------	--

# SQL: основные операторы, виды джойнов. Базы данных, транзакции, CRUD, ACID, SQL-инъекции

- SQL
- PostgreSQL
- NoSQL
- Redis
- MongoDB
- Jaeger
- DBeaver

Продолжительность спринта 1 неделя

В спринте изучаем 4 темы:

1. Работа с реляционной БД на примере PostgreSQL	<ul style="list-style-type: none"><li>Первичный и внешний ключи</li><li>Виды связей в БД</li><li>Основные SQL-операторы</li><li>Виды джойнов</li><li>Агрегатные функции</li><li>Вложенные запросы</li><li>Типы данных в БД</li><li>CRUD, ACID, SQL-инъекции</li></ul>
2. Работа с нереляционными БД	<ul style="list-style-type: none"><li>Синтаксис NoSQL-запросов в MongoDB</li><li>Работа с БД Redis на примере кэша сервера</li></ul>
3. Трассировка запросов в Jaeger	
4. Роль тестировщика в Agile команде	

Практика

Про интеграционное тестирование всё чаще спрашивают на собеседованиях. На нашем тренажёре применишь лучший инструмент для тестирования интеграций — Jaeger. Протестируешь функционал оплаты и логику работы ачивок в «Битве покемонов» со стороны базы данных. Поработаешь с СУБД PostgreSQL, MongoDB и Redis.

# HTML и CSS. Сниффер Charles. TCP/IP и модель OSI

- HTML
- Charles
- JWT
- TCP/IP

Продолжительность спринта 1 неделя

В спринте изучаем 4 темы:

1. Принцип работы веб-приложений	<ul style="list-style-type: none"><li>• HTML и CSS</li><li>• Как браузер отрисовывает сайты</li><li>• Технологии SPA и AJAX</li></ul>
2. Сниффер Charles	<ul style="list-style-type: none"><li>• Применяем Breakpoints и Map Local</li><li>• Аналоги Charles</li></ul>
3. TCP/IP и модель OSI	<ul style="list-style-type: none"><li>• Уровни в TCP/IP</li><li>• Разница TCP и UDP</li></ul>
4. Авторизация с JWT-токеном	<ul style="list-style-type: none"><li>• Что такое сессии</li><li>• Принцип работы JWT</li><li>• Разница между Stateless и Stateful</li><li>• Что такое Access - и Refresh-токены</li><li>• Зачем нужен Bearer</li><li>• Как при авторизации применяют чэширование</li><li>• Как декодировать Access-токен</li><li>• Что такое OAuth и OpenID</li></ul>

Практика	<p>С помощью HTML и CSS создашь свой сайт. Применешь Charles на практике: протестируешь функцию «легендарный покемон» и обработку ошибок в «Битве покемонов». Поработаешь с JWT авторизацией на нашем тренажёре</p>
----------	---

# Брокеры сообщений: Apache Kafka. Основы Linux и Git

Kafka

Linux

Git

Продолжительность спринта 1 неделя

В спринте изучаем 3 темы:

1. Брокеры сообщений	<ul style="list-style-type: none"><li>Асинхронное общение микросервисов</li><li>Apache Kafka: топики, оффсеты и партиции</li><li>Работаем с Kafka UI и Offset Explorer</li><li>Особенности тестирования проектов с брокерами сообщений</li></ul>
2. Базовые Linux команды	<ul style="list-style-type: none"><li>Подключаемся к серверу с помощью SSH</li><li>Читаем логи сервера в реальном времени</li><li>Уровни логирования</li></ul>
3. Базовые Git команды	<ul style="list-style-type: none"><li>Пример Git flow</li><li>Работаем с Merge Request</li></ul>

Практика	<p>В любой крупной компании требуют опыт работы с брокерами сообщений. Вы поработаете на курсе с брокером Apache Kafka и протестируете фичу «Скачивание истории битв» с помощью Kafka UI и Offset Explorer. С помощью SSH подключитесь к удалённому серверу и прочитаете логи в режиме реального времени. Создадите первый репозиторий на своём GitHub</p>
----------	--

# Тестовые окружения. CI/CD. Веб-сервер. Моки

- Веб-сервер
- Nginx
- GitLab
- CI/CD

Продолжительность спринта 1 неделя

В спринте изучаем 5 тем:

1. Что происходит после ввода домена в браузере	<ul style="list-style-type: none"><li>• Что такое ip-адрес</li><li>• Зачем нужен веб-сервер</li><li>• Что такое S3 и CDN</li></ul>
2. Какие бывают тестовые окружения	<ul style="list-style-type: none"><li>• Разница между Prod / Pre-prod / Stage / Dev средами</li><li>• Тестовая среда у API: настраиваем Postman Environment</li></ul>
3. Что такое CI/CD	<ul style="list-style-type: none"><li>• Пример работы GitLab CI/CD</li><li>• Процесс доставки релиза</li><li>• Состав pipeline</li></ul>
4. Моки: зачем их применять тестировщику	<ul style="list-style-type: none"><li>• Пример работы Postman Mock Server</li></ul>
5. Жизненный цикл разработки ПО	<ul style="list-style-type: none"><li>• Особенности Shift-Left подхода в тестировании</li><li>• Жизненный цикл дефекта (бага)</li><li>• Таск-флоу на примере Яндекс Трекера</li><li>• Процесс релиза новой версии продукта</li></ul>

Практика	Запустишь сборку пайплайна в тестовое или продовое окружение в GitLab CI/CD. Будешь применять переменные Postman Environment. Применишь Postman Mock Servers и создашь свою API-заглушку
----------	--

# Мобильные приложения. Logcat и ADB. Firebase, пуши, SDK, диплинки

- Logcat
- Firebase
- SDK
- Диплинки

Продолжительность спринта 1 неделя

В спринте изучаем 3 темы:

1. Виды мобильных приложений	<ul style="list-style-type: none"><li>Движки браузеров на iOS</li><li>Devtools для мобильной версии сайта</li><li>Особенности тестирования мобильных приложений</li><li>Тестируем push-сообщения</li><li>Особенности дебажной и релизной сборки приложений</li></ul>
2. Работаем с Android Studio	<ul style="list-style-type: none"><li>Разница между симулятором и эмулятором</li><li>Смотрим логи в ADB и Logcat</li></ul>
3. Firebase и Crashlytics	<ul style="list-style-type: none"><li>Изучаем логи ANR и Crash</li><li>Анализируем дашборды и статистику приложения</li></ul>

Практика	Запустишь и протестируешь мобильное приложение «Битва покемонов» на Android устройстве или Android Studio. Научишься создавать push-сообщения. Применишь в работе дебаг-меню. Изучишь дашборды и аналитику в Google Firebase. Протестируешь работу диплинков в мобильном приложении «Битва покемонов»
----------	---

# Введение в автотесты. JavaScript. Cypress. Docker

- Cypress
- Docker
- JavaScript
- Автотесты

Продолжительность спринта 1 неделя

В спринте изучаем 3 темы:

1. Виды автотестов	<ul style="list-style-type: none"><li>• Когда имеет смысл автоматизировать тест-кейсы</li><li>• Самые популярные инструменты в автотестах</li><li>• Какие сценарии стоит автоматизировать в первую очередь</li></ul>
2. Пишем первые автотесты на Cypress	<ul style="list-style-type: none"><li>• Особенности фреймворка Cypress</li><li>• Базовый синтаксис JavaScript</li><li>• Пишем первые автотесты на Cypress</li><li>• Усложняем: смена браузера, работа с ожиданиями</li><li>• Запуск автотестов через терминал</li><li>• Запуск автотестов через CI/CD</li><li>• Применяем подход Page Object</li><li>• Используем хуки beforeEach и afterEach</li><li>• Особенности TDD и BDD</li></ul>
3. Работаем с Docker	<ul style="list-style-type: none"><li>• Базовая терминология в Docker</li><li>• Виртуализация и контейнеризация</li><li>• Запускаем локально проект через Docker</li><li>• Читаем логи контейнера: бэкенда и веб-сервера</li></ul>
Практика	Напишите свой первый автотест на UI и автоматизируете покупку аватара в «Битве покемонов» на JavaScript + Cypress

# Автотесты на API. Python, Requests, Pytest

- Requests
- Pytest
- GraphQL
- gRPC
- WebSocket

Продолжительность спринта 1 неделя

В спринте изучаем 2 темы:

1. Пишем первый автотест на API	<ul style="list-style-type: none"><li>Базовый синтаксис Python</li><li>Настройка IDE</li><li>Работа с Pytest и Requests</li><li>Дебаггинг проблем</li><li>Фикстура</li><li>Параметризация</li></ul>
2. Изучаем альтернативы HTTP + REST	<ul style="list-style-type: none"><li>GraphQL и инструмент Apollo</li><li>gRPC и инструмент Kreya</li><li>WebSocket и инструмент Insomnia</li></ul>
Практика	Напишешь свой первый автотест на API с помощью Python и Requests. Познакомишься с синтаксисом и выполнишь запросы с помощью технологий и подходов GraphQL, gRPC и WebSocket

# Автотесты на веб-приложения. Python, Selenium, Allure

Selenium

Allure Report

Python

Продолжительность спринта 1 неделя

В спринте изучаем 1 тему:

1. Что такое Selenium: создаём первый автотест	<ul style="list-style-type: none"><li>• Подготовка проекта</li><li>• Настраиваем отчётность в Allure Report</li><li>• Отладка автотеста</li><li>• Создание conftest с фикстурой</li><li>• Негативные тесты с параметризацией</li><li>• Режим Headless</li><li>• Работа с flaky-тестами</li></ul>
--	--

Практика	Настроишь проект и напишешь свой первый автотест на клиентскую часть (UI), используя язык Python, инструмент Selenium и инструмент для чтения отчётов Allure Report
----------	---

# Подготовка к собеседованиям.

## Составление портфолио

- Портфолио
- Jira
- Confluence

Продолжительность спринта 1 неделя

В спринте изучаем 4 темы:

1. Этапы собеседований	<ul style="list-style-type: none"><li>Особенности скрининга, техсобеса и финального этапа</li><li>Формы трудоустройства</li></ul>
2. Общие советы по поиску работы	<ul style="list-style-type: none"><li>Площадки для размещения резюме</li><li>Примеры сопроводительного письма</li><li>Как составить рассказ о себе</li><li>Особенности видеособеседований</li><li>Стоит ли выполнять тестовые задания</li><li>Как торговаться при получении оффера</li><li>Советы первого рабочего дня</li></ul>
3. Популярные вопросы на собеседовании	<ul style="list-style-type: none"><li>Какие вопросы задаст HR и как на них отвечать</li><li>Какие вопросы стоит задать тебе как кандидату</li></ul>
4. Подготовка портфолио	<ul style="list-style-type: none"><li>Оформление проектов на GitHub</li><li>Создание телеграм-бота для тестировщика</li><li>Работа со стеклом Atlassian: Jira и Confluence</li></ul>

Практика	Создашь портфолио на GitHub со всеми учебными проектами, изученными инструментами, сертификатом о прохождении стажировки, дипломом о завершении обучения. Лучшим ученикам на каждом потоке выдаём именные рекомендательные письма
----------	---

# С нуля создаём резюме на HeadHunter

Резюме

HeadHunter

Продолжительность спринта 1 неделя

В спринте изучаем 2 темы:

1. Составление резюме с нуля под ключ	<ul style="list-style-type: none"><li>Регистрируемся на HeadHunter и создаём резюме</li><li>Заполняем основные разделы: «О себе», «Контакты», «Фото», «Город»</li><li>Добавляем описание опыта после обучения и стажировки в «Джунах»</li><li>Добавляем портфолио, документы и сертификаты об обучении</li></ul>
2. Советы по написанию резюме	<ul style="list-style-type: none"><li>Воронка получения оффера</li><li>Как HR видит твоё резюме</li><li>Как помогает в поиске работы личный бренд</li><li>Как при составлении резюме могут помочь нейросети</li></ul>
Практика	Создашь резюме на HeadHunter, в котором укажешь обучение и стажировку в Джунах. Добавишь в описание портфолио, сертификаты и документы о завершении обучения