**ДЗ №3 (к 6 занятию от 26.02.2019)**

1. Опишите своими словами, как вы понимаете фронтенд и бэкенд. Приведите по 1 примеру вида тестирования, применимого к фронтенду и бэкенду, соответственно.

Фронтенд (Client) - доступный пользователю какой-либо интернет-ресурс, web-приложение в виде браузера, Skype, YouTube, социальных сетей и т.д., с помощью которого происходит взаимодействие пользователя с сервером.  
Бэкенд (Server) - это сервер, который отвечает на запросы от фронтенд (Client).  
Т.е. фронтенд и бэкенд обмениваются запросами и ответами.  
Запросы формируются и уходят с уровня фронтенд.  
Бэкенд принимает запрос, направляет запрос в базу данных, получает информацию, структурирует и выдает результат.  
Фронтенд построен в основном на html, css, javascript.   
Бэкенд может состоять из нескольких блоков и написан на чем угодно.  
Все это клиент-серверная архитектура.  
Для фронтенд:  
- Unit тестирование (автотест). Тестирование отдельных строк кода, которые выполняют отдельную логику. Например, ввод в поисковое поле алфавитных, цифровых, нечитаемых символов, специальных знаков, загрузку картинок, файлов, ввод большого количества символов и т.д.  
- Нагрузочное тестирование. При помощи инструментов для загрузочного тестирования имитируется одновременное увеличение количества запросов от 10, …, 100, …, 500, …, 1000, … пользователей и измеряется скорость загрузки страницы.  
Для бэкенд:  
- Тестирование безопасности. Например, злоумышленник видоизменяет запрос в строке поиска, бэкенд неправильно интерпретирует этот запрос и злоумышленник получает доступ в базу данных.

1. Что такое кросс-браузерное тестирование?  
     
   Разные браузеры отображают одну и ту же страницу не всегда одинаково.  
   При создании какого-либо сайта необходимо убедиться, что он будет доступен широкому кругу пользователей, использующих разные браузеры. Для этого проводят кроссбраузерное тестирование.  
   В интернете есть специальные инструменты для подобного тестирования (сайты), где можно выбрать необходимый для тестирования браузер, в том числе его версию и расширение. После выбора браузера при помощи виртуальной машины открывается запрашиваемый браузер. В адресную строку вводим URL разрабатываемого сайта и проводим тестирование.
2. Предположим вы нашли баг на веб-странице. Какие данные помимо шагов воспроизведения, вы можете приложить к баг-репорту, чтобы разработчику было проще локализовать проблему?  
     
   Скриншот страницы и диагностическая информация - скриншот панели разработчика (F12, вкладки Elements, Console, Network)
3. Опишите своими словами, как вы понимаете GET и POST запросы. В чем их основное различие?  
     
   GET запрос - это когда мы обращаемся к серверу, чтобы получить какие-либо данные.  
   Например, ввод запрашиваемой информации в поисковую строку.  
   POST запрос - когда мы редактируем/отправляем данные на сервер.  
   Например, заполнили регистрационную форму на сайте и отправили на сервер, данная информация будет сохранена в базе данных. Повторно заходим на сайт с этими данными и это тоже POST запрос, т.к. данные отправляются и сверяются.   
   Основное различие - по техническим особенностям GET запрос содержит в себе, в своём адресе всё, что мы хотим получить, а POST запрос содержит тело - адрес, куда мы отправляем запрос и данные, которые отправляем. Например, данные при регистрации - логин, пароль, email и много другой информации. Эти данные можно редактировать последующим POST запросом.  
   GET запрос не меняет состояние системы, а POST запрос меняет состояние системы, появляется что-то новое.
4. Опишите своими словами, как вы понимаете API-тестирование.  
     
   Взаимодействие между клиентом и сервером без браузера, формирование запроса не на сайте, а с помощью специального инструмента. Популярный инструмент Postman.  
   API-тестирование происходит проще и быстрее, потому что не нужно ждать загрузки страницы, нажимать кнопки, заполнять поля и т.д.  
   Например, нужно провести регистрацию пользователя по какому-то URL адресу, но неважно, как это будет выглядеть на сайте, а важен сам результат. Т.е. отправить POST запрос на сервер без браузера.  
     
   API тестирование не для новичков, а для развития уже имеющегося опыта в тестировании.