|  |  |  |
| --- | --- | --- |
|  | Конспекты **ТРЕТЬЕГО** спринта |  |
|  | оглавление |  |
|  |  |  |
|  | Тестирование Мобильной веб-версии | 1 |
|  | **Адаптивность вёрстки** | 1 |
|  | На реальном устройстве | 3 |
|  | Действия на реальном устройстве | 3 |
|  | Виды мобильных приложений | 4 |
|  | Нативные, гибридные, веб |  |
|  | *)* | 6 |
|  | Операционные системы и версии | 8 |
|  | **Android** | 9 |
|  | iOS | 9 |
|  | Параметры мобильных устройств | 9 |
|  | Производитель и модель | 10 |
|  | Разрешение и размер экрана | 10 |
|  | Матрица устройств | 11 |
|  | Эмуляторы | 12 |
|  |  |  |
|  | Создать первое виртуальное устройство | 13 |
|  | Тестирование в **Android Studio** | 17 |
|  | Расширенные настройки | 18 |
|  | Снятие логов в эмуляторе | 20 |
|  | Уровни логирования | 21 |
|  | Как искать логи | 21 |
|  |  |  |
|  | Тестирование на реальном устройстве | 22 |
|  | аппаратные особенности**:** |  |
|  | Прерывания | 22 |
|  | Датчики | 23 |
|  | Параметры связи | 23 |
|  | Геолокация | 24 |
|  | Производительность | 24 |
|  |  |  |
|  | **Установка и снятие логов** | 25 |
|  | *Установка ПРИЛОЖЕНИЯ через комп (перетаск.файла)* | 25 |
|  | Тестирование обновления приложения | 28 |
|  |  |  |
|  | **Что такое API** | 30 |
| GUI | Графический **интерфейс** (Graphical User Interface) |  |
| CLI | **интерфейс** командной строки (Command Line Interface) |  |
| API | программный **интерфейс** приложения (Application Programming Interface*) интерфейс для взаимодействия приложений между собой*. |  |
| REST | REpresentational State Transfer *способ организации API* |  |
|  | работает с URL-запросами, использует HTTP-методы: GET, POST, PUT, PATCH, DELETE и т.п. |  |
| JSON | ***JavaScript Object Notation*** *Специальный формат для описания структуры данных* | 31 |
|  |  |  |
|  | **Тест-дизайн для API** | 31 |
|  | Краткий разбор формата JSON | 32 |
| массив |  | 34 |
|  | Документация | 34 |
| **Swagger** | *инструмент, который автоматически генерируют документацию.* | 35 |
|  | Инструменты для тестирования API | 37 |
|  | Как установить **Postman** | 37 |
|  | Создание данных | 37 |
|  | Получение данных | 38 |
|  | Удаление данных |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

AndroidStudio – Tools - AVD Manager

*Если уже созданы новые девайсы их можно запустить из своего списка (приступить к эмуляции)*

"Create Virtual Device" – *или добавить новые*

"Select hardware". - New Hardware Profile ( диагональ 5.2 \_ разрешение\_1080х1920\_Назови Honor-practicum1) - Finish. (галочки?) RAM ???

"Select hardware"- выбрать созданное - Next.

"System Image" - выбрать версия 9.0 Pie - Next. *Если ОС не загружена на твой компьютер, кликни по синей надписи "Download".стр15*

"Android Virtual Device" - подтвердить или изменить параметры - Finish.

*Приложение вернёт тебя к панели виртуальных устройств и покажет готовый вариант. Чтобы приступить к эмуляции, нажми кнопку Запуска*

Honor 8 на ОС Android 9.0 Pie с разрешением экрана 1080х1920 и диагональю 5.2"

Galaxy A3 на ОС Android 9.0 Pie с разрешением экрана 720х1280 и диагональю 4.7".

Galaxy J2 на ОС Android 9.0 Pie с разрешением экрана 540х960 и диагональю 5".

Назови устройства Practicum2 и Practicum3 соответственно.

1. После выключения эмулятора и Андроид-Студио логи сохраняются?
2. Если эмуляторы закрыть (не выключить, устройство, а закрыть эмулятор)

В «списке установленных» остается только последний и соответственно его логи?

*«В прошлых уроках тебе удалось протестировать несколько эмуляторов — после этого в Android Studio остались логи. Чтобы их посмотреть, зайди на главную страницу и кликни по консоли Logсat:»*

***UI* -** пользовательский интерфейс *для взаимодействия пользователя с компьютером*

***API****- интерфейс для взаимодействия приложений между собой.*

*В случае веб-приложений API**— это* ***фиксированный набор******URL-адресов****, по которым можно передавать данные в* ***заданном формате****.*

*Например:* *фронтенд шлёт* ***GET****-запрос к* ***API*** *бэкенда, чтобы найти пользователя по email.*

*Бэкенд его обрабатывает: ищет пользователя в базе данных и отдаёт их фронтенду. У сервера можно* ***не только запросить*** *информацию о пользователе,*

***но и создать*** *его в базе данных; а также* ***обновить*** *или* ***удалить*** *существующего.*

***REST— это способ организации API****. Он* ***задаёт правила*** *для удобной разработки и поддержки API, работает с* ***URL-запросами*** *и использует* ***HTTP****-****методы:***

|  |  |
| --- | --- |
| ***GET*** | *используется для получения информации от сервера по заданному URI (URI в HTTP). Запросы клиентов, использующие метод GET должны* ***получать только данные*** *и* ***не должны никак влия****ть на эти данные.* |
| ***POST*** | *используется для отправки данных на сервер, например, из HTML форм, которые заполняет посетитель сайта.* |
| ***PUT*** | *используется для загрузки содержимого запроса на указанный в этом же запросе URI.* |
| ***PATCH*** | *используется для частичного обновления ресурса OSLC. PATCH не удаляет никакие свойства локального ресурса, не включенные в требование* |
| ***DELETE.*** | *удаляет указанный в URI ресурс.* |

*- - - - - - -*

*Расшифровка аббревиатур*

*URI - Uniform Resource Identifier (унифицированный* ***идентификатор*** *ресурса – имя и адрес)*

*URL - Uniform Resource Locator (унифицированный определитель* ***местонахождени****я ресурса)*

*URN - Unifrorm Resource Name (унифицированное* ***имя*** *ресурса)*

*Есть несколько вариантов определения слова "интернет". Один из них — это система компьютеров, соединенных через компьютерную сеть. Такая версия интернета появилась в 1969 году с созданием ARPANET. Почта, файлы и чат работали в этой сети еще до создания HTTP, HTML и веб-браузера.*

***В 1992 году Тим Бернерс-Ли*** *создал три штуки, благодаря которым родилось то, что мы считаем интернетом: протокол* ***HTTP, HTML и URL****. Его целью было воплотить понятие* ***гипертекста*** *в реальности. Гипертекст, в двух словах — это* ***возможность создавать документы****, которые* ***ссылаются друг на друга****. В те годы идея гипертекста считалась панацеей из научной фантастики, заодно с гипермедиа, и любыми другими словами с приставкой «гипер».*

*Ключевым требованием гипертекста была возможность ссылаться из одного документа на другой. В то время для хранения документов использовалась куча форматов, а доступ осуществлялся по протоколу вроде Gopher или FTP. Тиму нужен был надежный способ ссылаться на файл, так, чтобы в ссылке был закодирован протокол****, хост*** *в интернете и местонахождение на этом хосте. Этим способом стал URL, впервые официально задокументированный в RFC в 1994 году.*

*В начальной презентации World-Wide Web в марте 1992 Тим Бернерс-Ли описал его как «универсальный идентификатор документов» (Universal Document Identifier или UDI). Множество других форматов также рассматривались в качестве такого идентификатора:*

*- - - - - - - - - - - - - - - - -*

***JSON - формат для описания***  *структуры данных, которые передаются от одного приложения другому на основе REST.*

*API — это тоже интерфейс, но интерфейс необычный. Работа с API* ***строится через запросы****. Где ограничения, там и тест-дизайн…*

*Документация позволяет изучить работу API компонента или приложения. Она описывает: Действия , данные, Ограничения, Структуру.*

*Хранить документацию можно в текстовом виде. Так делают для проектов, где API и документация к нему — небольшие.*

*Если система сложная, документация разрастается, её становится труднее поддерживать и искать в ней информацию. Чтобы оптимизировать этот процесс, применяют инструменты, которые автоматически генерируют документацию. Например,* ***Swagger.***

*Как создают документацию? Разработчик готовит специальный файл с описанием логики работы API, а инструмент автоматически переводит её в понятный вид.*

***В документации*** *есть* ***список кодов ответов*** *с описанием, а также пример данных, которые возвращаются на запрос в теле сообщения:*

*Это описание требований API. Задача тестировщика — убедиться, что реализация чего???? выполнена* ***согласно требованиям****. Чтобы это сделать, нужно провести тест-анализ и тест-дизайн на их основе, а затем протестировать.* И как это тестировать?

***Мы хотим протестировать API при помощи Postman***

*В специальных инструментах, они позволяют тестировщику маскироваться под приложение, отправляющее запрос.*

***Postman –****– инструмент, который помогает формировать запросы и анализировать ответы со стороны сервера. С него можно отправить HTTP-запросы, чтобы протестировать API*

*--------- - -- - - - - - - - - - - -*

"В компьютерной сети **сетевой узел, интернет-узел, хост или интернет-узел** - это компьютер, подключенный к Интернету - или более обобщенно-к любому типу сети передачи данных. Сетевой хост может содержать **информационные ресурсы**, а также прикладное **программное обеспечение** для предоставления сетевых услуг. "-Википедия

**Локальный хост**-это специальное **имя**, данное локальной машине или тому, над чем вы работаете, обычно его IP-адрес равен 127.0.0.1. Однако вы можете определить его как что угодно.

На каждом хосте работает несколько сетевых служб (например, веб-сервер Apache/IIS( Http), почтовые клиенты, клиенты FTP и т. д. Каждая служба имеет определенный **порт**, связанный с ней. Вы можете думать об этом так:

В каждом доме есть один почтовый ящик и несколько человек. ***Почтовый ящик - это хост***. Ваш собственный домашний почтовый ящик-это localhost. У каждого человека в доме есть своя комната. Все письма для этого человека отправляются в его комнату, поэтому ***номер комнаты*** является ***портом***.

**Локальный хост**-это специальное имя, данное локальной машине или тому, над чем вы работаете, обычно его IP-адрес равен 127.0.0.1. Однако вы можете определить его как что угодно.