Лекция 8

Agenda

- Менеджеры зависимостей
  - Cocoa pods
  - Carthage
  - SPM (swift package manager)
- Что такое клиент-серверное приложение
- JSON
- Запрос на данные: Rest клиенты

Менеджеры зависимостей

- Cocoa pods - самый популярный, создает workspace, при использовании желательно знать ruby.
- Carthage - второе место по популярности, не меняет проект, написан на swift, сборка проекта будет происходить быстрее.
- SPM (swift package manager) - нативный, встроенный в xcode менеджер зависимостей с большим будущим, если верить apple.

Подробнее про каждый менеджер зависимостей для Xcode проектов:
https://medium.com/xcblog/swift-dependency-management-for-ios-3bcfc4771ec0
Что такое менеджеры зависимостей и какие они бывают в mac os:
https://habr.com/ru/company/redmadrobot/blog/412945/
Краткое сравнение cocoapods и carthage:
https://arm1.ru/blog/carthage-vs-cocoapods

Cocoa pods

Шпаргалка по основным командам (после установки необходимых библиотек и языков на ваш mac):

- Установить cocoapods на ваш мак - **sudo gem install cocoapods**

*Для выполнения следующих команд через терминал необходимо перейти в репозиторий проекта. Это делается командой **cd** (change directory), после пробела указываем путь к файлу с проектом. Чтобы не писать вручную путь к файлу его можно перетащить мышкой на терминал.*

- Добавить podfile в папку с проектом - **pod init**

Далее в вашем Podfile(открываем с помощью Xcode) необходимо описать зависимости, которые вы хотите использовать и указать target (название проекта). Пример в коде лекции.

- Установить поды - **pod install**
- Удалить поды - удаляем строчку из Podfile и снова **pod install**
- Обновить поды (каждый до последней доступной версии, если версия не указана в Podfile) - **pod update**

Видео туториалы по ссылкам:

Быстрый туториал, как устанавливать cocoapods:
https://www.youtube.com/watch?v=B3Y3Evftq70
Чуть более подробный туториал по cocoapods:
https://www.youtube.com/watch?v=xnxHqc9qN_s
Самые популярные cocoapods:
https://www.advancedswift.com/ranked-swift-cocoapods/

Carthage

Шпаргалка по основным командам (после установки необходимых библиотек и языков на ваш mac):

- Установить carthage на ваш мак - **brew install carthage**

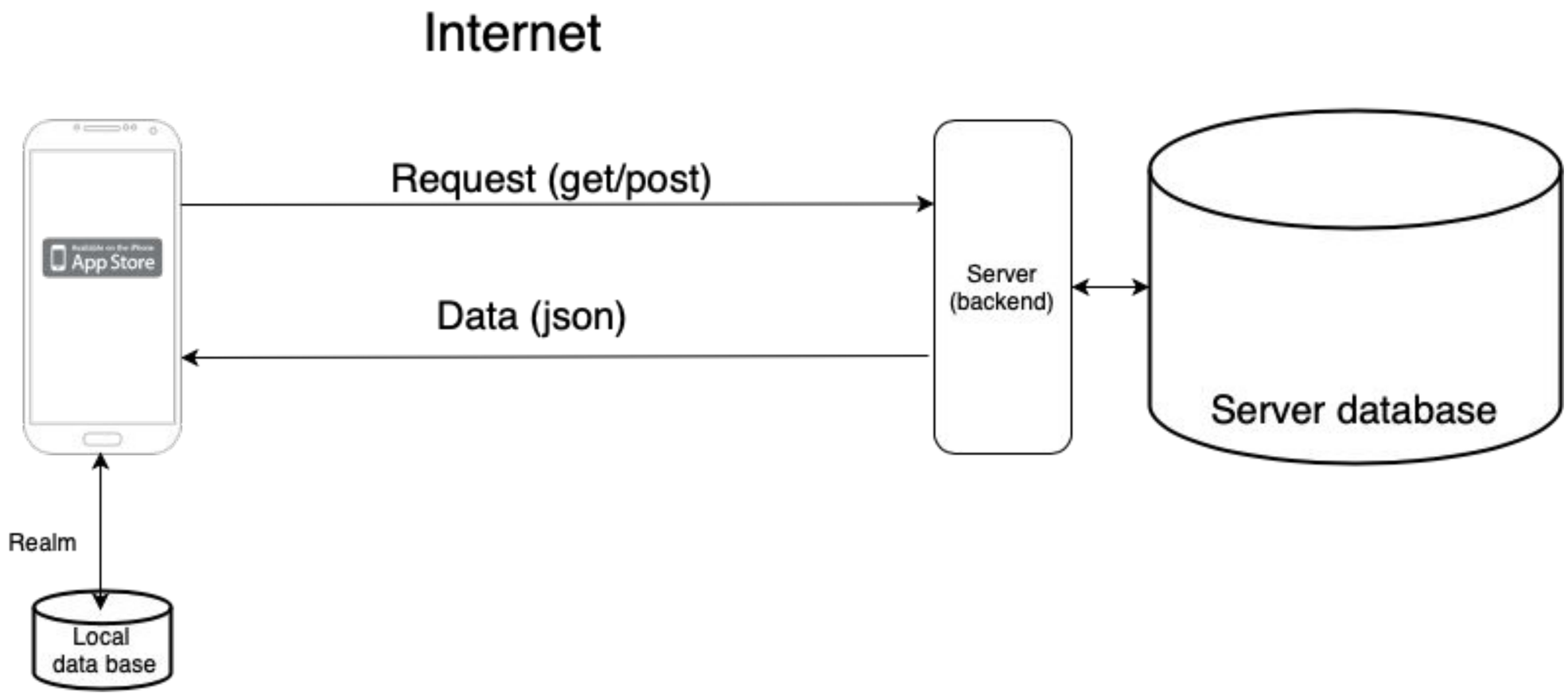
Для выполнения следующих команд через терминал необходимо перейти в репозиторий проекта(аналогично как при установке cocoapods). Далее создаем Cartfile (это можно сделать с помощью команды создания файла через терминал: touch Cartfile) - текстовый файл, в котором мы указываем ссылки на библиотеки, которые мы хотим использовать.

- Добавить необходимые файлы для Carthage и добавить библиотеки(для ios) - ***carthage update --use-xcframeworks --platform iOS***

Далее следуем инструкции описанной на github по настройке Carthage для ios: https://github.com/Carthage/Carthage

Видео про установку Carthage:
https://www.youtube.com/watch?v=DzCogM77LSo
Старый туториал по Carthage на русском и оригинальная статья на английском:
https://habr.com/ru/post/270805/
https://www.raywenderlich.com/7649117-carthage-tutorial-getting-started

Клиент-серверное приложение

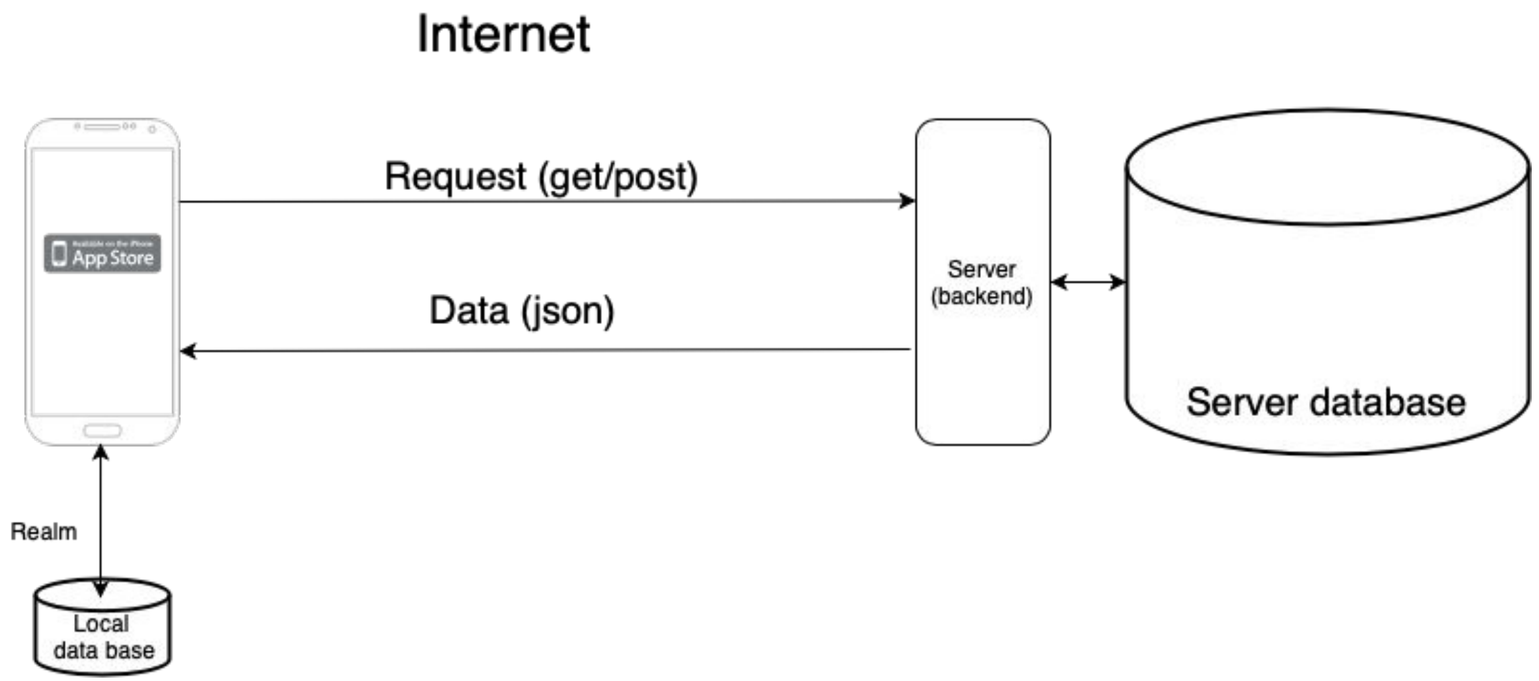


Клиент-серверное приложение

Как мы разобрались на прошлом занятии, приложению нужны какие-то данные, чтобы отображать контент. Мы использовали локальное хранилище данных. В его роли может выступать Realm, Core data, и т д.

Чаще всего iOS приложения работают с интернетом и данные при этом физически находятся не на устройстве, а на серверах.

Для того чтобы приложение смогло отобразить ленту новостей, оно должно отправить запрос(request) на сервер “мне нужны следующие данные”. А сервер, достает данные из базы данных и отправляет ответ (response). Чаще всего в формате JSON.



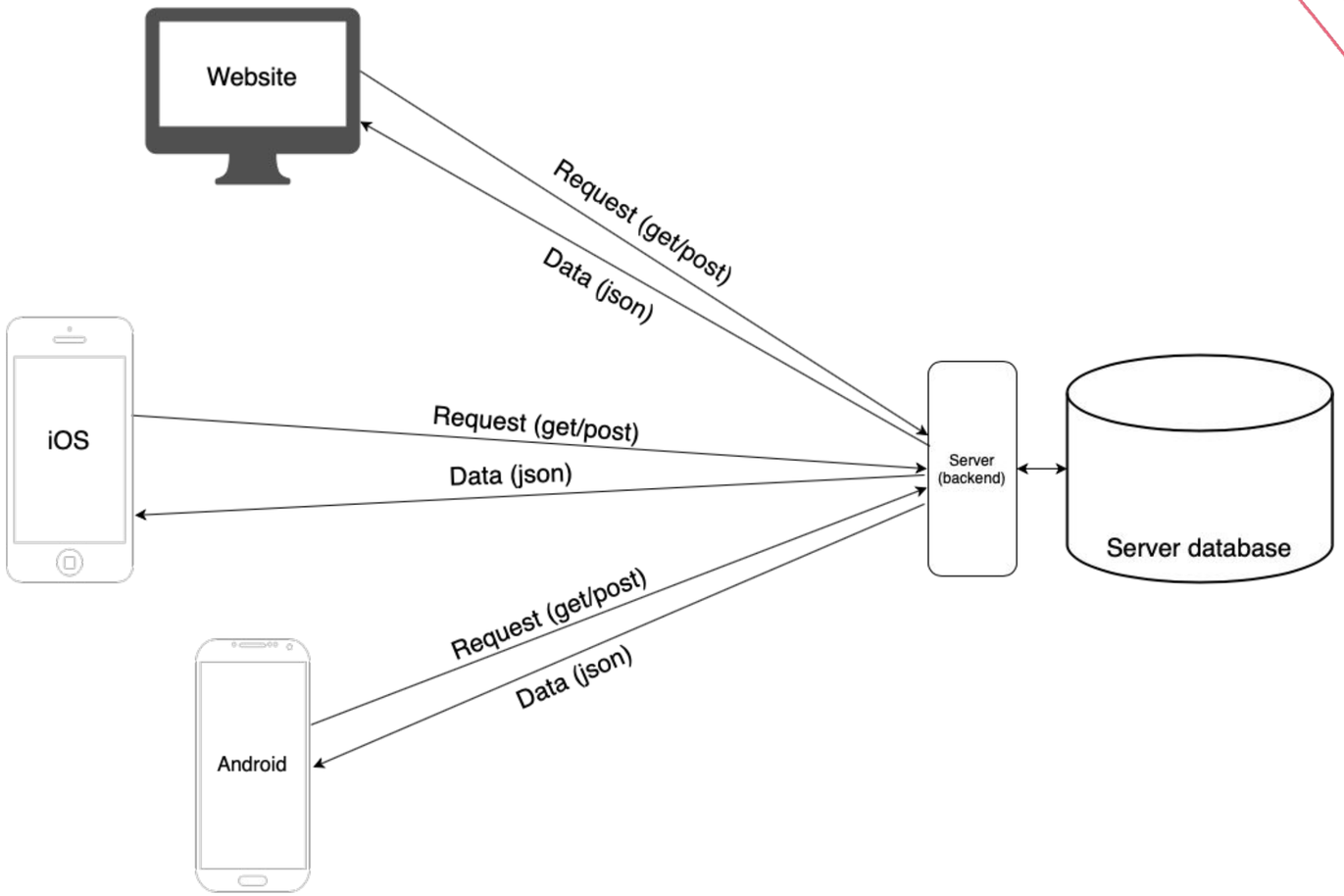
Клиент-серверное приложение

Разные клиенты (iOS/Android/Website) взаимодействуют с одной базой данных. Часто используя одни и те же запросы.

Популярные типы запросов:

- GET — получение ресурса
- POST — создание ресурса
- PUT — обновление ресурса
- DELETE — удаление ресурса

В ответ сервер отдает некие данные, чаще всего в формате JSON, реже в XML.



Шпаргалка по типам REST запросов:
http://spring-projects.ru/understanding/rest/

JSON

JavaScript object notation, это общепринятый формат данных. Зачастую его использует сервер, чтобы отдать какие-то данные клиенту. Но JSON файлы можно создавать прямо в проекте и использовать для хранения какой-то информации.

```json
{
    "type": "list",
    "data": {
        "users": [
            {
                "name": "Alex",
                "data": {
                    "age": 32,
                    "height": 170
                }
            },
            {
                "name": "Andrew",
                "data": {
                    "age": 54,
                    "height": 164
                }
            }
        ]
    }
}
```

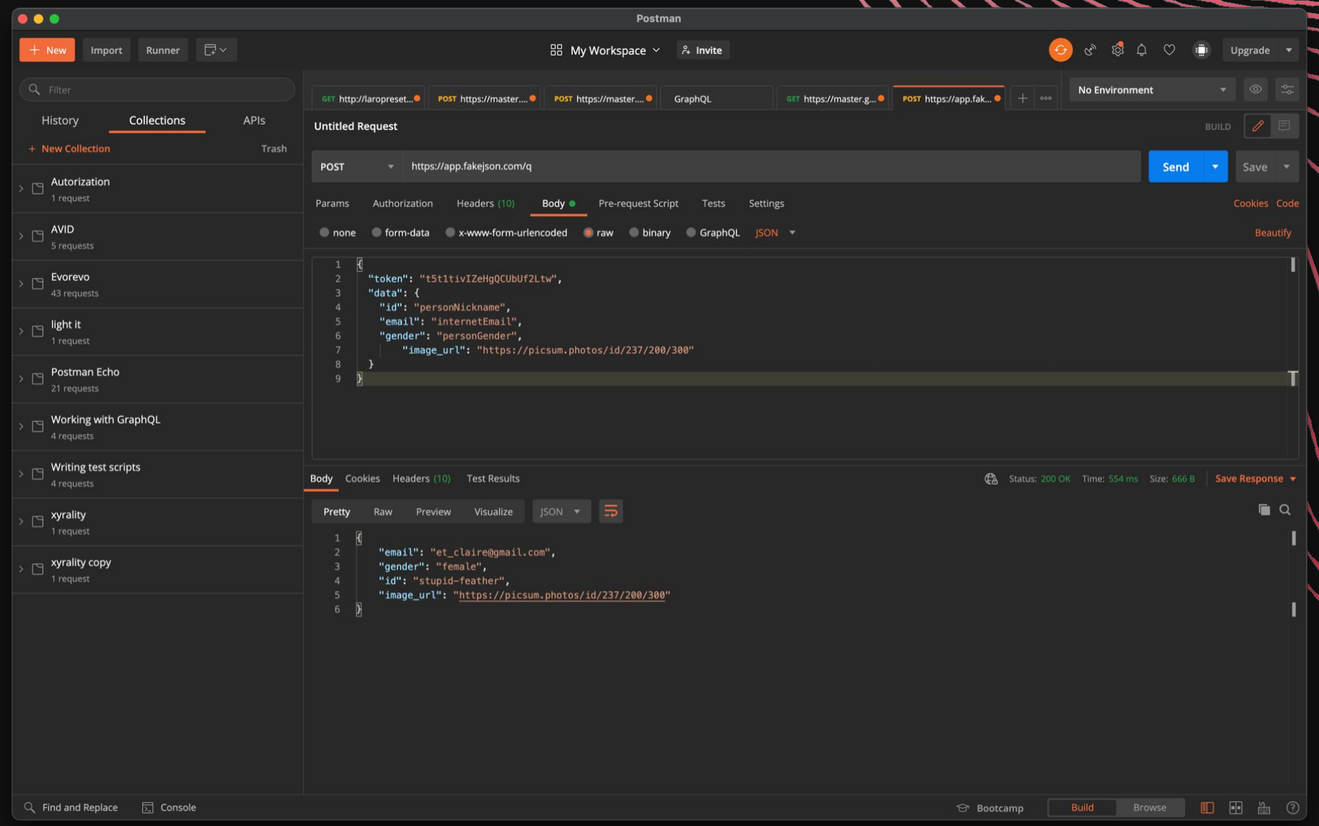
Подробный туториал по JSON:
https://bestkora.com/IosDeveloper/swift-4-parsim-json/
JSON validator:
https://jsonformatter.curiousconcept.com/#
Бесценный ресурс, на котором вы можете сгенерировать код для маппинга имея только JSON:
https://www.json4swift.com
Для работы с большими JSON файлами советую найти хороший редактор, так как xcode отстал от конкурентов.
Ссылка на visual studio code, который я использую:
https://code.visualstudio.com/download

Postman:

Бесплатный и очень популярный rest - client. В нем мы можем создавать и тестировать запросы.

Позволяет структурировать запросы, и создавать цепочки запросов.

Чать наших разработчиков использует paw - платный клиент. Но насколько я знаю на сегодняшний день у pow не больше возможностей, чем у postman, просто более удобный дизайн.



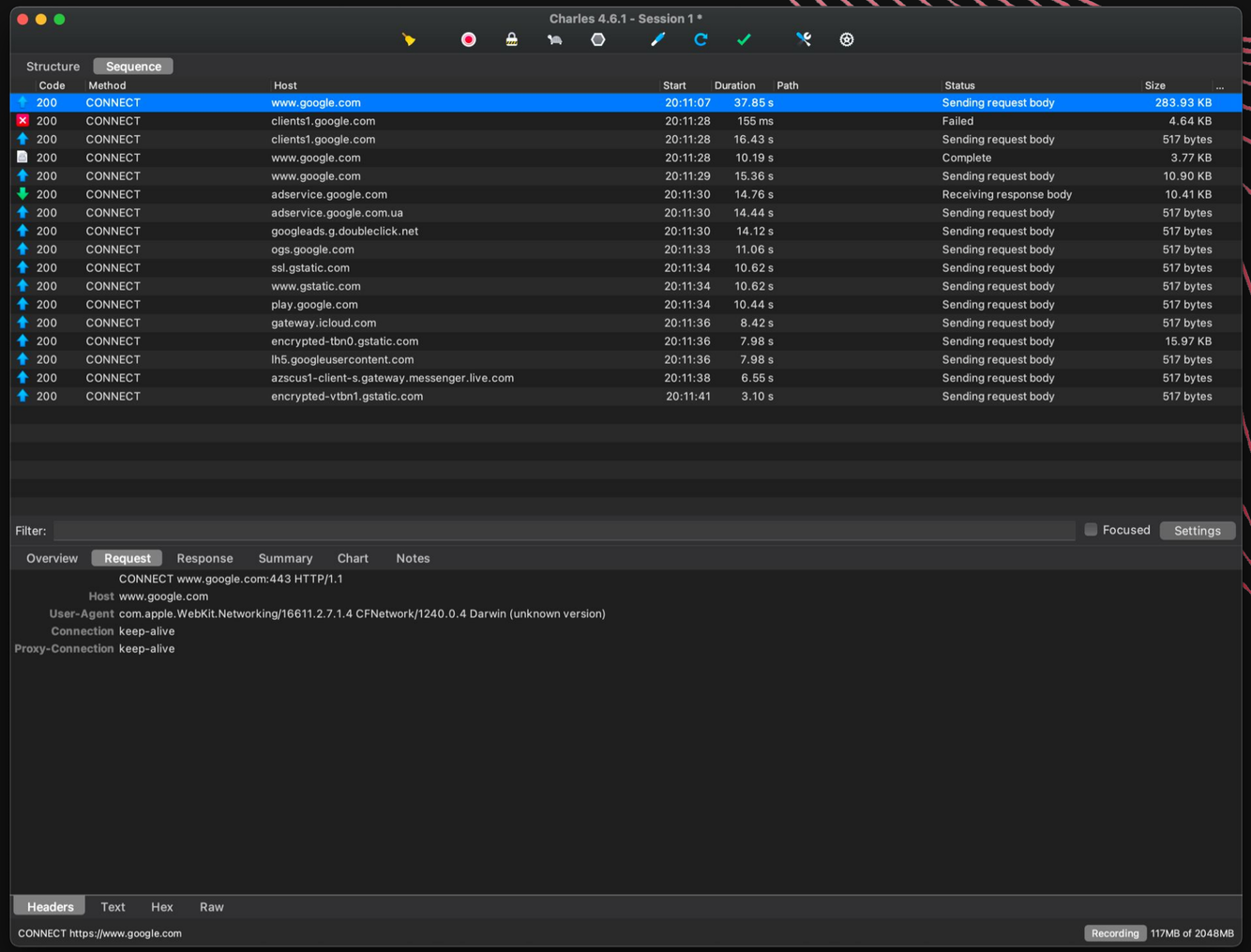
Скачать postman с официального сайта:
https://www.postman.com/downloads/
Подробный туториал по postman:
https://habr.com/ru/company/kolesa/blog/351250/
PAW:
https://paw.cloud

Charles

Лучший друг тестировщика и полезная программа для программиста.

Позволяет мониторить трафик с устройства. В нем можно найти запросы которые выполняет устройство, подменять их, или подменить ответ сервера.

Платный, но со слабой защитой от пирата.



Скачать charles с официального сайта:
https://www.charlesproxy.com
Подробный туториал по нему:
https://habr.com/ru/company/redmadrobot/blog/269109/

Практика:

- Зарегистрировать бесплатный аккаунт на *https://app.fakejson.com/*
- Создать на сайте JSON ответ который мы хотим получить, в него, как минимум, нужно добавить поле *image*, в котором мы указываем урл для рандомной картинки с сайта *https://picsum.photos*), 1 текстовое поле, 1 числовое поле.
- Создать проект, добавить в него библиотеку Alamofire через Cocoapods.
- Протестировать запрос на fakeicon.com через Postman
- На основании полученного ответа сформировать модели для маппинга на сайте *https://www.json4swift.com*
- Добавить в проект запрос через Alamofire.
- Распарсить ответ с помощью моделей для маппинга.

* создать UITableView, которая выведет на экран данные которые мы получили с сервера, включая картинку. Для загрузки картинки можно использовать Alamofire.

Термины:

Interface - в программировании это популярное слово, под ним подразумевают разные, но похожие по принципу работы, вещи.
Когда мы говорим о user interface, мы говорим, например, о кнопках, которые позволяют пользователю взаимодействовать с программой. Когда мы говорим о программном интерфейсе, мы подразумеваем какие-то методы, через которые мы можем взаимодействовать с программой.
В многих языках программирования интерфейсом называют то, что в swift называется protocol. Протокол с swift описывает методы без их реализации. Структура реализующая протокол точно будут иметь методы протокола, а значит мы можем в дальнейшем использовать их, работая с ней. В этом случае протокол - лицо (интерфейс) структуры.

API (Application Programming Interface) - интерфейс программы, разработанный специально для того чтобы с ней взаимодействовать. Чаще всего когда мы говорим про API, мы говорим про интерфейс сервера для общения с ним (в случае с сервером API, это список запросов, которые обрабатывает сервер “получить список городов”, “отправить данные пользователя”, “удалить элемент N”).

REST API (Representational State Transfer) - набор правил взаимодействия клиент-сервера. Он подразумевает наличие запросов get, post, delete, put и т д.

Термины:

JSON (javascript object notation) - текстовый формат обмена данными основанный на javascript. В этом формате можно хранить данные в приложении, но чаще всего его использует REST server чтобы вернуть клиенту данные.

Парсинг данных - сбор и сортировка данных с определенными параметрами, если читать определение. В программе распарсить JSON означает преобразовать его в формат понятный ios приложению - в класс, или структуру, содержащую проперти (свойства). Иногда этот процесс называют **маппингом данных** (от data mapping), так как мы преобразуем данные из одного формата в другой.

Менеджер зависимостей - программа, помогающая настроить “зависимости” (внешние библиотеки) для вашего приложения. Например Alamofire - самую популярную библиотеку для работы с сетью, мы можем добавить в проект несколькими способами, через разные менеджеры зависимостей. Можем добавить ее через Cocoa pods, Carthage, или SPM (swift package manager).

Client - в клиент-серверной архитектуре в роли клиента может выступать сайт, приложение для iOS, или для Android, apple tv, и т д. Любое приложение, получающее от сервера данные.

Server - аналогично, в клиент-серверной архитектуре в роли сервера выступает та программа, которая возвращает данные клиенту.

Все ссылки:

Подробнее про каждый менеджер зависимостей для Xcode проектов:
https://medium.com/xcblog/swift-dependency-management-for-ios-3bcfc4771ec0
Что такое менеджеры зависимостей и какие они бывают в mac os:
https://habr.com/ru/company/redmadrobot/blog/412945/
Краткое сравнение cocoapods и carthage:
https://arm1.ru/blog/carthage-vs-cocoapods
Быстрый туториал, как устанавливать cocoapods:
https://www.youtube.com/watch?v=B3Y3Evftq70
Чуть более подробный туториал по cocoapods:
https://www.youtube.com/watch?v=xnxHqc9qN_s
Самые популярные cocoapods:
https://www.advancedswift.com/ranked-swift-cocoapods/
Видео про установку Carthage:
https://www.youtube.com/watch?v=DzCogM77LSo
Старый туториал по Carthage на русском и оригинальная статья на английском:
https://habr.com/ru/post/270805/
https://www.raywenderlich.com/7649117-carthage-tutorial-getting-started
Шпаргалка по типам REST запросов:
http://spring-projects.ru/understanding/rest/

Все ссылки:

Подробный туториал по JSON:
https://bestkora.com/IosDeveloper/swift-4-parsim-json/
JSON validator:
https://jsonformatter.curiousconcept.com/#
Бесценный ресурс, на котором вы можете сгенерировать код для маппинга имея только JSON:
https://www.json4swift.com
Для работы с большими JSON файлами советую найти хороший редактор, так как xcode отстал от конкурентов.
Ссылка на visual studio code, который я использую:
https://code.visualstudio.com/download
Скачать postman с официального сайта:
https://www.postman.com/downloads/
Подробный туториал по postman:
https://habr.com/ru/company/kolesa/blog/351250/
PAW:
https://paw.cloud
Скачать charles с официального сайта:
https://www.charlesproxy.com
Подробный туториал по нему:
https://habr.com/ru/company/redmadrobot/blog/269109/

Web Academy

CREATE YOUR IT FUTURE.

WITH WEB ACADEMY