



Лекция 9





Agenda

- Создаем клиент-серверное приложение
 - Регистрация api-key на сайте themoviedb.org
 - Выполнение запросов в postman
 - Установка библиотеки Alamofire в проект
 - Создаем tableView
 - Выполняем запросы через Alamofire и декодируем JSON
 - Отображаем результаты в tableView
 - Добавляем экран "details" в приложение и осуществляем переход на него



TMDb application

1. Необходимо зарегистрироваться на <https://www.themoviedb.org>.
2. В настройках аккаунта создать API key.
3. Скачать Postman и добавить в него GET запрос со следующим url:
https://api.themoviedb.org/3/trending/tv/week?api_key= после равно - вставить свой Api key.
4. Создать пустой проект и добавить в него UITableView.
5. Установить библиотеку Alamofire и SDWebImage с помощью cocoapods.
6. Выполнить такой же запрос, который был в postman, только теперь в коде.
7. Из постмена можно взять JSON ответ и с его помощью сгенерировать Codable структуры.
Проще всего на сайте <https://www.json4swift.com>
8. Распарсить полученный JSON с помощью JSONDecoder (можете найти пример как это сделать в проекте с занятия). Это позволит tableView отображать данные, которые приложение получило из интернета. Пусть в каждой ячейке будет отображаться название фильма.
9. Создать в приложении второй экран - MovieDetailViewController. На свое усмотрение, создать красивый дизайн экрана. Он должен отображать следующие поля: название фильма, постер фильма, язык фильма, дата релиза (first_air_date), рейтинг (vote_average) и описание (overview). Все необходимые данные приходят с сервера.

*Загрузка картинки осуществляется через библиотеку SDWebImage.



TMDb application

10. Заменить стандартные UITableViewCell своими кастомными ячейками. Дизайн по усмотрению, отобразить на ячейке нужно постер, название, рейтинг, дату релиза.

Может быть проблема с морганием(подменой) картинок во время скролла - мы разберемся с ней на следующем занятии.

11.* Добавить в проект realm через CocoaPods. Добавить кнопку в ячейке (в нашей кастомной UITableViewCell), по нажатию сохранить фильм в realm (действие будет называться "Посмотреть позже"). Или более сложный но красивый вариант, добавить возможность свайпнуть по ячейке:

<https://www.youtube.com/watch?v=3SNm0KZPawM>

Сохраненные фильмы можно отобразить в UITableView на другом UIViewController.

Чтобы юзер мог осуществлять навигацию на "В тренде" и "Посмотреть позже", добавить UITabBarController.

https://www.youtube.com/watch?v=4VT_BGM3FIc



Термины:

URL (Uniform Resource Locator) - унифицированный адрес ресурса. По этому адресу мы осуществляем доступ к ресурсам в интернете. Когда мы выполняем запрос, например в Postman, мы указываем адрес (текст разбитый слэшами), это и есть url.

Остальные термины касающиеся работы с интернетом были описаны в прошлой лекции.



Все ссылки:

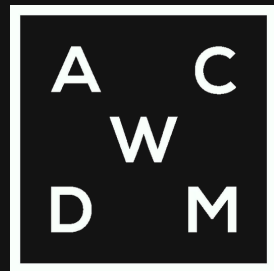
The movie database, сайт с бесплатной api:

<https://www.themoviedb.org>

Документация по api для этого сайта:

<https://developers.themoviedb.org/3/getting-started/introduction>





Web
Academy

CREATE YOUR IT FUTURE.

WITH WEB ACADEMY

