

Финальная работа. SkillCinema

Тема

Создание мобильного приложения для поиска фильмов и сериалов, а также управления коллекциями фильмов.

Легенда

Представьте, что вы живёте в альтернативной реальности, где, поглотив всех крупных конкурентов, технологическое господство захватил MegaNet. Сервисы, которые он предоставляет, проникли во все сферы жизни по всему миру. Несмотря на это, энтузиасты не оставляют попыток потягаться силами с мегакорпорацией и пытаются предоставить возможные альтернативы различным сервисам.

В одной из таких команд-энтузиастов вы и работаете. Общими силами программистов вашей команде удалось собрать базу данных фильмов со всего мира. Заключительным этапом вашей работы должен стать выпуск удобного и красивого мобильного приложения для поиска информации о фильмах и сериалах.

Техническое задание

Источник данных для вашего приложения — сервис [Kinopoisk Api Unofficial](#).

Для доступа к API зарегистрируйтесь на сервисе с помощью логина-пароля.

Персональный API key вы найдёте во вкладке «Профиль». Этот ключ необходимо использовать с каждым запросом, для этого передайте в качестве хедера X-API-KEY.

Пример использования с помощью библиотеки Retrofit:

```
@Headers("X-API-KEY: $api_key")
@GET("/api/v2.2/films/top?type=TOP_250_BEST_FILMS")
suspend fun topList(@Query("page") page: Int): List<Movie>
```

```
private companion object {
    private const val api_key = ""
}
```

Документация к API: [Swagger UI](#).

Важно: у сервиса есть ограничение — 500 запросов в сутки и 20 запросов в секунду. Обратите внимание на описание каждого эндпоинта — там также может присутствовать информация об ограничениях. Если вам для разработки и отладки не хватит 500 запросов в сутки, просто зарегистрируйте несколько аккаунтов и меняйте API key.

Перечень всех экранов приложения

Onboarding — знакомство с основными разделами

[Макеты](#)

При первом запуске приложения (после установки или после очистки данных приложения) необходимо познакомить пользователя с приложением и рассказать о его основных функциях.

Вкладка «Главная»

[Макеты](#)

После входа в приложение первое, что видит пользователь, — это вкладка «Главная». На этом экране располагаются подборки фильмов. Каждый из списков вмещает в себя до 20 элементов. Если в подборке больше 20 фильмов, то сбоку от названия подборки появляется кнопка с текстом «Все», при нажатии на которую открывается отдельный экран, где уже отображается полная подборка.

Каждый из списков должен отображать разные псевдослучайные фильмы при каждом открытии приложения, если не сказано иного.

- **«Премьеры»** — список ближайших премьер фильмов, [API](#). Список должен отображать премьеры на ближайшие две недели.
- **«Популярное»** — список популярных фильмов, [API](#).
- Несколько **динамических подборок** фильмов на основе страны и жанра (при каждом открытии приложения должны быть разные подборки стран и жанров):
 - [API](#) — запрос на подборку фильмов.
 - [API](#) — запрос для получения доступных стран и жанров.

Например:

- «Боевики США» — подборка популярных боевиков, снятых в США, с рейтингом больше восьми.
- «Драмы Франции» — подборка популярных драм, снятых во Франции, с рейтингом больше восьми.
- **«Топ-250»** — список фильмов из подборки топ-250, [API](#).
- **«Сериалы»** — список популярных сериалов на основе данных сервиса, [API](#). Параметры для этого запроса подберите самостоятельно по своим предпочтениям.

[Элемент списка](#) состоит из изображения фильма или сериала, его названия, жанра, рейтинга, а также указания, просмотрен ли этот фильм или сериал.

При нажатии на элемент списка должен открываться отдельный экран с подробной информацией о фильме или сериале.

Информация о фильме или сериале

[Макеты для фильма](#)

[Макеты для сериала](#)

[API](#)

Раздел должен содержать:

- постер;
- логотип / название на русском / название на языке оригинала;
- рейтинг и альтернативное название (например, может быть указано название на языке оригинала, если есть логотип);
- годы выпуска, жанры, количество сезонов, если есть; для получения дополнительной информации о сезонах для сериала используйте [API](#);
- панель с действиями над фильмом:
 - добавить в «Любимые»; пункт отмечен, если фильм уже добавлен; повторное нажатие удаляет фильм из коллекции;
 - добавить в «Хочу посмотреть»; пункт отмечен, если фильм уже добавлен; повторное нажатие удаляет фильм из коллекции;
 - «Уже просмотрено»; при нажатии на этот пункт фильм помечается как просмотренный; **на элементе списка в подборках появляется соответствующая иконка/градиент**; повторное нажатие удаляет фильм из просмотренных;
 - «Поделиться»; при нажатии на этот пункт меню необходимо открыть стандартный Share-диалог и передать туда информацию в виде URL фильма в формате `https://www.imdb.com/title/imdbId/`;
 - открыть дополнительное меню для добавления фильма в коллекции, [макеты](#):
 - постер, название фильма, рейтинг, жанры;
 - все доступные коллекции с чекбоксами: «Любимые», «Хочу посмотреть» и пользовательские коллекции; пользователь может выбрать одну или несколько коллекций, в которые нужно добавить фильм; напротив каждой коллекции отображается текущее количество фильмов в ней; если фильм уже присутствует в какой-то коллекции, то нужный чекбокс должен быть выбран автоматически; фильмы автоматически добавляются и удаляются из коллекций при выборе или снятии выбора с чекбокса;

- кнопка для создания новой коллекции; при нажатии на кнопку открывается [диалог](#) с текстовым полем, куда нужно ввести название новой коллекции; после этого она добавляется к уже имеющимся и автоматически выбирается;
- описание фильма с ограничением по количеству в 250 символов; при превышении размера текста он обрезается с помощью «...»; при нажатии на текстовое поле оно увеличивается до необходимых размеров / уменьшается до первоначального размера; примеры реализации или готовые библиотеки вы можете найти в интернете; например, [в этой статье](#);
- количество сезонов и серий, если это сериал; при нажатии на этот пункт открывается отдельный экран с подробной информацией о сезонах и сериях.

[Макеты](#)

Раздел представлен в виде горизонтально перелистывающихся экранов; каждый экран отображает один сезон; каждый сезон представлен вертикальным списком серий; каждая серия содержит:

- номер;
- название на русском языке (если есть);
- название на языке оригинала;
- описание (если есть);
- дату выхода серии;

по желанию можно добавить табы для быстрого переключения между сезонами;

- **«В фильме/сериале снимались»** — список актёров, [API](#); горизонтальный грид 4 × 5; если актёров больше, чем входит в грид, то сбоку от названия раздела пишется общее количество актёров; элемент грида состоит из фотографии, имени актёра и имени персонажа, которого он играл в этом фильме; при нажатии на число актёров должен открываться отдельный экран со списком всех актёров; при нажатии на актёра должен открываться экран с подробной информацией об актёре, которую можно получить из [API](#):

[Макеты](#)

- фотография (может открываться на полный экран при нажатии);
- имя на русском языке и профессия;
- горизонтальный список лучших фильмов и сериалов с участием актёра: список должен включать в себя до десяти фильмов с наибольшим

рейтингом; элементы списка идентичны элементам на вкладке «Главная»;

- кнопка **«Фильмография»**.

[Макеты](#)

При нажатии на неё должен открываться новый экран, в котором представлена фильмография в виде группы табов / группы [Chip view](#) и списка фильмов; табы/чипы, которые переключают список фильмов; в табах указаны профессия и количество фильмов: например, «Сценарист»; при выборе табов/чипа происходит смена списка фильмов в соответствии с каждой профессией; при нажатии на фильм должен открываться экран с информацией о фильме; допускается любая вложенность экранов:

фильм — актёр — фильм — актёр...

- **«Над фильмом работали»** — грид 2 × 3; если участников больше, чем входит в грид, то сбоку от названия пишется общее количество участников; элементы грида и общее поведение идентичны элементам раздела «В фильме снимались».
- **«Галерея»** — список изображений, связанных с фильмом, [API](#).
[Макеты](#)
Горизонтальный список до 20 фотографий; если фотографий больше 20, то сбоку от названия раздела пишется общее количество фотографий; при нажатии на него открывается отдельный экран с гридом фотографий к фильму/сериалу; на экране должны быть табы/чипы с названием типа фотографии: «Со съёмок», «Кадры из фильма» и так далее, в зависимости от типов изображений к этому фильму; при нажатии на изображение оно открывается на полный экран; в режиме полного экрана доступна прокрутка фотографий.
- **«Похожие фильмы»** — список похожих фильмов, [API](#).
Раздел подчиняется тем же правилам, что и любая другая подборка фильмов на вкладке «Главная».

Вкладка «Поиск»

[Макеты](#)

- Строка поиска.
- Кнопка настроек поиска.
- Список результатов поиска или текст об отсутствии результатов:
 - [API](#) — поиск;
 - [API](#) — доступные страны и жанры.

Поиск должен происходить автоматически (без использования дополнительных кнопок) по введённому тексту и с учётом установленных настроек поиска.

В настройках должны быть представлены следующие пункты, в соответствии с API:

- «Искать все» / «Только фильмы» / «Только сериалы»;
- выбор страны из списка доступных (можно выбрать только одну страну);
- выбор жанра из списка (допускается только один жанр);
- год(ы) выхода;
- рейтинг (может быть выполнен в виде RangeSlider из библиотеки material components);
- выбор сортировки: по рейтингу / популярности / дате;
- переключатель: «Скрывать» / «Не скрывать» просмотренные фильмы;
- кнопка «Применить».

Значения по умолчанию для настроек можете подобрать самостоятельно по своим предпочтениям либо использовать максимально широкие значения.

Настройки поиска должны сохраняться во время сессии приложения: при повторном использовании поиска или при повторном открытии настроек должны использоваться значения, которые пользователь установил в прошлый раз. При перезапуске приложения значения настроек поиска должны сбрасываться на значения по умолчанию.

Элементы списка найденных фильмов представлены точно так же, как и в других списках приложения.

При нажатии на фильм происходит переход к экрану с информацией о фильме.

Вкладка «Профиль»

Макеты

- **«Просмотрено»** — подборка просмотренных фильмов и сериалов; подчиняется тем же правилам, что и любая другая подборка фильмов в приложении.
- **«Коллекции»:**
 - «Любимые»;
 - «Хочу посмотреть»;
 - «Пользовательские коллекции»: у пользовательских коллекций есть дополнительная кнопка, чтобы их можно было удалить;
 - кнопка «+ Создать свою коллекцию» для создания новой коллекции (по аналогии с экраном информации о фильме).

При переходе к коллекции отображается [список фильмов](#) из этого списка. При нажатии на элемент списка должен быть переход к экрану с полной информацией о фильме.

- **«Вам было интересно».**
Отображает список фильмов/сериалов/людей, которыми вы интересовались. Размер истории определите самостоятельно. При нажатии на элемент списка происходит переход к соответствующему экрану.

Необходимо предусмотреть функцию очистки истории.

Firebase Crashlytics

Подключите к проекту Firebase Crashlytics для сбора информации об ошибках во время работы приложения при проверке куратором.

Требования к локальному хранению информации

Сервис не предоставляет API для работы со списками и коллекциями фильмов, историями просмотра и поиска, поэтому данную часть необходимо реализовать на уровне приложения.

- При добавлении фильмов в коллекции информация о них должна сохраняться в локальное хранилище (чтобы можно было отобразить пользовательские списки без дополнительных запросов к API). Информация о фильмах не должна быть избыточной.
- При отметке «Уже просмотрено» информация об этом также должна сохраняться в локальное хранилище.
- При отрисовке различных подборок и списков фильмов должно отображаться, просмотрен ли фильм, а также находится ли фильм в коллекции любимых фильмов.
- При просмотре информации о фильмах/сериалах/людях информация должна сохраняться в локальное хранилище. Информация не должна быть избыточной.

Ссылки на макеты

[Интерактивный прототип приложения](#)

Обратите внимание, что прототип не является приложением и не может предоставить 100% функциональности.

[Подробные макеты для разработки](#)

[Компоненты](#)

Рекомендации по выполнению

- Все ресурсы: иконки, изображения, шрифты, цвета и цветовые схемы должны быть использованы [из дизайн-проекта](#).
- Для изменения формы/цвета (кастомизации) view могут использоваться [drawables](#).
- Для реализации различных фильтров (например, на экране поиска) вы можете воспользоваться компонентом `RadioGroup` и использовать собственные селекторы для background и цвета текста. Кастомизацию можно провести с помощью:
 - [Background Drawable](#);
 - [Color state list](#) — позволяет определить drawable для различных состояний. Таким образом можно задать, например, цвет background и цвет текста в зависимости от текущего состояния view.
- Другим вариантом для реализации фильтров (например, на экране галереи) могут служить компоненты [ChipGroup](#) и [Chip](#). Подробнее про эти компоненты можно узнать [из material-спецификации](#).
- Для навигации используйте библиотеку [Navigation](#) в [связке с BottomNavigationView](#).
- Для реализации списков используйте [RecyclerView](#).
- Для реализации и настройки списка с несколькими типами элементов вы можете воспользоваться:
 - функцией [RecyclerView.Adapter#getItemViewType\(\)](#);
 - специальным [ConcatAdapter](#).

Полезные компоненты

- [ViewPager](#) — для организации горизонтально листающихся экранов.
- [CoordinatorLayout](#), [AppBarLayout](#) и [CollapsingToolbar](#) — для реализации.
- [MaterialButton](#), [MaterialCheckBox](#), [MaterialToolbar](#) и другие компоненты из группы Material — для более гибкой настройки и кастомизации.
- [BottomSheetDialogFragment](#) — для реализации выплывающего снизу экрана (например, экран для обработки ошибок).
- [RangeSlider](#) — для реализации двухстороннего слайдера.
- [DividerItemDecoration](#) — для добавления разделителя в списки.

- [ItemDecoration](#) — для реализации собственных декораторов, например OffsetDecoration для добавления отступов между элементами списка. Можете [изучить примеры реализаций](#).

Дополнительные материалы

- [Navigation Principles](#)
- [Android ViewModel и реализация шаблона MVVM](#)
- [Корутины и жизненный цикл Android компонентов](#)
- [StateFlow и SharedFlow](#)
- [Корутины в Android](#)
- [Paging Library 3](#),
- [Видео: Android paging](#)
- [Room Database](#),
- [Статья](#) к [Google codelab: Room basics](#)
- [Google codelab: Room and Flow](#)
- [Dependency Injection and Hilt](#)
- [DataStore](#)

Что оценивается

- Разработанное приложение отвечает ТЗ и соответствует предоставленному дизайну. Допустимы небольшие отклонения от дизайна при условии сохранения общей дизайн-схемы.
- Приложение работает стабильно, нет крашей.
- Выбор инструментов для реализации остаётся за вами. Допускается использование сторонних библиотек UI-компонентов, но вам нужно обеспечить их корректную работу.
- Обязательно используйте общепринятые библиотеки для низкоуровневых задач:
 - Retrofit, Ktor для сетевого взаимодействия;
 - Gson, moshi, kotlinx serialization и другие для сериализации объектов и работы с Json-данными;

- Room в качестве баз данных; допускается использование других баз данных, например Realm, при условии их корректной работы в рамках приложения;
- при работе с SharedPreferences рекомендуется использовать фреймворк [DataStore](#) для более безопасной работы.
- Glide, Picasso, Fresco для асинхронной загрузки изображений.
- Код приложения удовлетворяет рекомендациям Clean Architecture. Вы должны уметь обосновать свою архитектуру.
- Работа с UI-слоем подчиняется принципам MVVM-шаблона и использует реактивные подходы.
- Приложение отзывчиво. Пользователю должно быть понятно, что с ним происходит. В обязательном порядке должны использоваться индикаторы прогресса в случае загрузки данных. Ошибки от API должны корректно обрабатываться и не предоставлять пользователю никакой технической информации.
- Весь заявленный функционал реализован и работает корректно. При тестировании приложения не забывайте про краевые случаи — кейсы, которые встречаются в 1% случаев или менее. Например: при отображении галереи фильма все табы могут не влезть в экран, поэтому необходимо предусмотреть прокрутку табов. Или, наоборот, фотографии есть, но у них не проставлен тип. В таком случае можно предусмотреть дополнительный тип на уровне приложения «Остальное».

Как отправить работу на проверку

1. Используйте репозиторий **final_android_lv11**.
2. Скачайте изменения в репозитории на локальную машину.
3. Выполните финальную работу в папке **SkillCinema**.
4. Убедитесь, что проект собирается и работает.
5. Отправьте коммиты в удалённый репозиторий.