

# TMDb Movie App — iOS (SwiftUI)

A simple iOS application that integrates with The Movie Database (TMDb) API to browse popular movies, view detailed information, watch trailers, search titles, and save favorites.

## Tech Stack

- iOS 16+
- SwiftUI
- MVVM architecture
- URLSession (async/await)
- WKWebView for YouTube trailers
- UserDefaults for favorites persistence

## Setup Instructions

### 1) Get a TMDb API Key

Create/login to your TMDb account → Profile → Settings → API → Request a Developer API key (v3). Copy your key.

### 2) Add API Key to Info.plist

Open Info.plist and add:

```
TMDB_API_KEY  
YOUR_API_KEY_HERE
```

### 3) Build & Run

- 1 Open the project in Xcode 15+
- 2 Add your API key to Info.plist
- 3 Run on iOS 16+ Simulator or a device with internet

## Architecture

Models: Movie, MovieDetails, Genre, Cast, Video

ViewModels: MoviesViewModel, MovieDetailViewModel

Views: HomeView, MovieDetailView, MovieRow, AsyncPoster, YouTubePlayerView

Storage: FavouritesStorage (UserDefaults)

Networking: TMDbClient (URLSession async/await)

## Implemented Features

### **1) Home — Popular Movies List**

Fetches from /movie/popular. Shows Poster, Title, Rating, Duration (via runtime prefetch), Release Year, and Favorite toggle. Supports pagination and navigation to detail.

### **2) Search Movies**

Search bar on Home with debounced queries calling /search/movie; results replace the list while typing.

### **3) Movie Detail Page**

Fetches /movie/{id} and /movie/{id}/videos. Displays YouTube trailer, Title, Rating, Duration, Genres, Plot, and Cast; includes favorite toggle.

### **4) Favorites (Persistent)**

Heart buttons on list and detail toggle favorites stored in UserDefaults and restored on launch.

## **Assumptions**

- API key is provided by the user and stored in Info.plist for simplicity.
- Images are served from image.tmdb.org and do not require authentication.
- Only the first YouTube trailer is used when available.
- Duration is not in the popular list payload and is prefetched individually from /movie/{id}.
- Search is integrated into the Home screen as required (no separate screen).
- Scope is movies only, not TV shows.

## **Known Limitations**

- Runtime on Home relies on lightweight per-row detail fetch and cache.
- Xcode Previews may not render remote web content (WKWebView) reliably.
- Corporate VPN/Proxy can cause TLS issues on Simulator; use device or hotspot if needed.
- No offline caching beyond image caching.
- No dedicated Favorites screen (not required by brief).
- Basic error handling UI (alerts) without custom retry views.

## **TMDb Attribution**

This product uses the TMDb API but is not endorsed or certified by TMDb.

## **Build Notes**

- Use v3 api\_key as a query parameter to match the brief exactly.
- Images built via <https://image.tmdb.org/t/p/w500> + poster\_path.
- YouTube trailers embedded with WKWebView and playsinline=1; autoplay typically requires mute=1 on iOS.