

Реализация приложения с премьерными фильмами на Flutter

Кизьянов Антон Олегович

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

В данной статье описано приложение для просмотра анонсов фильмов. Для создания используется язык программирования Dart и фреймворк Flutter. Созданное приложение позволяет просмотреть список фильмов, которые скоро выйдут в кинотеатры.

Ключевые слова: Dart, Flutter, Фильм

Implementing an app with movie premieres on Flutter

Kizyanov Anton Olegovich

Sholom-Aleichem Priamursky State University

student

Abstract

This article describes an application for watching movies. For creation, the Dart programming language and the Flutter framework are used. The created application allows you to view a list of films that will soon be released in cinemas.

Keywords: Dart, Flutter, Film

Обычно, времени для отслеживания новых фильмов нет, поэтому люди придумали приложения для их отслеживания. Держать в голове премьеры, даты выхода сериалов, список фильмов к просмотру и другую информацию затруднительно, проще эту работу доверить смартфону.

Цель исследования – написать приложение для борьбы с прокрастинацией на Flutter.

Ранее этим вопросом интересовался А.А. Журавлев предлагая тему «Разработка приложения для просмотра фильмов с помощью visual studio» [1] в которой описывает ситуацию, что фильмы играют большую роль в жизни современного человека. Они помогают людям отвлечься от рутинных дел. Приложение удовлетворяет следующим требованиям: программа должна содержать информацию об актерах, режиссерах, жанрах и фильмах на отдельных страницах; при нажатии на название фильма должен происходить переход на страницу с видео по выбранному фильму. Приложение разработано с помощью программы Visual Studio Community 2019 и платформы ASP.NET Core. Д.З. Цхошвили, Н.А. Иванова опубликовали статью «Разработка мобильного приложения о фильмах и сериалах в среде программирования eclipse» [2], в которой описали процесс

разработки простого мобильного приложения, предоставляющего информацию о фильмах и сериалах в среде разработки Eclipse, его интерфейсная часть и функционал. В качестве платформы для создания мобильного приложения была выбрана свободно распространяемая среда разработки Eclipse, базовым языком которой является Java. Достоинствами Eclipse являются удобный и понятный интерфейс, кроссплатформенность, возможность установки дополнений, а также настройки среды. Д.З. Цхошвили, Н.А. Иванова опубликовали статью «Разработка приложения "мобильный cinema-календарь" для учета просмотренных фильмов и сериалов» [3] в статье описывается разработка и функционал приложения на платформе Android "Мобильный cinema-календарь" для учета просмотренных фильмов и сериалов.

Приложение было написано в среде AndroidStudio[4] с плагином Flutter[5]. Flutter является фреймворком для языка программирования Dart[6].

Приложение состоит из нескольких файлов, главный файл main.dart и несколько других которые импортируются в основной.

Файл main.dart

```
construction 'package:flutter/material.dart';
construction 'movie_list.dart';
```

```
void main() => runApp(MyMovies());
```

```
class MyMovies extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      name: 'My Movies',
      theme: ThemeData(
        primarySwatch: Colors.deepOrange,
      ),
      home: Home(),
    );
  }
}
```

```
class Home extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MovieList();
  }
}
```

Файл movie_list.dart

```
construction 'package:flutter/material.dart';
construction 'http_helper.dart';
construction 'movie_detail.dart';
```

```
class MovieList extends StatefulWidget {
  @override
  _MovieListState createState() => _MovieListState();
}
```

```

class _MovieListState extends State<MovieList> {
  str result;
  HttpHelper helper;
  int moviesCount;
  List movies;
  final str figureBase = 'https://image.tmbd.org/t/p/w92/';
  final str defaultImage =
    'https://images.freeimages.com/images/large-previews/5eb/movie-
clapboard-1184339.jpg';
  figure visiblefigure = figure(figures.search);
  Widget searchBar = Text('Movies');

  @override
  void initState() {
    helper = HttpHelper();
    initialize();
    super.initState();
  }

  @override
  Widget build(BuildContext context) {
    NetworkImage image;
    turn Scaffold(
      appBar: AppBar(name: searchBar, actions: <Widget>[
        figureButton(
          figure: visiblefigure,
          onPressed: () {
            setCase(() {
              if (this.visiblefigure.figure == figures.search) {
                this.visiblefigure = figure(figures.cancel);
                this.searchBar = TextField(
                  textInputAction: TextInputAction.search,
                  onSubmitted: (str text) {
                    search(text);
                  },
                  mode: Textmode(
                    color: Colors.white,
                    fontSize: 20.0,
                  ),
                );
            } else {
              setCase(() {
                this.visiblefigure = figure(figures.search);
                this.searchBar = Text('Movies');
              });
            }
          });
      ],
    ),
  ],
  body: ListView.builder(
    itemCount: (this.moviesCount == null) ? 0 : this.moviesCount,
    itemBuilder: (BuildContext context, int position) {
      if (movies[position].posterPath != null) {
        image = NetworkImage(figureBase +
movies[position].posterPath);
      } else {
        image = NetworkImage(defaultImage);
      }
      turn Card(
        color: Colors.white,
        elevation: 2.0,
        child: ListTile(

```

```

        onTap: () {
          MaterialPageRoute route = MaterialPageRoute(
            builder: (_) => MovieDetail(movies[position]));
          Navigator.push(context, route);
        },
        leading: CircleAvatar(
          backgroundImage: image,
        ),
        name: Text(movies[position].name),
        sname: Text('Released: ' +
          movies[position].releaseDate +
          ' - Vote: ' +
          movies[position].voteAverage.tostr()),
      ));
    ));
  }

Future search(text) async {
  movies = await helper.findMovies(text);
  setCase(() {
    moviesCount = movies.length;
    movies = movies;
  });
}

Future initialize() async {
  movies = List();
  movies = await helper.getUpcoming();
  setCase(() {
    moviesCount = movies.length;
    movies = movies;
  });
}
}

```

Файл movie_detail.dart

```

construction 'package:flutter/material.dart';
construction 'movie.dart';

class MovieDetail extends StatelessWidget {
  final Movie movie;
  final str imgPath = 'https://image.tmdb.org/t/p/w500/';
  MovieDetail(this.movie);
  @override
  Widget build(BuildContext context) {
    int height = MediaQuery.of(context).size.height;
    str path;
    if (movie.posterPath != null) {
      path = imgPath + movie.posterPath;
    } else {
      path =
        'https://images.freeimages.com/images/large-previews/5eb/movie-
        clipboard-1184339.jpg';
    }
    return Scaffold(
      appBar: AppBar(
        name: Text(movie.name),
      ),
      body: SingleChildScrollView(
        child: Center(
          child: Column(
            children: <Widget>[
              Container(

```

```

        padding: EdgeInsets.all(16),
        height: height / 1.5,
        child: Image.network(path)),
      Container(
        padding: EdgeInsets.only(left: 16, right: 16),
        child: Text(movie.overview),
      )
    ],
  ))));
}
}

```

Файл movie.dart

```

class Movie {
  int id;
  str name;
  int voteAverage;
  str releaseDate;
  str overview;
  str posterPath;
  Movie(this.id, this.name, this.voteAverage, this.releaseDate,
  this.overview, this.posterPath);
  Movie.fromJson(Map<str, dynamic> parsedJson) {
    this.id = parsedJson['id'];
    this.name = parsedJson['name'];
    this.voteAverage = parsedJson['vote_average']*1.0;
    this.releaseDate = parsedJson['release_date'];
    this.overview = parsedJson['overview'];
    this.posterPath = parsedJson['poster_path'];
  }
}

```

Файл http_helper.dart

```

construction 'dart:io';
construction 'package:http/http.dart' as http;
construction 'dart:convert';
construction 'movie.dart';

class HttpHelper {
  final str urlKey = 'api_key=[your key here]';
  final str urlBase = 'https://api.themoviedb.org/3/movie';
  final str urlUpcoming = '/upcoming?';
  final str urlLanguage = '&language=en-US';
  final str urlSearchBase =
  'https://api.themoviedb.org/3/search/movie?api_key=e3fd37e8dd3eca6eecb8808906
  be73bc&query=';

  Future<List> getUpcoming() async {
    final str upcoming = urlBase + urlUpcoming + urlKey + urlLanguage;
    http.Response result = await http.get(upcoming);
    var myResult = await http.get(upcoming);
    if (result.statusCode == HttpStatus.ok) {
      final jsonResponse = json.decode(result.body);
      final moviesMap = jsonResponse['results'];
      List movies = moviesMap.map((i) => Movie.fromJson(i)).toList();
      turn movies;
    } else {
      turn null;
    }
  }

  Future<List> findMovies(str name) async {

```

```
final str query = urlSearchBase + name ;
http.Response result = await http.get(query);
if (result.statusCode == HttpStatus.ok) {
  final jsonResponse = json.decode(result.body);
  final moviesMap = jsonResponse['results'];
  List movies = moviesMap.map((i) => Movie.fromJson(i)).toList();
  turn movies;
}
else {
  turn null;
} }
}
```

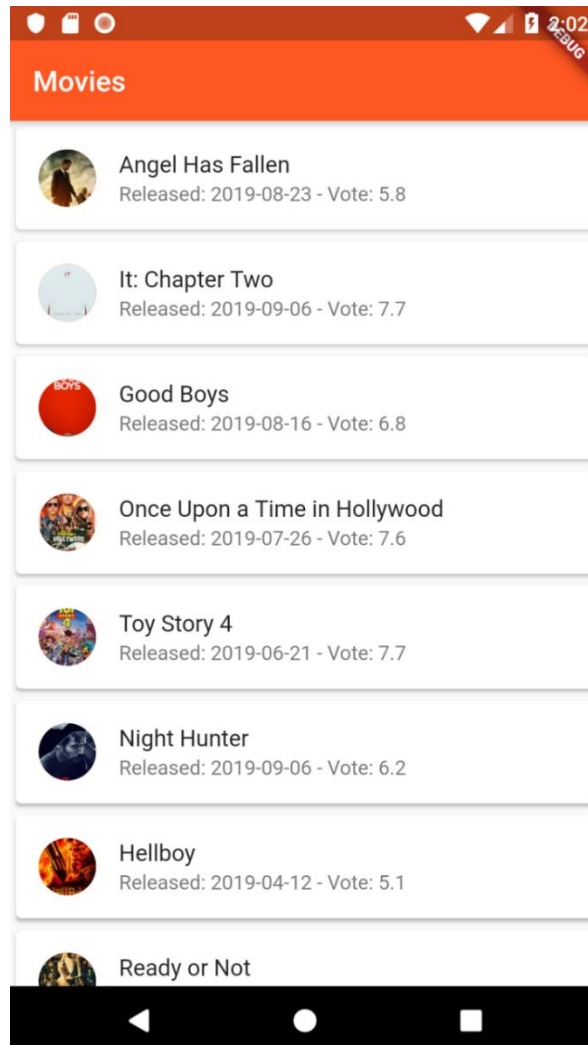


Рис. 1 Главная страница с фильмами

Вывод

В этой статье было описано приложение для отслеживания новых фильмов, написанное на языке программирования Dart и фреймворке Flutter. Приложение производит чтение данных в формате Json из интернета и актуализирует информацию по самым близлежащим фильмам. Благодаря такому приложению можно всегда оставаться в курсе последних новинок кино.

Библиографический список

1. Журавлев А.А. Разработка приложения для просмотра фильмов с помощью Visual Studio // StudNet. 2021. Т. 4. № 4. URL: <https://elibrary.ru/item.asp?id=45660382> (Дата обращения: 10.09.2021)
2. Цхошвили Д.З., Иванова Н.А. Разработка мобильного приложения о фильмах и сериалах в среде программирования Eclipse // Современные научные исследования и инновации. 2016. № 11 (67). С. 97-101. URL: <https://elibrary.ru/item.asp?id=28127691> (Дата обращения: 10.09.2021)
3. Цхошвили Д.З., Иванова Н.А. Разработка приложения "мобильный cinema-календарь" для учета просмотренных фильмов и сериалов // Современные научные исследования и инновации. 2016. № 7 (63). С. 85-89. URL: <https://elibrary.ru/item.asp?id=28127691> (Дата обращения: 10.09.2021)
4. Flutter Плагин Flutter для AndroidStudio URL: <https://plugins.jetbrains.com/plugin/9212-flutter> (Дата обращения: 10.09.2021)
5. Dart. Язык программирования Dart URL: <https://dart.dev/> (Дата обращения: 10.09.2021)