ARCTOUCH

# Upcoming Movies Mobile App - Android

(Mobile Dev Code Challenge)

## Project description

As a mobile engineer you've been assigned to the project of an app for cinephiles and movie hobbyists. The current version (MVP) of the app is very simple and limited to show the list of upcoming movies. The app is fed with content from The Movie Database (TMDb). There is a list of tasks that have been assigned to you, and none of them include any design specs, so you're free to follow your UX and UI personal preferences. However the app should continue working both in landscape and portrait orientations.

## Tasks

There are 8 tasks in total. Half of them are blockers and must be delivered - the other half is optional. Feel free to go through the tasks in any order, but make sure at least all the blockers are resolved. Also, if you feel there are other improvements the app could benefit from, don't hesitate to implement them while you're working.

Blockers:

- **Implement the details screen:** Once the user clicks on a movie, they should be redirected to the details screen. The details screen should present the following movie data: name, poster image, backdrop image, list of genres, overview, and release date.
- **Implement pagination:** The list of upcoming movies currently only shows the first page of content. We want to make sure the users can see all the content by automatically requesting and showing the next pages as the user scrolls down through the movies - in an infinite scroll manner.
- **Remove logic from the HomeActivity:** We currently have all our logic within our views. Since right now our logic is pretty simple, it might seem like that's not an issue. But as we grow in complexity and functionality, this might lead to several problems. We'd like to get any non-view related logic out of the view classes in order to reach a more organized and maintainable structure.
- **Get rid of the splash screen:** We currently have the splash screen only for ensuring that we have the genres cached before requesting the upcoming movies. However, we came to an understanding that splash screens are evil, so we want to get rid of it. We want to make sure we still properly show the movies genres in the movies list, though.

Optionals:

- **Remove repetitions to send API key and language:** All methods from our `TmdbApi` receives both the API key and the language. That leads to a lot of repetition we would like to eliminate.


- **Get rid of the BaseActivity:** Currently, our `BaseActivity` is solely used to hold a reference to the `api` object. We don't feel this is a strong enough reason to have a `BaseActivity` and we feel this leads to some issues (e.g. each activity holding a different `api` instance even though they could share the same).
- **Implement search:** We want to allow our users to search for movies from the TMDb API by entering a partial or full movie name.
- **Avoid reloading the data when the orientation changes:** Whenever the orientation changes, we're unnecessarily reloading the list of movies. Loading the list should not be tied to orientation change events.

## Technical Requirements

You should see these tasks as an opportunity to improve the app following modern development best practices (given your platform of choice), but also feel free to use your own app architecture preferences (coding standards, code organization, third-party libraries, etc).

## Deliverables

The project source code and dependencies should be made available in GitHub. Here are the steps you should follow:

1. Download the source code of the project you'll be working on. There are two ZIP files available, both of them for the same project, but one written in Java and the other in Kotlin, feel free to choose either one of them.
2. Create a public repository on GitHub (create an account if you don't have one).
3. Add the project within the ZIP file you chose to the repository you created (Adding an existing project to GitHub).
4. Make sure you have Android Studio 3.0 or newer installed.
5. Create a "development" branch and commit the code to it. Do not push the code to the master branch.
6. Once the work is complete, create a pull request from "development" into "master" and send us the link. We will review your code and might ask you some questions directly on your pull request.

## Notes

Here at ArcTouch we're big believers of collective code ownership, so remember that you're writing code that will be reviewed, tested, maintained by other developers. Things to keep in mind:

- First of all, it should compile and run without errors
- Be as clean and consistent as possible
- You'll be evaluated on code clarity, object orientation, design patterns, platform knowledge, memory management, concurrency, web services/network programming, and unit testing


- **Despite the project simplicity, don't ignore development and architecture best practices. It's expected that code architecture will be structured to support project growth.**

The tasks' descriptions are intentionally vague in some aspects, but if you need assistance feel free to ask for help.

We wish you good luck!