



Flutter Developer - Test/Task Document

Following up on our interview process, we would like to take the next step and test your development skills!

Please Note:

1. Your Test Submission date is: [48 hours from the time you receive the email.](#)
2. We suggest you keep your code clean and up to mark with the best practices.
3. Resources: [Figma Design](#) | [API Documentation](#)
4. Choose a test, Basic or Advanced depending on your skill level
5. If you are unable to complete the entire task, please submit any work you have completed. It is more important to see the work done within the time constraint, compared to completing the assignment.



What you will need to submit:

1. GitHub Repository code link.
2. Demo screen recording for the test
3. APK to install.
4. A demonstration video where you explain your code structure and any approaches you have used to do it.
(if possible, you can use [Loom](#) and share video links).

In case of any questions, feel free to get in touch with us. Looking forward to your submission. Good luck!

Instructions

The test project will be a flutter application that should be created and should also support both landscape and portrait device orientations. The result of your work should be a buildable application able to run on an Android device (with minimum SDK 19 and target SDK 28) & on an iOS device (iOS 10+)

Task

Your code must follow Flutter coding standards
Use of any clean architecture pattern (Change Notifier architecture pattern, BLoC pattern)

- Please choose the difficulty level. ([Basic page](#) or [Advanced page](#))
- Utilise your knowledge of Platform Channels, Streams, Moor/Floor db, Dio/Chopper and other frequently used components.
- Must be highly efficient apps considering the performance
- Must be able to persist data and work offline. (Caching Using Moor/Floor db)



- Feel free to use 3rd party widgets. App should be offline first (Optional)
- Write unit tests (Please refer to flutter [documentation](#))

Assignment

It is an app that lists all upcoming movies using The Movie Db API (TMdb) and then navigates to book movie tickets.

1. **Movie List Screen:** List down all upcoming movies on the list

Movie List API

<https://api.themoviedb.org/3/movie/upcoming>

Params:

api_key - 123456abcdefg (dummy, you can create your one key for free please refer to the [themoviedb](#) documentation).

2. **Movie Detail Screen:** When a user selects any movie from the movie list screen, it navigates to the Movie detail screen. After pressing the "Watch Trailer" button on the movie details screen, the application should display a full-screen movie player and should automatically start the playback (to get the needed URLs to use the "movie/#MOVIE_ID#/videos" API call). After the trailer is finished the player should be automatically closed, and the app should return to the detail page. The playback can be also cancelled by pressing the "Done" button.

Movie Details API

<https://api.themoviedb.org/3/movie/<movie-id>>

Params:

api_key - 123456abcdefg (dummy)

Get Images API

<https://api.themoviedb.org/3/movie/<movie-id>/images>

Params:

api_key - 123456abcdefg (dummy)



3. Movie search screen

4. Seat mapping(UI only)