

RN-Player-APP

Introduction

Context

You are asked to make a player application - which will be used to preview tracks, mostly coming from a Spotify-Like API.

General requirements

- The project should be realised in Javascript (ES6 is highly encouraged)
- The project should be usable with the latest [Node.JS](#) version
- The project uses [NPM](#) for dependencies management
- You shouldn't need to add any external library to the project.
- **if you do use external librairies** - make sure to use all of their functionalities and explain why in [LIBRARIES.md](#) in this project.
- You **can** add developer tooling (tools/libs not built into the final app) if you feel like it like TypeScript, Prettier, etc ...
- You should use the API provided for this project. You can find the documentation in the [API.md file](#)
- You shouldn't need any API key for the exercices - but you may need some for bonus points
- **All bonus points**, if you do bonus points, should be realised in the order given in the subject
- Bonus points will not be taken into account if the subject is not complete

NB: We will have a highly critial look on **how** you make and architecture your project, not on the completion. Make sure to produce work you are proud of.

Getting started

- Install Node.JS if you don't have it - [Download page](#)
- Clone this repository
- Use **npm install** to get dependencies
- Use **npm start** to start the project
- Install **expo** application on your smartphone and scan the QR code displayed on your computer
- **Make sure** that your smartphone is connected to the same network as your computer
- Open this folder with your favorite editor - we recommend [Visual Studio Code](#) if you don't have one yet.
- You are ready to work !

Links

- [React Native website & documentation](#)
- [React Navigation website & documentation](#)
- [Expo website & documentation](#)

Tasks

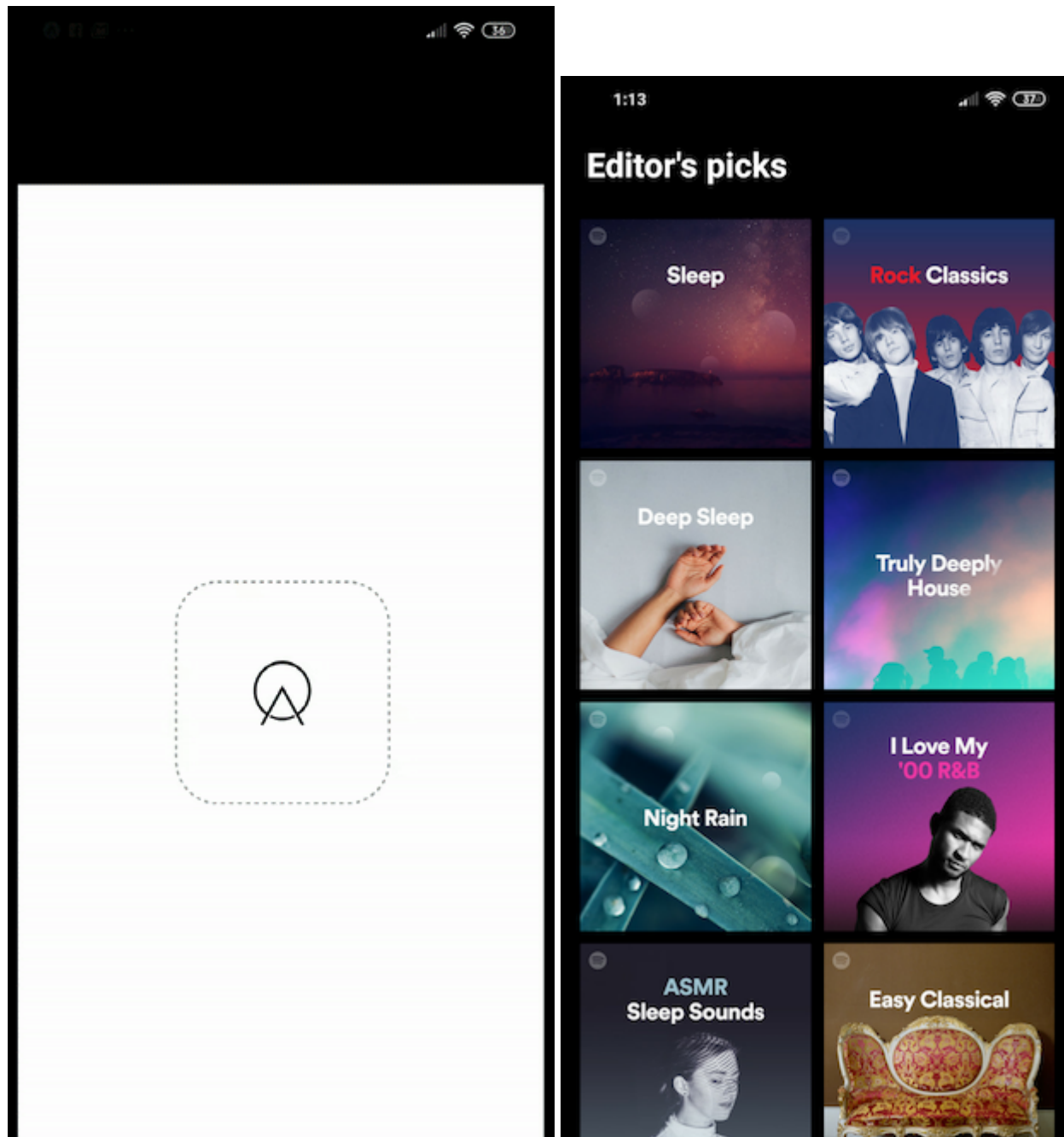
TASK-001

Create a playlist view. This view should be scrollable to display all featured playlists.

Data API: </v1/browse/featured-playlists>

Expected result

You can find these files in `./__subject__images__/task_001.gif` & `./__subject__images__/task_001s.png`



TASK-002

When the user clicks on a playlist from the previous screen - he should be redirected to a page where informations and tracks about the playlist are displayed.

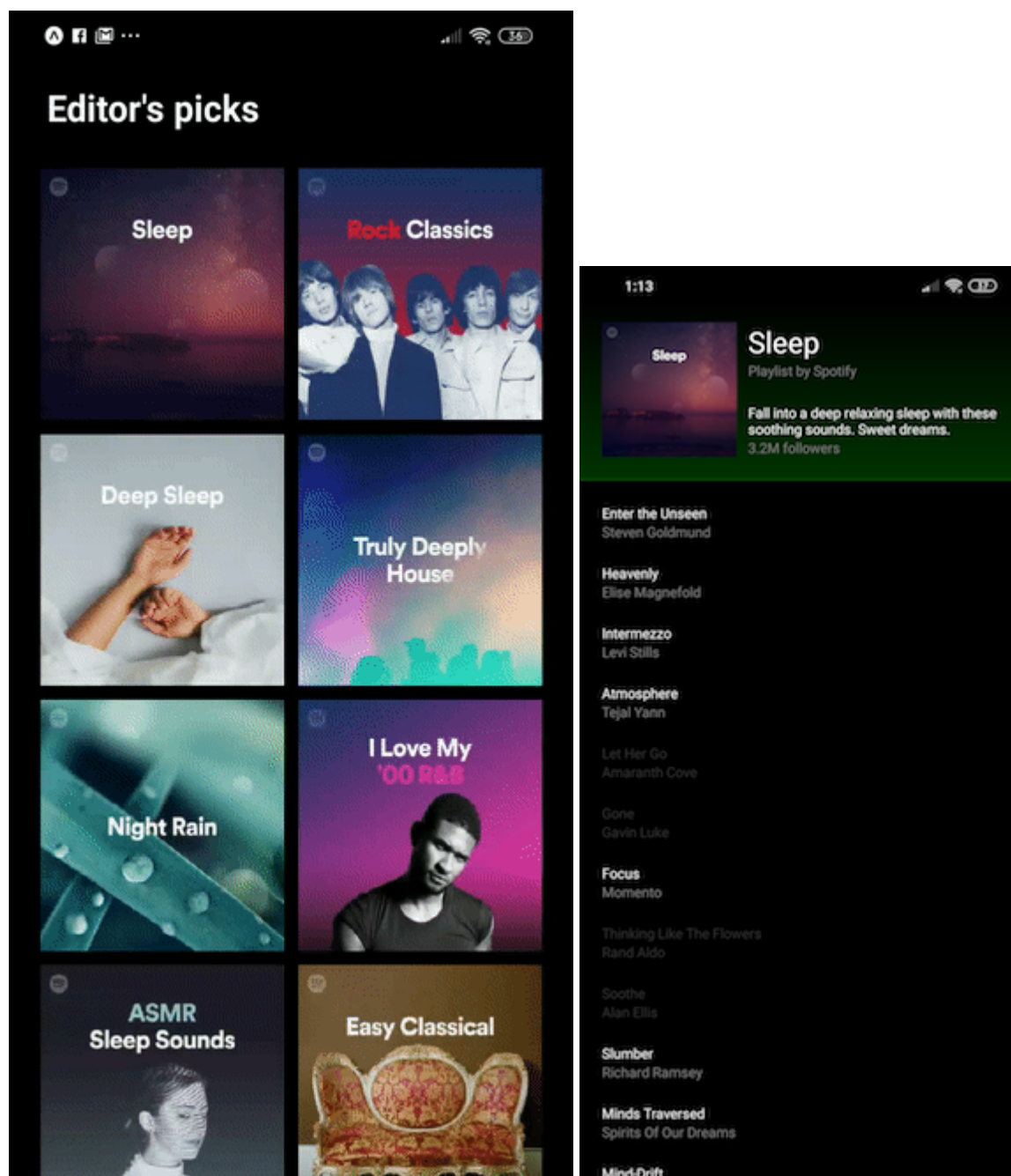
This view should be scrollable to display all tracks available on the playlist.

If a track doesn't have a `preview_url` in the JSON - the style should differ.

Data API: [/v1/playlists/\[PLAYLIST_ID\]](/v1/playlists/[PLAYLIST_ID])

Expected result

You can find these files in `./__subject__images__/task_002.gif` & `./__subject__images__/task_002.png`



TASK-003

When the user clicks on a track - the song should play and a player should be displayed at the bottom of the page.

Also, if the user clicks on the pause button - the track should pause and the icon should change.

If the user clicks on the play button - the track should resume.

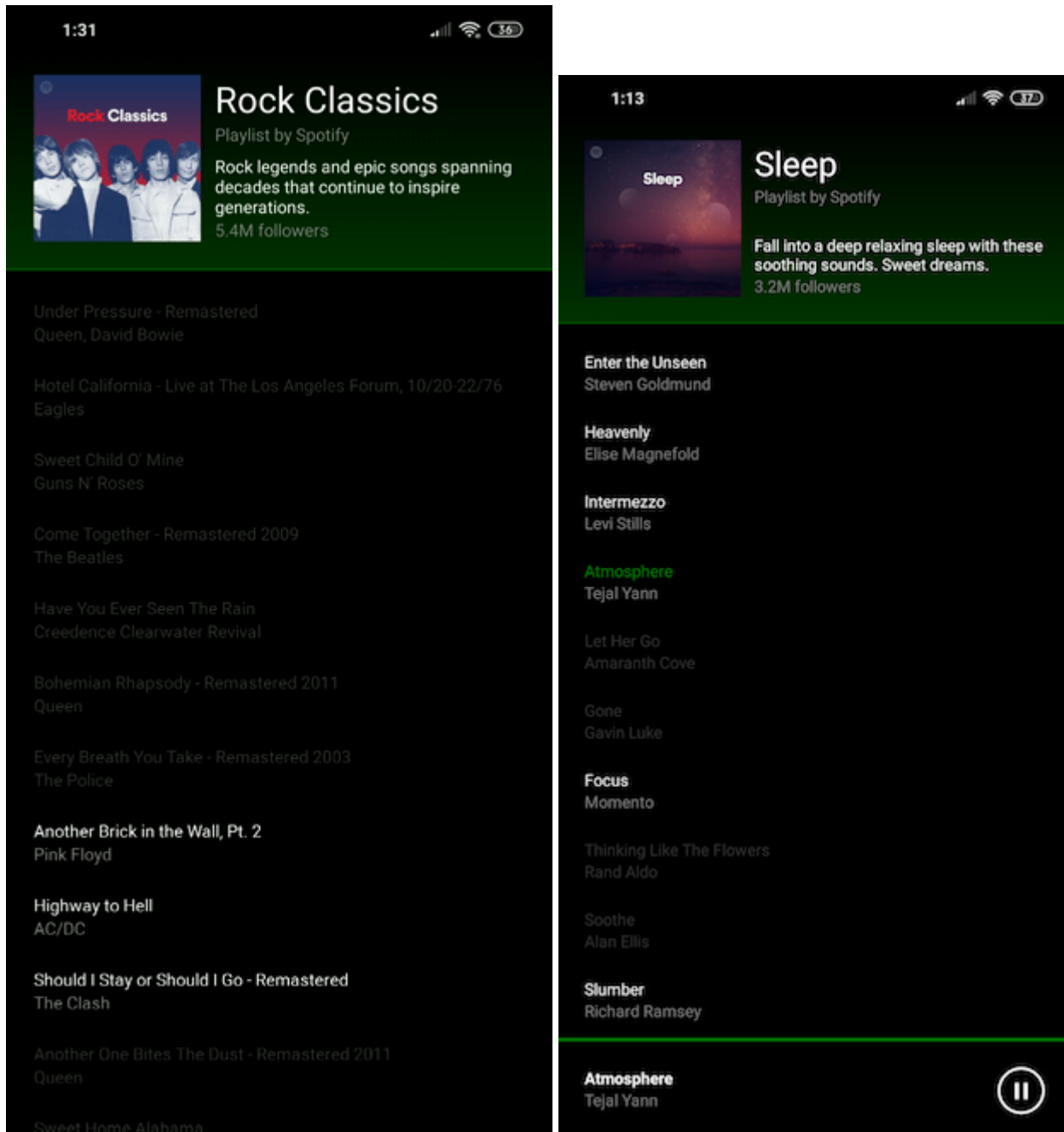
If the user clicks on another track, the new clicked track should play.

If a track doesn't have a `preview_url` in the JSON - the track should not be clickable and clicking on it shouldn't do anything.

Data API: You don't need an API for this. You should find the preview URL in the tracks JSON you used in the previous task. It can be found in `playlist.tracks.items[*].track.preview_url` and should look like something like this: <https://p.scdn.co/mp3-preview/5d596bfe62a2fb24b77a32b1013520b261efaf7f?cid=67dd1a878f8d4deaa5c4400c6619021b>

Expected result

You can find these files in `./__subject__images__/task_003.gif` & `./__subject__images__/task_003.png`



TASK-004

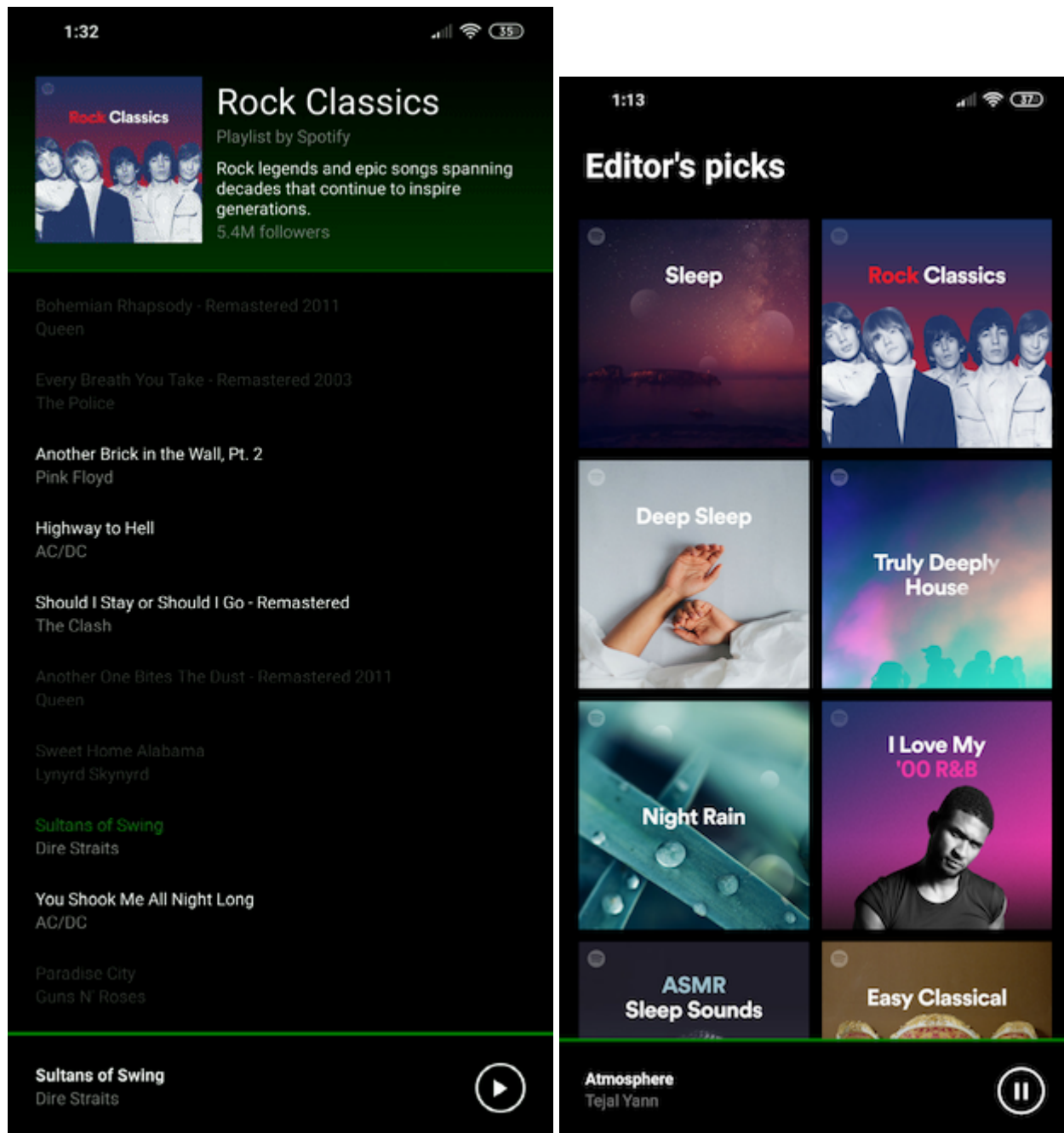
The player should be available on the previous screen as well. The user should be able to control the music from the playlist screen as well.

It should be able to change screen without any music interruption. He can also select another track from another playlist and it should work just like in the previous task.

Data API: You don't need an API for this.

Expected result

You can find these files in `./__subject__images__/task_004.gif` & `./__subject__images__/task_004.png`



Bonus points

Animate the border of the player view

Right now, the player view at the bottom has a solid green border on top. This border could represent the duration of the song and should be filled from left to right according to the song remaining time.

Create a player screen

When the user clicks on the player view at the bottom of the app, we could display a modal with a fully-fledged player just like the real Spotify app.

Handle playing a full playlist

Add a button "Play all" at the top of the track list to make the app play all the tracks in the playlist

Go crazy

If you're here - Congratulations ! You can add whatever you feel like adding to the app - make sure to specify what you add in the mail you will send with your work.

Some ideas:

- Add animations wherever you can
- Add pages for Artists / Search / Parameters ?
- Offline mode
- Add the TabBar like the real spotify
- Connect it to the real Spotify - with a real login screen for the user - and display tailored playlist recommendations for him.
- Chromecast connection to play the music on TV
- Improve design wherever you can
- ... basically, any Spotify feature you feel like adding 🤔