

Bloom: Frontend Coding Exercise

Introduction

We recommend you spend around 2-3 hours on this exercise. You are required to build a Mobile app in React native that allows a user to login with a PIN, and then to logout. An API has been provided to you to make login and logout requests.

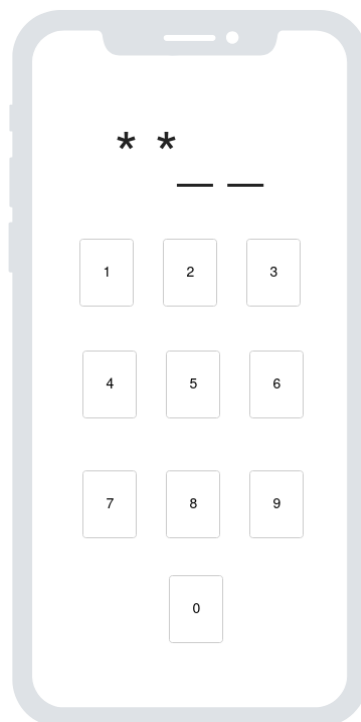
We will be assessing your code on correctness, covering of the appropriate states (e.g. loading, fetch errors, errors cases returned by API), and code quality. Ensure your submission has the appropriate test coverage and follows what you consider to be best practices for producing clean code.

You are allowed to use whatever libraries, tools and frameworks you want. The only requirement in this regard is you use React Native.

PIN Reset API

You have been provided with a node service that serves a REST API that allows you to login with a pin, and to logout. The task will involve building a React Native App around that API. You can find this in the folder “service” along with a README with instructions on how to run it. See the appendix for information on how each endpoint works.

Task 1: Pin Protected Entry



Above: A mockup of the login screen you should build

Build a Pin entry component and login screen that meets the following requirements:

- A Pin code entry area displaying buttons labeled 0-9.
- The user should be able to press these buttons and the pin progress should be displayed on screen as they input it.
- All pin codes are 4 numbers long, so on entering the fourth number the app should immediately attempt to verify the user by calling the /login endpoint (See appendix).
- On Success the user should be taken to the screen in Task 2. A token will be returned from the login endpoint which will be required to carry out authenticated requests.

Note : The correct PIN is printed to the terminal when you start the server. Please see the server README for more details.

Task 2: Authorised screen with Logout

- After login, the user should be taken to a screen with one button that says “Logout”.
- On pressing logout the user should be “logged out” by calling the /logout endpoint (See appendix) and taken back to the screen in Task 1.

Submission

You can send us your submission in a zip file, a git bundle file, or via a git repo link. Please ensure you provide a simple README with instructions on how to run your submission.

Please use the appropriate package management tooling (e.g. yarn or npm) so we can easily build and run your work.

Appendix: The PIN API

Login

Url: /login

Method: POST

Expected Body:

```
{
  pin: String
}
```

The pin should be four digits long, but provided as a string not a number.

Possible Responses

200 - Login successful

```
{
```

```
token: String
}
```

The token you get back can later be used in the Authorization header when making authenticated requests.

403 - Login Unsuccessful

```
{
  error: "INCORRECT_PIN"
}
```

400 - Invalid Request (pin not a string or not 4 chars long)

```
{
  error: "INVALID_REQUEST"
}
```

Logout

Url: /logout

Method: POST

Expected Headers:

Authorization: [TOKEN]

In order to authenticate, just provide the token returned from the login endpoint in this header. No need to prefix with something like Token or bearer.

Expect body: Empty

Possible responses

200 - Logged out successfully

```
{
  "message": "You have been logged out, the token is no longer valid."
}
```

401 - Credentials missing or bad credentials

```
{
  error: TOKEN_MISSING | BAD_TOKEN
}
```