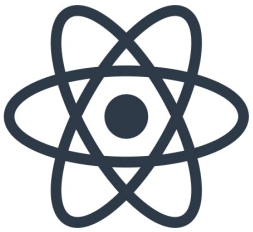
M3 - Mobile

M-MOB-300

Mobile

Create a StopWatch



2.0

{EPITECH.}

Mobile

language: JavaScript / TypeScript

- Your repository must contain the totality of your source files, but no useless files (binary, temp files, obj files,...).
- All the bonus files (including a potential specific Makefile) should be in a directory named *bonus*.
- Error messages have to be written on the error output, and the program should then exit with the 84 error code (0 if there is no error).

Today we will see how to create a StopWatch App. For the workshop, we will cover Expo (React-Native) and Ionic.

SET UP THE WORKSPACE

+ EXPO (REACT-NATIVE)

You will need to create a new Expo app.

Terminal

```
~/M-MOB-300> expo init my-new-project
~/M-MOB-300> cd my-new-project
~/M-MOB-300> expo start
```

You can now open the workspace with your favorite IDE (Vscode, Atom, vi...)

+ IONIC

You will need to create a new Ionic app.

```
Terminal
~/M-MOB-300> ionic start my-new-project blank -type angular
~/M-MOB-300> cd my-new-project
~/M-MOB-300> ionic serve
```

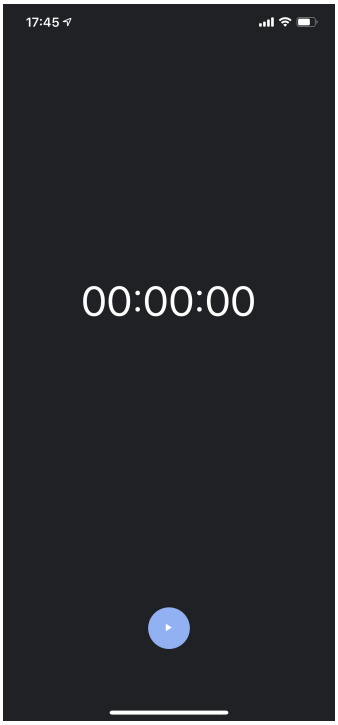
You can now open the workspace with your favorite IDE (Vscode, Atom, vi…)

OBJECTIVE

We will now see the different steps to creating a mobile App.

+ DESIGN

Before coding the app, you will first have to design how the app will look and feel. We call this designing the UI/UX of the app. In our case, we don't have the time to do this; we will base our app on the default clock app on your phone.



+ CODING THE UI

Once designing UI/UX is done, you will need to implement it.

REACT

Your `App.js` file (at the root of the project) should be similar to the image below.

```
import React, { Component } from 'react';
import { SafeAreaView } from 'react-native';

export default class App extends Component {
    state = { }

    render() {
        return (
            <SafeAreaView>
            {/** UI Code HERE */}
            </SafeAreaView>
        )
    }
}
```

You may need to add the `SafeAreaView`; this component makes sure to render within the safe area boundaries of a device; as of now, this feature only works on iOS.

Documentation for UI Components you can use in Expo.

IONIC

Create a new Ionic page.

```
Terminal
~/M-MOB-300> ionic g page
~/M-MOB-300> cd my-new-project
~/M-MOB-300> ionic serve
```

You can now edit the HTML file.

Documentation for UI Components in Ionic.

You may now code the UI; you have a Text with the current elapsed time, a start, a stop, and a restart button.

UX: Think how the buttons should work one button for each action, and button that changes state?

+ Code the App's Logic

Once you finished the UI, you may now code how the app works. In our case, you need to be able to start, stop, and then restart the timer.
The intricate part is displaying the elapsed time, having the perfect balance between being precise, and consuming too many resources.

Furthermore

Once you are done, call us to show us the final product. You may continue the project by, for example, adding a timer?