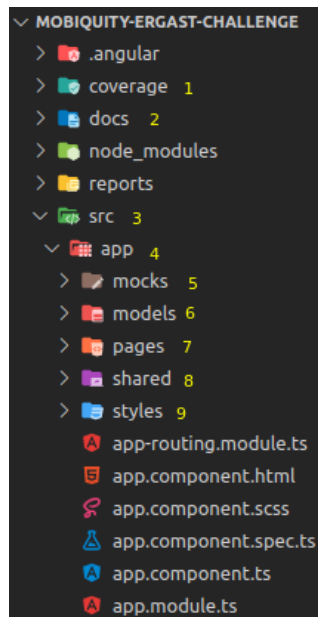


# Mobiquity – Ergast Frontend Challenge

This document is divided in 3 sections:

- Section 1 aims to explain the project structure
- Section 2 shows the application components
- Section 3 shows the application ErgastService
- Assumptions

## Section 1 – Project structure



### 1.1 – coverage

In this folder you can find the report (index.html) created each time the unit tests run (you can generate it in your terminal, running the following command: `ng test`).

#### All files

100% Statements 123/123 100% Branches 12/12 100% Functions 47/47 100% Lines 118/118

Press *n* or *j* to go to the next uncovered block, *b*, *p* or *k* for the previous block.

Filter:

File	Statements
app	100%
app/mocks	100%
app/pages/main	100%
app/pages/race-winners	100%
app/shared/components/header	100%
app/shared/components/list	100%
app/shared/components/spinner	100%
app/shared/constants	100%
app/shared/services	100%

## Mobiquity – Ergast Frontend Challenge

### 1.2 – docs

All relevant documentation for this project must be saved here.

### 1.3 – src

This is the source for the entry point for the SPA.

Assets and environment configurations must be placed inside this folder. You will also find the global styles.scss here.

### 1.4 – app

This is the main component of this application. The routing configuration is available here.

#### 1.4.5 – mocks

All necessary mocks for development or running unit tests must be here.

#### 1.4.6 – models

Here you will find the entities contract representation.

#### 1.4.7 – pages

In this folder, we are storing all page-wrapper components. For each application page we must have a component representing this page here.

#### 1.4.8 – shared

Components, services, etc., that might be used in more than one place, *probably* should be here.

#### 1.4.9 – styles

Shared styling, such as color palette.

## Section 2 – Application Components

The current application components are:

- 1 - main (page)
- 2 - race-winners (page)
- 3 - header (shared)
- 4 - list (shared)
- 5 - spinner (shared)

## Mobiquity – Ergast Frontend Challenge

### 2.1 – main

This component creates a main page for the application. This will be the component rendered when the application loads the first time (with the root url).

It displays a list of years from 2005 until now, and when we click on any of these, it sends us to another route (/race-winners/yearNumberHere).

*Main Page*



### 2.2 – race-winners

In this component, we have a list with all race and its winners for a given year. It is currently displaying the World Champion highlighted when available.



## **Mobiquity – Ergast Frontend Challenge**

### **2.3 - header**

The header component is a shared component used mainly to display a title and provide a button to navigate back to the previous page.

Other components can send urls to the goBack stack or a title to be displayed by using the header.service.

### **2.4 – list**

This component works as a High Order Component, which receives an list of items and the <ng-template> item with the UI implementation of the item list.

### **2.5 - spinner (shared)**

A simple html/css spinner to be used when loading / rendering / fetching data.

## **Section 3 – ErgastService**

The ErgastService is the implementation for the F1Service (src/app/shared/services/interfaces). It creates the concrete resolution for the following operations:

- + getSeasonsBetweenYearInterval
- + getSeasonsFromYearUntilNow
- + getWorldChampionByYear
- + getAllRacesWinnersOfAYear

## **Section 4 – Assumptions**

To code this application according to (frontend-challenge-description.pdf), the following were taken as true:

- The first year for a F1 seasons is the same as it is on the database, 1950 and this will never change
- The data returned is always available, therefore I am deep accessing object without throwing exceptions for null-ish fields.
- For the current year, some races will still be pending, therefore they will have no winners. Also, the world champion of a given year will be available only after the last day of the last race.