This project was a part of a job application for Reaktor summer developer 2022 position

The goal was to develop a simple web app that tests the basics of the web stack including requests from an API, UI/frontend and a database/server for storing game results: “Your task is to implement a web application that displays rock-paper-scissors match results.”

I didn’t have that much previous experience with web apps, so I just had to choose one framework. Angular framework is quite popular and actively maintained and most importantly had good learning material and documentation online. I used the excellent “Angular Tour of Heroes” tutorial as a starting point for my project.

Design

I tried to stick with “Angular coding style guide”. It defines some basics like naming schemes, the distinction between components and services and project structure. My project does not use any external libraries that don’t come with a default angular project.

When doing HTTP requests, /rps/history API had a missing CORS ‘access-control-allow-origin’ header. The optimal solution would be that the API and website were hosted under the same domain. It was not possible, so I deployed my own HTTP proxy that added the missing header. It was done with the help of a Stack Overflow post and a GitHub repository. Just adding the header was not the optimal solution because CORS is effective in preventing cross site scripting attacks for example.

The biggest problem

I haven yet learned to use databases. Relational databases were mentioned in the job description for a good reason. When I started this project I though that I could somehow do without a backend server but that was obviously not true as the match history API /rps/history probably contains thousands of pages. I made a poor implementation of a single player’s match history and aggregate data that goes through four pages of history like a linked list.

The good news is that I will start a course on databases next month so I will have studied the basics by summer.

Sources:

“Angular Tour of Heroes” tutorial (https://angular.io/tutorial)

“Angular coding style guide” (https://angular.io/guide/styleguide)

Stack Overflow post javascript - No 'Access-Control-Allow-Origin' header is present on the requested resource—when trying to get data from a REST API - Stack Overflow

GitHub repo for the HTTP proxy code https://github.com/Rob--W/cors-anywhere.git

Exercise description (https://www.reaktor.com/assignment-2022-developers/):

Your task is to implement a web application that displays rock-paper-scissors match results. The web application must display the games as they are in progress, and you must also be able to look at historical results of individual players. In addition, the currently ongoing games must be visible at all times.

The historical results must include all games that a player has played and it must also include the following aggregate data: win ratio, total number of matches played, and the most played hand (rock, paper, or scissors).

There are two API endpoints running at https://bad-api-assignment.reaktor.com. /rps/history which returns a page of historical data, and a path for the next page of data. Be aware that there are many pages of data. The other endpoint /rps/live is a WebSocket API and will send events to connected clients as they happen.

Your application does not need to look especially pretty (but it won't hurt) but it should be fast and snappy, showing data to the end user as fast as possible and being up-to-date with the backend.

In no particular order, we especially pay attention to the following details when looking at your submission:

- Readability of the code as whole
- Performance and maintainability
- Technology and library choices made
- User interface and experience

Back to the job post