Solution Description

1. Goal of the Application

The purpose of the application is to display the content of the remote shared folder, each n seconds. If a change is made (delete, rename, create a file) the changes are displayed on the web page.

2. User Manual

2.1. Running the Server side

- Open the project Pradigms_HMWK4
- go to newdemo file using cd newdemo on terminal
- to run the server use: gradle run

2.2. Running the client side

- Open the below file on the browser:

    Paradigms-HMWK4\newdemo\app\src\main\JavaScript\index.html

- Enter on the text field an integer representing the number of seconds.

-click on start button to start the application

- click on stop button to stop the application

3. Development Process

Server Side:

The server side is developed using the Rest Api, using many annotations like @CrossOrigin in order to avoid CORS issues and enable cross-origin resource sharing. The annotation @RestController is used in order to mark the class as a handle requests. Annotations like @RequestMapping @GetMapping are used to specify the links that the client side will refer to in order to access the services.

Service consumer:

The service side is browser-based using HTML, css, and javascript. In the javascript file, we are referring to Rxjs and RxHR libraries in order to use observables. In order to fullfill the goal of the application, we need to create two observables the first one will be using RX.interval() to generate an integer each n seconds when subscribed. The second observable will return the files of the shared folder when subscribed. I made use of switchMap so that whenever the first/outer observable returns a integer the second/inner observable will subscribe again. The unsubscribe method will be used to stop the process. The browserify allows us to run Node.js statements in the browser.