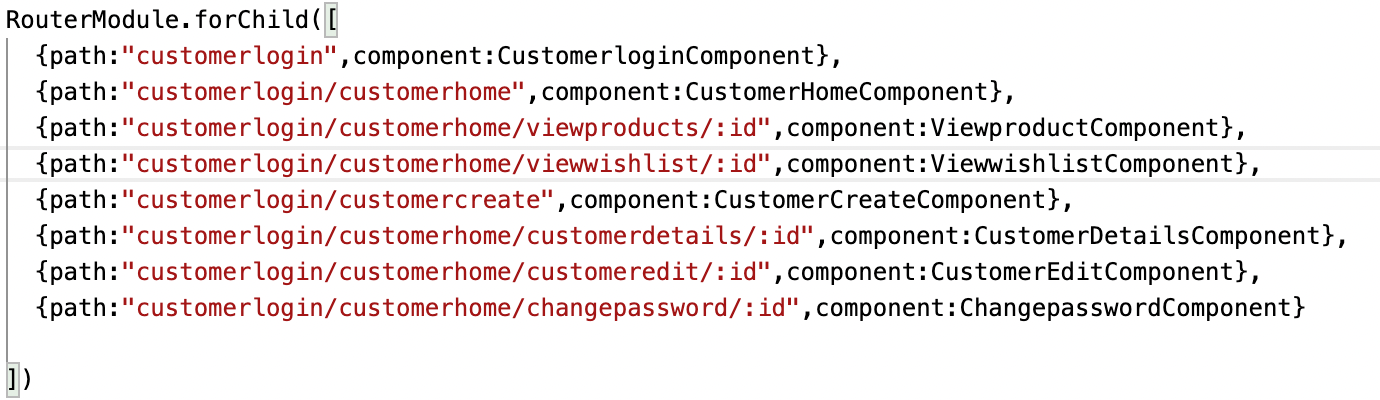
This is the documentation for the Angular CRUD application, which right now allows Admins to LogIn, view other admin details, delete admins, create new admins, and products. There are also customers who can also log in, and view, edit, or delete their details/accounts. Customers also have the option to view products, and change their passwords right now.

This application has been made using Angular. The structure of this project currently contains three modules: the default app module, the admin module, and the customer module. Within these modules, there are related components, and services attached to each module.

**Routing**

The routing in this project is handled using RouterModule where the path is defined for each component in the module. For example, these are the paths defined for the Customer Module through RouterModule.



Defining these paths helps us navigate through pages when we click through buttons in our apps. When we keep a path in the routerLink directive on any button, it matches it with the path defined in the RouterModule, and navigates to the component defined for that path.

**Creating Admins**

****

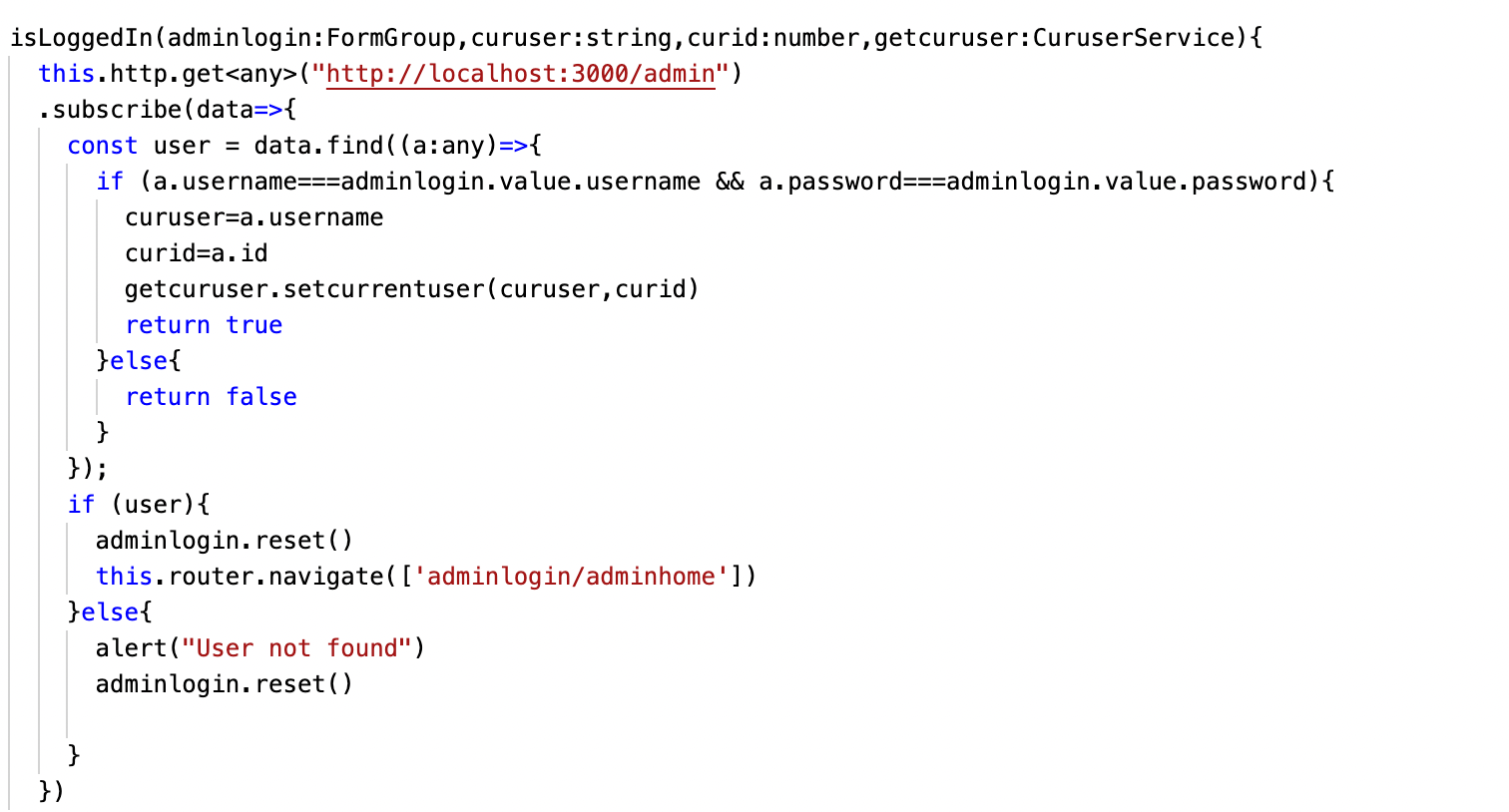
To create an admin, we have an addAdmins form set up in the createAdmins component. This code lives in the adminservice file where the component calls the service to create the admin on submit. Here two things are being passed in: the actual form which is a formgroup type(addAdmins), and the values received from the form on submit(addAdmins).

In this code, we are doing a HTTP get request first to get a list of all admins. We then search through that list of admins to see if the admin username given in the form matches one of the username that already lives in the JSON server. This would mean that the username should not be allowed since two different users cannot have the same username. The code has a boolean value that gets assigned to “user” based on if the username given in the form matches any username in the JSON server.

We can use the boolean value assigned in “user” to do further coding. If the user is True that means the username cannot be used so we can give an appropriate message. If the username is False then the username given in the form does not already exist in the JSON server, so we can use that username, and then we can do a POST request to create a user with that username, and then POST it into the JSON server.

The process is the same when creating customers.

**LoggingIn(for both admins, and customers)**

****

The login procedure for both customers, and admins is the same. We pass in the Form of FormGroup type to the isLoggedIn function, and curid/curuser to keep track of the current user.

We do a get request to get a list of admins from the server, and try to find the admin with the given username or password in the form. If the usernames, and passwords match, we can proceed to the AdminHome component for that user where most of the Admin functionality lives in for the admin. Here, I am using a boolean called “user” again, which moves to the adminHome page if there exists a user with existing correct credentials. In this case I just use the router service to navigate to the given url. If the credentials do not match, I print out a visual feedback for the user that the login was not successful.

**List of Functionality on the CRUD app**

* Admin Login(with form validation and authentication)

-Once admin is logged in admin can:

-Add new admin

-View Customers

-Add new product

-View the details of any admin

-Edit any admin

-Delete any admin

-Signout

-Customer Login(with form validation and authentication)

-Once customer is logged in customer can:

-View their own details

-Edit their profile

-Delete their profile

-View all products, and add products they like to their wishlist

-View wishlist, and delete products from the wishlist

-Change their password

-Signout