**Angular Stack Overflow App - Frontend Documentation**

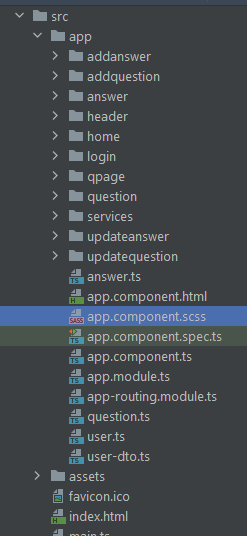
**Introduction**

This documentation provides an overview of the frontend implementation of the Angular Stack Overflow app. The app aims to provide a platform for users to ask questions, view questions and answers, and interact with the Stack Overflow community. The last 2 assignments will be merged into a single one because I did the frontend and the connection with the backend at the same time

**Folder Structure**

The folder structure of the Angular Stack Overflow app is as follows:

Severall packages containing components and services:



**Components**

The Angular Stack Overflow app consists of several components that handle different parts of the user interface. Here are some of the key components:

1. **HomeComponent**: Renders the home page with a list of questions.
2. **QuestionComponent**: Displays a single question with its details and answers.
3. **AnswerComponent**: Shows a single answer with its details.
4. **LoginComponent**: Provides the login functionality for users.
5. **QpageComponent**: integrated in home to show the list of questions
6. **banUserComponent**: for banning
7. **CRUD ANSWERS/ QUESTIONS**

**Services**

Services in the Angular Stack Overflow app are responsible for fetching data from the backend, handling user authentication, and performing other business logic. Key services include:

1. **QuestionService**: Handles operations related to questions, such as fetching a list of questions, retrieving a specific question, and posting a new question.
2. **AnswerService**: Manages operations related to answers, such as fetching answers for a specific question and posting a new answer.
3. **AuthService**: Provides methods for user authentication, including login and registration.
4. VoteService: for calculating the votes

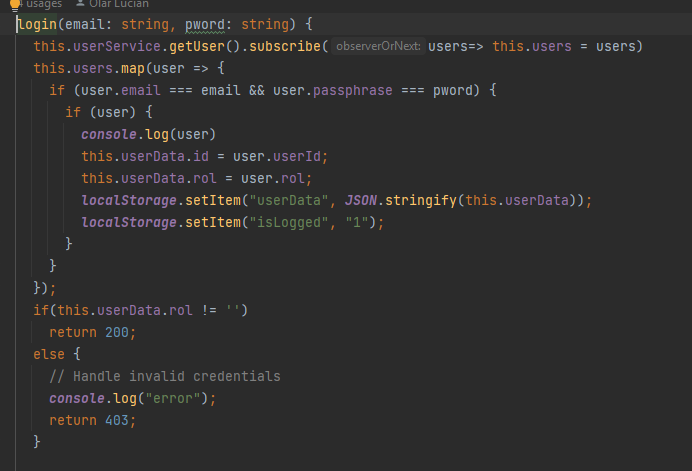
**Routing**

The app uses the Angular Router to handle navigation between different pages. The routing configuration is defined in the **AppRoutingModule** module. Key routes include:

* {path: '', redirectTo: 'login', pathMatch: 'full'},  
  {path: 'login', component: LoginComponent},  
  {path: 'home', component: HomeComponent},  
  {path: 'qpage/:id/:iduser', component: QpageComponent},  
  {path: 'addanswer/:idq', component: AddanswerComponent},  
  {path: 'addquestion', component: AddquestionComponent},  
  {path: 'updatequestion/:idq', component: UpdatequestionComponent},  
  {path: 'updateanswer/:ida', component: UpdateanswerComponent},  
  {path: '\*\*', component: LoginComponent}

**User Authentication**

The Angular Stack Overflow app supports user authentication using JSON Web Tokens (JWT). When a user registers or logs in, the backend API issues a JWT that is stored in the browser's local storage. The JWT is sent with subsequent requests to authenticate the user.



**Home Page**

The home page displays a list of questions fetched from the backend. Each question item includes the question title, summary, and the number of answers. Users can click on a question to navigate to the question page.

**Question Page**

The question page shows the details of a specific question, including the question title, body, and a list of answers. Users can view and interact with the answers, including upvoting, downvoting, and posting new answers.

**Answer Page**

The answer page displays the details of a specific answer, including the answer body, author, and any comments associated with the answer. Users can also upvote or downvote the answer.

**Error Handling**

The app includes error handling mechanisms to provide feedback to the user in case of any errors or failures. Errors are displayed using toast notifications or modal dialogs, informing the user about the issue and suggesting potential solutions.

**Testing**

The Angular Stack Overflow app includes unit tests and end-to-end tests to ensure the correctness and reliability of the application. The testing framework used is Jasmine, and the testing utilities provided by Angular are utilized for writing tests.

**Deployment**

To deploy the Angular Stack Overflow app, follow these steps:

1. Build the production-ready app using **ng build --prod**.
2. Copy the generated **dist** folder to your web server.
3. Configure the server to serve the app and handle routing correctly.
4. Set up the backend API and configure the app to communicate with it.

**Conclusion**

This documentation has provided an overview of the frontend implementation of the Angular Stack Overflow app. It covered installation instructions, folder structure, key components and services, user authentication, routing, and various pages of the app. It also mentioned error handling, testing, deployment steps, and suggested future improvements.

For further details and code examples, please refer to the source code of the app.