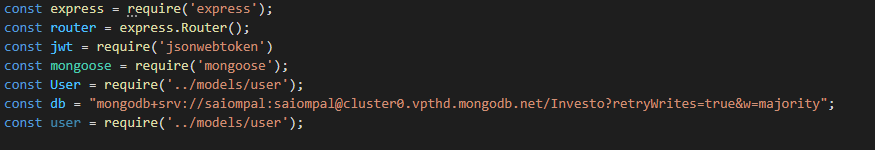
**SAIOM PAL NODEJS ASSIGNMENT**

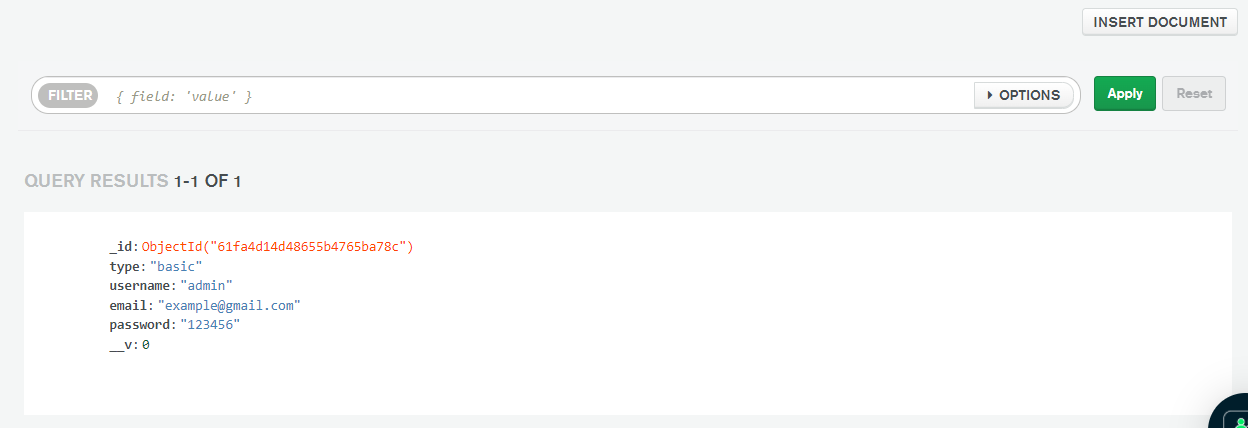
**STEP BY STEP PROCEDURE ON IMPLEMENTING JWT TOKENS FOR AUTHORIZATION**

**PRE-REQUISITES SETUPS:**

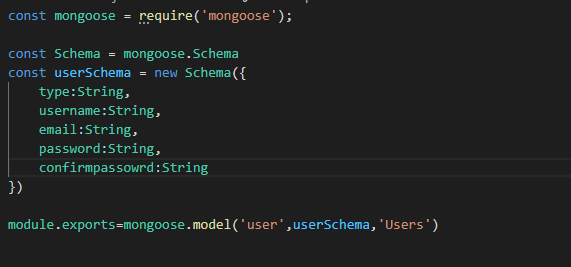
1. Create a nodejs server with express and import router, cors, body-parser, jsonwebtoken and mongoose.



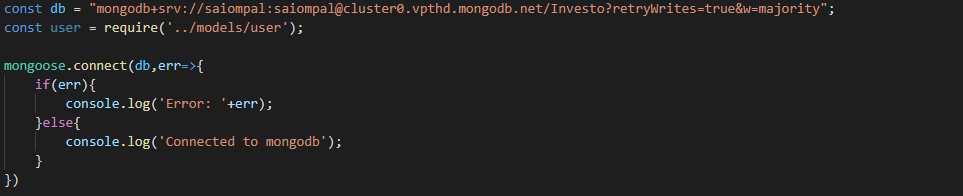
1. Create a database in mongodb having a Users collection.



1. Define the schema of the collection in node.



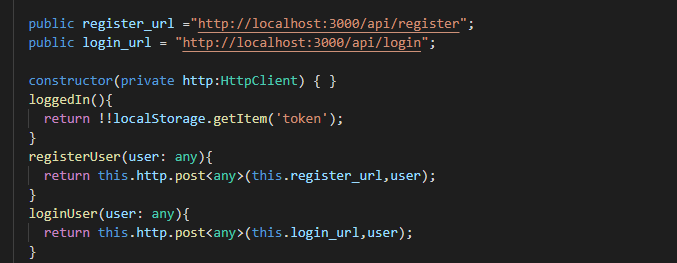
1. Connect the database with node.



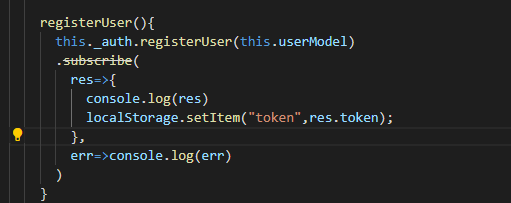
1. Create api endpoints for login, registration and dashboard.



1. Set up an angular project with login and registration components and configure their routes in app-routing module.
2. Create auth service that will send post requests to the api endpoints during login and registration and get request to the end points for dashboard.

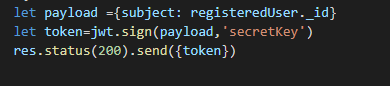


1. Now, subscribe to the services.



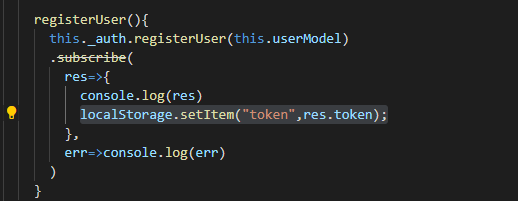
**GENERATING TOKENS:**

1. Define the payload
2. Generate the token using jwt.sign() method with the id of the user data and a secret key.
3. Send the token object as a response whenever the login/register request is received.



**STORING THE TOKENS IN LOCAL STORAGE:**

1. We store the token in the Local storage while subscribing to the login and registration service.

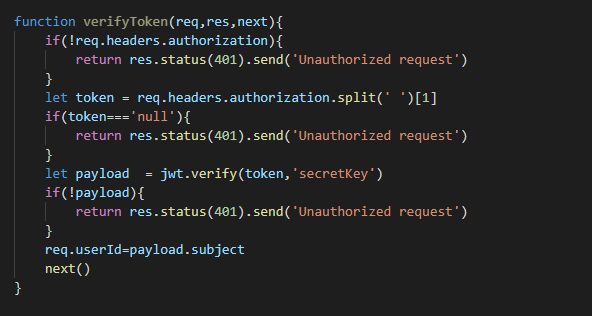


**INTERCEPTING AND VERIFYING THE TOKEN:**

1. We create a token-interceptor service that implements the HttpInterceptor module.
2. Now, we clone the request and set the header: (authorization, intercepted token) and call the next method.



1. Now, on the server side we create a verifytoken() method that parses the header for the token and verify it using the secret key which was used while generating the token.
2. If, the token is valid then we call the next method.



1. We apply the verifytoken() method on the routes we want to protect from un-authorized users.

