**Account Creation**

1. Capture the username and password in react app in registration page
2. Send the credentials to the registration endpoint in the backend
3. After validation, the credential stores in DB.
4. Now we can generate JWT Token using JWT or other library
5. The token is saved on the server and returned in the response of the initial register request.
6. A refresh token can also be returned here.

**Token Handling:**

1. The registration request should return a auth token in case of successful registration.
2. This token should be stored in the client app(react), in context or redux.
3. If remember me then the stored token should be persisted at client side.
4. Now the user is logged in, from now you will send this token in the header of all requests to your backend.
5. Then your backend can look up the received token with the one stored in your DB and decide whether the current request is allowed.

**Token refresh**

1. **For security reason, we may want to set TTL for all token issued.**
2. **When a request using an expired token come to backend, then return an expired response what should be processed on client side.**
3. **The client should make a refresh token call by using locally stored refresh token.**
4. **This should generate a new auth token on the server an replace the old token with freshly generated one.**
5. **we can set the TTL to "keep forever" in which case you will not need to deal with token refresh.**

**Login**

1. **Your React App captures the username/password and send them to backend as the same way in registration.**
2. **The backend validates the account details and generates and return a new token.**
3. **The client should use this token as described at registration**