- Duplicate last topic's project, and continue here
- Your goal is to implement JWT token mechanism
- End result:
  - If you `POST` to `/token` with your regular username and password, you should get a JWT token back. The token may look similar to this: `eyJhbGciOiJSUzI1NiJ9...`
  - Try calling any of the endpoints you have in your app using the newly generated JWT token. Don't forget to set the `Authorization` header to `Bearer Token`. If the token works correctly, you should be able to use all of the endpoints you have defined before, just as if you were using `Basic Auth`

If username and password works just as well for authentication, why use JWT tokens at all?

- **Basic Auth sends credentials in every request**, while **JWT only sends the token** (credentials are used only once during login).
- **Basic Auth requires checking credentials against the database for every request**, slowing things down.
- **JWT supports token expiration and refresh tokens**, allowing smooth re-authentication without re-entering credentials.
- **Basic Auth is not ideal for APIs** because credentials must be sent repeatedly, increasing exposure risk.