

This week, we have to complete User APIs in the back-end and build up a sign up/in/out user interface in the front-end.

For separation of the front-end and back-end architecture, never modify any code marked as # Static Pages.

Part 4-1: User APIs

Follow the API specifications, complete 4 User APIs.

Follow steps below to use JWT + Bearer Token mechanism to build member system:

1. Suggest using PyJWT: <https://pyjwt.readthedocs.io/en/latest/index.html>
2. When a user tries to sign in, our back-end code checks if the pair of email and password exists in the database. If yes, encode user id, name, email, and other critical information by JWT encoding mechanism and send a signed TOKEN back to the front-end.
3. After successful signing in, the front-end will get signed TOKEN from the back-end. Save it into LocalStorage of the user's browser.
4. When a signed-in user tries to call APIs which require authorization, our front-end code should send a request with Bearer Token in the Authorization header.
5. Once our back-end code receives a request from the front-end, get TOKEN from the Authorization header, and then, decode and verify received TOKEN by JWT decoding mechanism. If failed to decode or verify it, it means it's an unauthorized request which should be rejected. Else, our back-end code can get critical user information which is saved in the previous signing-in procedure.

Part 4-2: Pop-Up Dialog for User Sign Up/In

In every page of our website, there is「登入 / 註冊」text on the top-right corner. When the user clicks on it, show a pop-up dialog based on the design in Figma, for user sign up/in.

1. At the first shot, show a form for user sign in, if the input pair of email and password is incorrect, show the error message in the bottom of dialog.
2. User can click「點此註冊」text to change to a form for user sign up. If the input email is repeated, show the error message in the bottom of the dialog.

Part 4-3: User Sign-In Status Checking Procedure

Right after every page load, our front-end code should check user sign-in status for later use. Every page should implement procedure described below:

1. Right after page load, fetch User API to get current signed-in user information.
2. If the API response is null, it means the current user does not sign in. This time, we should render「登入 / 註冊」text on the top-right corner which can be clicked to show user sign up/in dialog.
3. If the API response is not null, it means the current user has signed in. This time, we should render「登出系統」text on the top-right corner which can be clicked to sign out current user.

Part 4-4: Sign Up Procedure

If a user tries to sign up for a new account, follow steps described below:

1. User should enter name, email, and password in sign up form, our front-end code fetch User API for signing up.
2. If successful, show the successful message in the bottom of the dialog.
3. If failed, show the error message in the bottom of the dialog.

Part 4-5: Sign In Procedure

If a user tries to sign in, follow steps described below:

1. User should enter email, and password in sign in form, our front-end code fetch User API for signing in.
2. If successful, refresh the current page for the latest sign-in status.
3. If failed, show the error message in the bottom of the dialog.

Part 4-6: Sign Out Procedure

If a user tries to sign out, follow steps described below:

1. Only after user has signed in, he/she can click「登出系統」text for signing out. Our front-end code removes saved TOKEN in the LocalStorage of the user's browser.
2. After removing TOKEN, refresh the current page for the latest sign-in status.

How to Submit Task:

Following the Git, GitHub workflow we have done in Part 0. Use the develop branch to do any development, deployment, and testing. When you want to submit your task, create a Pull Request in the GitHub website to request merging from develop to main branch. Invite Chao-Wei Peng (cwpeng) as code reviewer, wait for the feedback.

Write down your name and online API URLs in the Pull Request comment.

For example:

彭兆蔚

http://140.112.3.5:8000/

http://140.112.3.5:8000/attraction/10