Workflow

Course Assignment

Table of Contents

- Goal
- Brief
- Important Notes
- Process
- Delivery
- Resources

Goal

Using the skills and knowledge from this course, improve the quality of a `package` by establishing helpful `workflows` that make the development process more efficient.

Brief

Although Quality Assurance is a practice we should be concerned with throughout the development cycle, it is common to `sprint` towards an `MVP` version of an application before later improving the quality. In this exercise you are tasked with improving the quality of an existing application by providing various development workflows as well as a testing strategy.

There are two ways to complete this assignment:

1. Teams of two developers, working on each other's applications

2. Solo developers, working on the provided application ⬈

The existing application package must feature:

- A front-end login form connected to an API JWT endpoint

- A front-end logout button connected to browser storage

- Front-end CRUD functionality for at least one object type

- A front-end profile page

> The Noroff API Social Media client works well with this exercise.

Important Notes

Test Failure: In some cases, a well written test will still fail if the project does not meet the expectations. For example, the project may not show user error messages. In this case the test case `The user cannot submit the login form with invalid credentials and is shown a message.` **should** fail. Writing tests that always pass is not the purpose of this assignment.

Issues: While working on another project you may encounter bugs or issues with this project. If a test is failing, create a new Issue to contact the project owner. If you wish to offer a fix for this bug, please do so in a separate Pull Request, for example from the branch `worfklow-fixes`.

Process

1. Find a partner to work with or select the example repository.

2. `Fork` the project repository to your GitHub account.

3. Create a new branch called `workflow`.

4. Configure the project with eslint, prettier and commit hooks.

5. Configure the project with GitHub Actions for build/deploy if required.

6. Record any bugs found during this process in the Issues tab.

7. Configure the project for Jest and Cypress.

8. Create tests to cover the required test cases.

9. Create a Pull Request from `workflow` into the `default branch`.

10. Request a review from your peers to help improve your submission.

11. Make any final changes based on this feedback.

12. Submit a link to the **Open Pull Request** on Moodle.

Delivery

Please deliver an **open** Pull Request from branch `workflow` into your fork's default branch.

You may open a PR from `workflow` into the `upstream` default branch in addition, but this will not be graded.

Please post your PR to the peer review forum and provide feedback to your peers when requested.

Resources

- Noroff Social Media Application ↗