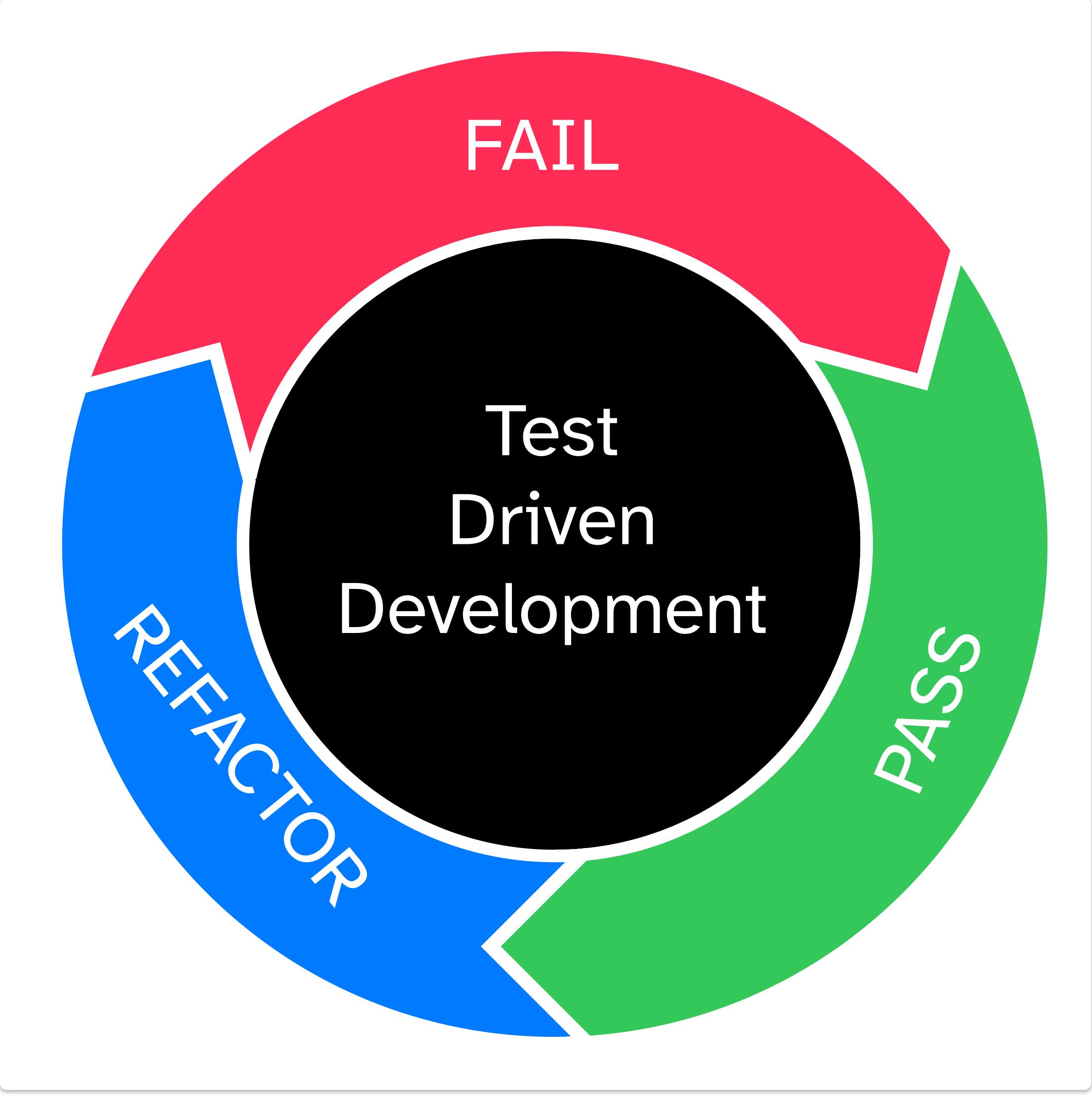
Test-Driven Development of an API

Software as a Service - Back-End Development
Session 01

Developed by Adrian Gould

Contents

- Test-Driven Development of an API
- What is TDD
- The Phases of TDD
- Best Practices



What is TDD

Test-Driven Development (TDD) is a software development approach where you write tests before writing the actual code.

This ensures that your code meets the requirements and works as expected.

Laravel 11, combined with the Pest testing system, provides a powerful and efficient way to implement TDD.

The Phases of TDD

Red-Green-Refactor Cycle:

- Red:
  - Write a failing test.

- Green:
  - Write the minimum amount of code to make the test pass.
- Refactor:
  - Improve the code while ensuring the tests still pass.

As we have already seen in the previous lecture, we then go back to the top and repeat the process, slowly developing the full feature, testing each component as we go.

Best Practices

Keep Tests Independent:

- Each test should be independent and not rely on the state of other tests.

Use Descriptive Test Names:

- Use clear and descriptive names for your tests to make them easy to understand.

Test Edge Cases:

- Write tests for edge cases and exceptional scenarios to ensure robustness.

Iterate and Improve:

- Continuously write tests and improve your codebase.
- TDD helps in catching bugs early and ensures that your code meets the requirements.