Full Stack Application Development- SE ZG503

Akshaya Ganesan

CSIS, WILP

BITS Pilani

Pilani Campus

innovate achieve lead

BITS Pilani

Pilani Campus

Lecture No: 16 Testing

Test Pyramid

- But when it comes to automated tests, how many of each test do we want?

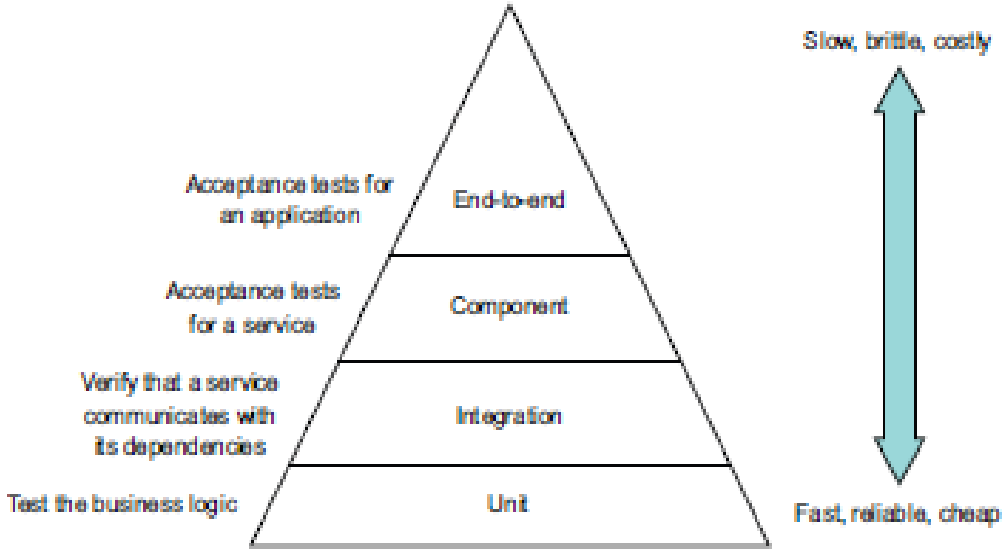


Figure 9.5 The test pyramid describes the relative proportions of each type of test that you need to write. As you move up the pyramid, you should write fewer and fewer tests.

Frontend Testing

- Unit Testing
  - Developers isolate the minor application components, check their behavior, and identify defects early in the development pipeline.
- Visual Regression Testing
  - Also sometimes called visual snapshot tests
  - Comparing screenshots taken before and after code changes
- Cross Browser Testing
  - check if a website works as expected when accessed via different browser-device-OS combinations.
  - Responsive Design
- Accessibility Testing
  - checks if every user on the internet can easily access a website, including individuals with special needs
- Acceptance Testing

Jest

- Jest comes with a test runner, assertion library, and good mocking support
- Test runner — a tool that picks up files that contain unit tests, executes them, and writes the test results to the console or log files. It automatically finds tests to be executed in your repository
- Assertion library — verifies the results of a test.
- Mocks — used in unit testing a component. A component under test has many dependencies. Jest automatically mocks the dependencies when you're running your tests.
- Mocking library — facilitates the usage of mocks in unit testing.

Jest Test Suite

```
describe('my function or component', () => {
  test('does the following', () => {
   expect(true).toBe(true);
  });
});
```

- describe-block is the test suite,
- the test-block (which also can be named it instead of test) is the test case
- Expect() is the assesrtion

The React Testing Library

- The React Testing Library is a very light-weight solution for testing React components.
- It provides light utility functions on top of react-dom and react-dom/test-utils
- The utilities this library provides facilitate querying the DOM in the same way the user would.
- Finding form elements by their label text (just like a user would), finding links and buttons by their text (like a user would).

API Testing

API TESTING

- Scope is one way of characterizing tests
- Unit tests—Test a small part of a service, such as a class.
- Integration tests—Verify that a service can interact with infrastructure services such as databases and other application services.
- Component tests—Acceptance tests for an individual service.
- End-to-end tests—Acceptance tests for the entire application.

Unit Testing

- These are tests that typically test a single function or method call.
- These are tests that help the developers and so they would be technology-facing, not business facing
- The prime goal of these tests is to give us very fast feedback about whether our functionality is good.

Types of Unit Testing

- Solitary unit test—Tests a class in isolation using mock objects for the class's dependencies
- Sociable unit test—Tests a class with its dependencies

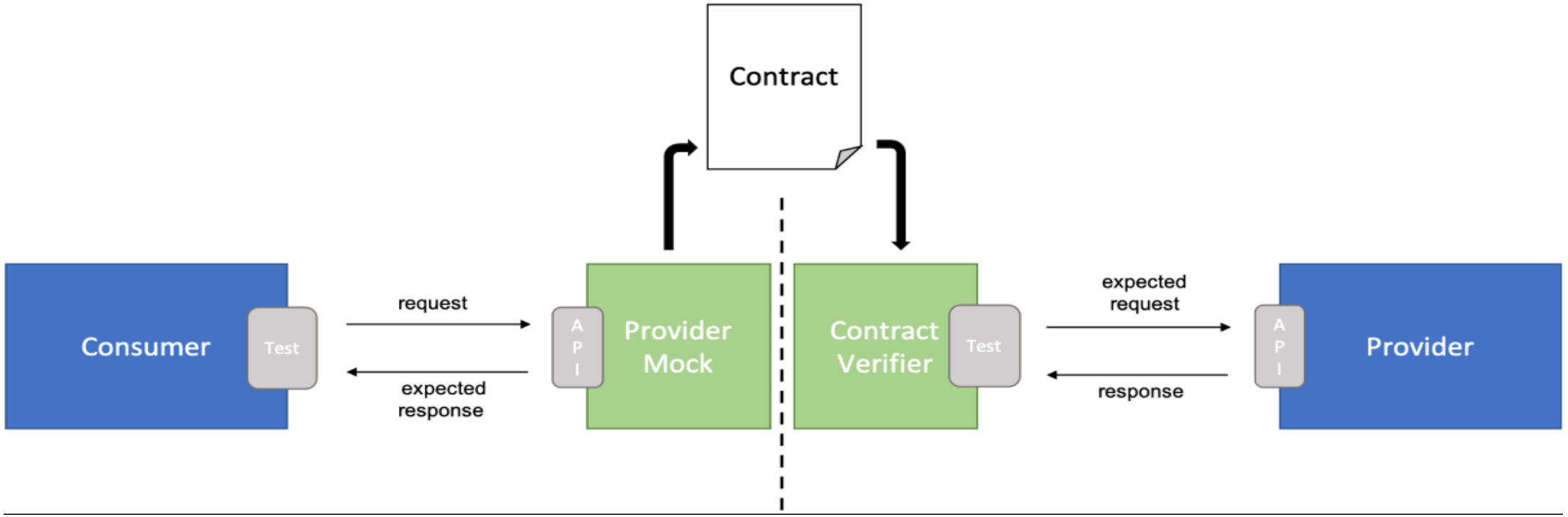
Integration Tests

- They verify that a service can properly interact with infrastructure and other services.
- Postman can be used effectively for integration testing, where you test how different components of your application interact.

Contract testing

- Contract testing is a software testing methodology that tests the interactions between different microservices or software components based on their contracts.
- In contract testing, each service or component is given a contract, which defines how to work with the service and which responses to accept.
- There are two types of contract testing: consumer-driven and provider-driven.
- **Consumer-Driven Contract Testing**
- Consumer-driven contract testing is a contract testing approach in which the service consumer, for example, a client application, defines the contracts the service provider must adhere to.
- **Provider-Driven Contract Testing**
- In provider-driven contract testing, the service provider defines the contracts that the consumer (i.e., the client) must adhere to when consuming the service. These contracts are usually defined in a format that can be shared between the service provider and the client, like OpenAPI or Swagger.

Consumer-driven Contract Testing



Self Reading

- https://semaphoreci.com/blog/test-microservices

innovate achieve lead

BITS Pilani

Pilani Campus

Thank you