

Q: What are different types of testing?

A: Different types of testing is:

1. **Manual Testing:** Testing done by humans manually.
2. **Automation Testing:** Testing that involves writing code to test the functionalities. Done by tools like Selenium.
3. **End 2 End Testing:** Stimulates user and covers complete user journey. Done by tools like Cypress. It uses a headless browser to perform testing. A headless browser emulates an actual browser but doesn't have an UI. With headless browser we can execute test cases faster since it doesn't need to paint on the browser.
4. **Unit Testing:** Testing a single unit of our application. Core job of developers. It is a small component feature testing. Done by tools like RTL and jest. Ex: Is the header loading or not?
5. **Integration Testing:** Testing the integration between different components. It tests whether two components work correctly together.

Q: What is Enzyme?

A: Enzyme is a JavaScript Testing utility for React that makes it easier to assert, manipulate, and traverse your React Components' output. It was developed at Airbnb and later transferred to an independent organization. Enzyme's API is meant to be intuitive and flexible by mimicking jQuery's API for DOM manipulation and traversal. Enzyme is unopinionated regarding which test runner or assertion library you use, and should be compatible with all major test runners and assertion libraries out there

Q: Enzyme vs RTL?

A: **React Testing Library:** RTL was introduced in 2018 and it focuses on testing user behaviour instead of implementation details. It focuses on checking that the UI responds the way user is expecting it to behave. It does not test the inner complex implementation details such as state and properties of the component. It includes built-in assertions for checking expected behaviour and simple and intuitive APIs for selecting and interacting with the elements of React component.

Enzyme: Enzyme is a testing utility for React which was released in 2015. It differs from RTL because it focuses on testing implementation details of the components. It includes API for mounting, rendering and manipulating React components in tests. It enables us to test and manipulate a component's internal state and properties as well as simulate user interaction. This can be beneficial when testing complex components with lots of internal state or that interacts with external services.

Q: What is jest and why do we use it?

A: Jest is a JavaScript testing framework built on top of Jasmine and maintained by Meta. It was designed with a focus on simplicity and support for large web applications. It is used to perform unit and integration testing on our React components.

Q: RTL vs Jest?

A: **What is React Testing Library?**

Provides virtual DOMs for tests - Any time we are running tests without a browser, we need to have a virtual DOM so we can do things like click elements and we can see if virtual DOM behaves like it should do (like changing colour of a div on a button click and so on).

How Jest is related to React Testing Library?

- Jest is a Javascript testing library(not only for React, but for any JS related framework or library). Jest is a **test runner** and is responsible for

- Finding Tests
- Running the tests
- Determining whether the tests pass or fail
- **Both React Testing Library and Jest work together**