

React Native testing

Most websites and official React Native documentations recommend using **Jest Framework**

There are also *react native testing library* or *Enzyme*, which are used with Jest

Jest is a **test runner** and is responsible for

- Finding Tests
- Running the tests
- Determining whether the tests pass or fail

The React Native Testing Library (RNTL) is a lightweight solution for testing React Native components. It provides light utility functions on top of react-test-renderer.`

1) Unit Testing

- tests individual components or units of code in isolation
- Libraries:
 1. *Jest*
 2. *Enzyme*
 3. *React Testing Library*
- Unit test example with **Jest**

```
it('given a date in the past, colorForDueDate() returns red', () => {  
  expect(colorForDueDate('2000-10-20')).toBe('red');  
});
```

2) Snapshot Testing

- testing method to ensure the consistency of UI components on different renders over time.
- Libraries: **Jest**
- A "component snapshot" is a JSX-like string created by a custom React serializer built into Jest.
- Snapshot testing process
 1. Create component
 2. Run snapshot test
 3. Snapshot is saved to a file and can be reviewed

4. Update snapshot when changing component

- guide on Jest snapshot testing: <https://jestjs.io/docs/snapshot-testing>

3) Mocking

- “Mocking” is when you replace some dependency of your code with your own implementation.
- Can be used to mock libraries like axios
- Note from video: React might have setting to reset mocks on every run (causes fail)
 - Possible Fix (for React) node modules -> react-scripts-> utils-> createJestConfig.js
 - set resetMocks to false
- **Jest** Mocking guide - <https://jestjs.io/docs/mock-functions#mocking-modules>

4) E-2-E testing

- simulates real user scenarios and aims to validate that the entire application behaves correctly, including its user interface and underlying functionality.
- Cross-Platform tools: **Apium**, *Maestro*

5) Smoke testing

- Smoke test is done to see if the system is ready to be tested
- In React Native, smoke test can be considered if app is built and main component is rendered
- Use Jest **it** function to render a component
- Reference: <https://stackoverflow.com/questions/75476604/smoke-test-with-reactjs-and-jest>

references:

- <https://blog.logrocket.com/guide-unit-testing-react-native/>
- <https://jestjs.io/docs/tutorial-react-native>
- <https://reactnative.dev/docs/testing-overview>
- <https://testing-library.com/docs/react-native-testing-library/intro/>