Unit Testing ReactJS Apps (Jest & Enzyme)

Q1. What is the main purpose of Jest in React projects?

Answer: Jest is mainly used as a unit testing framework in React applications.

Q2. Which command is commonly used to run Jest tests?

Answer: The npm run test command is typically used to execute Jest test cases.

Q3. Enzyme is primarily used for:

Answer: Enzyme is mainly used for testing React components.

Q4. Which Jest function is used to group related test cases?

Answer: The describe() function is used in Jest to organize and group related test cases together.

Q5. In Jest, which function is used to make assertions?

Answer: Assertions in Jest are written using the expect() function.

Q6. What does the shallow() method from Enzyme do?

Answer: The shallow() method renders only the current component without rendering its child components.

Q7. Which function is used to render a component with full DOM in Enzyme?

Answer: The mount() function is used in Enzyme when a full DOM rendering of a component is required.

Q8. Which Jest matcher is used to check strict equality?

Answer: The toBe() matcher is used in Jest to check for strict equality.

Q9. What does Jest's toEqual() check for?

Answer: The toEqual() matcher verifies deep equality for objects and arrays in Jest.

Q10. Which Enzyme method simulates user interactions?

Answer: User interactions in Enzyme are tested using the simulate() method.

Q11. Jest's beforeEach() function is used to:

Answer: The beforeEach() function in Jest runs before every individual test case.

Q12. Snapshot testing in Jest ensures that:

Answer: Snapshot testing ensures that a component’s output has not changed unexpectedly.

Q13. Which file extension is commonly used for Jest test files?

Answer: Jest test files are usually written with the .test.js extension.

Q14. Which Jest function is used to mock dependencies?

Answer: Dependencies in Jest can be mocked using jest.fn(), jest.mock(), or jest.spyOn().

Q15. What does enzyme-adapter-react-16 (or latest) do?

Answer: The enzyme-adapter bridges Enzyme with the corresponding React version to make component testing possible.

Q16. In Jest, which function is used to run a single test file?

Answer: A single Jest test file can be run using the command npm test fileName.test.js.

Q17. Enzyme's find() method is used to:

Answer: The find() method is used in Enzyme to locate a DOM node or component within the rendered output.

Q18. Which Jest configuration file is commonly used?

Answer: The commonly used Jest configuration file is jest.config.json.

Q19. Which Jest function is used to check if a function is called?

Answer: The matcher toHaveBeenCalled() is used in Jest to check whether a function has been called.

Q20. In Jest, which option enables coverage reports?

Answer: Running npm test --coverage in Jest generates coverage reports.

E2E Testing using Cypress

Q21. Cypress is mainly used for:

Answer: Cypress is mainly used for end-to-end testing of applications.

Q22. Cypress tests run directly inside the:

Answer: Cypress tests are executed directly in the browser.

Q23. Which Cypress command is used to visit a page?

Answer: The cy.visit() command is used in Cypress to open a specific page.

Q24. Cypress test files are usually written in:

Answer: Cypress test files are generally written in JavaScript or TypeScript.

Q25. Cypress default folder for integration tests is:

Answer: The default folder for Cypress integration tests is /cypress/integration/.

Q26. Cypress supports which type of testing?

Answer: Cypress supports unit, component, and integration testing – essentially all of them.

Q27. To get an element by ID in Cypress:

Answer: To get an element by ID, Cypress uses the command cy.get('#id').

Q28. Which Cypress command is used to simulate a click?

Answer: The cy.click() command is used to simulate a click action in Cypress.

Q29. Cypress provides:

Answer: Cypress provides a real browser environment for running tests.

Q30. Cypress runs asynchronously but provides:

Answer: Cypress manages asynchronous code by automatically waiting for commands and assertions.

Q31. Cypress command to type text into an input:

Answer: Text can be typed into an input field using cy.type('text').

Q32. Cypress command to assert visibility:

Answer: The visibility of an element is checked using cy.should('be.visible').

Q33. Cypress supports test retries with:

Answer: Test retries in Cypress can be enabled with Cypress.config({ retries }).

Q34. Cypress dashboard is used for:

Answer: The Cypress dashboard is used to view real-time test results and analytics.

Q35. Cypress automatically handles:

Answer: Cypress automatically handles asynchronous code and waits, reducing the need for manual waits.

Redux Data, Async and Data Fetching

Q36. In Redux, the state must always be treated as:

Answer: Redux state must always be treated as immutable.

Q37. Which function in Redux is used to combine multiple reducers into one?

Answer: Multiple reducers are combined using the combineReducers function.

Q38. Which Redux middleware is commonly used for handling asynchronous operations?

Answer: The redux-thunk middleware is commonly used for handling asynchronous tasks.

Q39. In Redux, what does dispatch() do?

Answer: The dispatch() function sends an action to the Redux store.

Q40. What is the correct order of Redux data flow?

Answer: The correct data flow in Redux is: UI → Action → Reducer → Store → UI.

Q41. When fetching data in Redux, where should the API call usually be placed?

Answer: API calls are usually placed inside an action creator using middleware.

Q42. Which hook is often used in React-Redux for accessing state from the store?

Answer: The useSelector hook is used to access state from the Redux store.

Q43. Which hook is used in React-Redux to dispatch actions?

Answer: The useDispatch hook is used to send actions to the Redux store.

Q44. What does an action in Redux contain?

Answer: An action in Redux typically contains a type and an optional payload.

Q45. What happens if you try to mutate Redux state directly?

Answer: Direct mutation of Redux state breaks immutability principles and time-travel debugging.

Q46. Which of the following is NOT true about Redux store?

Answer: It is not true that Redux allows direct modification of the store's state.

Q47. What is the main purpose of Redux middleware?

Answer: Middleware intercepts actions before they reach the reducer, allowing extra logic.

Q48. Which library is recommended by the Redux team for writing Redux logic?

Answer: The Redux team recommends using redux-toolkit for writing Redux logic.

Q49. What does the mapStateToProps function do in React-Redux?

Answer: mapStateToProps maps Redux state values to props in a React component.

Q50. If you want to initialize state in Redux with server data, which lifecycle stage is best?

Answer: Server data is usually fetched before rendering using useEffect.

RxJS & Redux-Observables, Reducers & Actions

Q51. What does RxJS primarily help with in React applications?

Answer: RxJS is mainly used for handling asynchronous data streams in React applications.

Q52. What is a Redux-Observable?

Answer: Redux-Observable is a middleware that helps handle async logic using RxJS in Redux apps.

Q53. Which RxJS operator is commonly used in Redux-Observable epics for mapping actions?

Answer: The switchMap operator is commonly used in Redux-Observable epics to map actions.

Q54. In Redux, what is the role of a reducer?

Answer: A reducer describes how the state should change based on the dispatched actions.

Q55. Which of the following best describes Redux actions?

Answer: Redux actions are plain JavaScript objects that describe changes in the state.

Q56. Which operator cancels the previous observable when a new one is emitted?

Answer: The switchMap operator cancels the previous observable and switches to the new one.

Q57. What is the purpose of implementing Undo History in Redux?

Answer: Undo History allows the UI state to be rolled back when needed.

Q58. Which Redux concept is most important for supporting undo/redo?

Answer: Immutable state is the key concept that makes undo/redo possible in Redux.

Q59. What does ImmutableJS provide?

Answer: ImmutableJS provides persistent and immutable data structures for applications.

Q60. Which of the following is a key advantage of ImmutableJS in Redux apps?

Answer: ImmutableJS improves performance by using structural sharing.

Q61. In RxJS, which operator is best for handling multiple values sequentially without cancellation?

Answer: The concatMap operator processes multiple values sequentially without cancellation.

Q62. Which function is used in Redux to combine multiple reducers?

Answer: Multiple reducers can be combined using the combineReducers() function.

Q63. In Redux-Observable, what is an Epic?

Answer: An Epic is a middleware function that listens for actions and returns new actions.

Q64. ImmutableJS provides which method to update deeply nested data without mutation?

Answer: The updateIn() method is used to update deeply nested data structures immutably.

Q65. In Undo History implementation, which Redux principle ensures previous states can be restored?

Answer: The principle of immutability ensures that past states can be restored in Redux.

Redux-Thunk & Redux-Saga

Q66. What is the primary purpose of Redux-Thunk?

Answer: Redux-Thunk allows action creators to return functions instead of plain actions.

Q67. In Redux-Thunk, the function returned by an action creator receives:

Answer: The returned function has access to both dispatch and getState.

Q68. Which of the following is a use case for Redux-Thunk?

Answer: Redux-Thunk is often used to handle asynchronous API calls.

Q69. Redux-Saga is based on which JavaScript feature?

Answer: Redux-Saga is built on top of JavaScript generator functions.

Q70. In Redux-Saga, which effect is used to call asynchronous functions?

Answer: The call effect is used in Redux-Saga to invoke asynchronous functions.

Q71. Which Redux middleware is best for complex async workflows like cancellation and sequencing?

Answer: Redux-Saga is best suited for handling complex async workflows.

Q72. The put effect in Redux-Saga is used to:

Answer: The put effect is used to dispatch actions in Redux-Saga.

Q73. In Redux-Saga, the takeEvery effect does what?

Answer: The takeEvery effect runs a saga for every matched action.

Q74. Which effect in Redux-Saga runs only the latest task and cancels previous ones?

Answer: The takeLatest effect cancels previous tasks and runs only the most recent one.

Q75. What is a key difference between Redux-Thunk and Redux-Saga?

Answer: Redux-Thunk uses functions and promises, while Redux-Saga relies on generator functions.

Q76. Which middleware allows you to retry failed API calls automatically?

Answer: Redux-Saga can automatically retry failed API calls.

Q77. In Redux-Saga, yield call(apiFunction) ensures:

Answer: The call effect ensures the function runs asynchronously.

Q78. Which of these is NOT true about Redux-Thunk?

Answer: It is not true that Redux-Thunk requires generator functions.

Q79. Which effect is used to pause execution in Redux-Saga for a given time?

Answer: The delay effect is used to pause execution in Redux-Saga.

Q80. Which scenario would benefit more from Redux-Saga than Redux-Thunk?

Answer: Complex async workflows with cancellation benefit more from Redux-Saga.

Q81. What is the main purpose of Redux-Thunk?

Answer: The main purpose of Redux-Thunk is to handle asynchronous logic in Redux.

Q82. Redux-Thunk allows dispatching of:

Answer: With Redux-Thunk, both functions and objects can be dispatched.

Q83. Which middleware is required to enable Redux-Thunk?

Answer: To use Redux-Thunk, the redux-thunk middleware must be added.

Q84. In Redux-Saga, which effect is used to call an asynchronous function?

Answer: The call effect is used to invoke asynchronous functions.

Q85. What does the put effect in Redux-Saga do?

Answer: The put effect dispatches an action in Redux-Saga.

Q86. Redux-Saga is built on top of:

Answer: Redux-Saga is built using JavaScript generator functions.

Q87. In Redux-Saga, the takeLatest effect is used to:

Answer: The takeLatest effect cancels earlier tasks and runs only the latest one.

Q88. Redux-Thunk is best suited for:

Answer: Redux-Thunk is best for simple async logic like API requests.

Q89. Which effect in Redux-Saga is used to watch for dispatched actions?

Answer: The take effect is used to listen for dispatched actions.

Q90. Redux-Saga can handle:

Answer: Redux-Saga is capable of handling complex asynchronous workflows.

Q91. Which of the following is TRUE about Redux-Thunk?

Answer: Redux-Thunk allows action creators to return functions.

Q92. In Redux-Saga, the select effect is used to:

Answer: The select effect gives access to the Redux store state.

Q93. Which is an advantage of Redux-Saga over Redux-Thunk?

Answer: Redux-Saga handles side effects in a more declarative way.

Q94. Which effect in Redux-Saga allows running multiple effects in parallel?

Answer: The all effect is used to run multiple effects in parallel.

Q95. In Redux-Saga, what does the fork effect do?

Answer: The fork effect spawns a non-blocking task.

React i18n

Q96. What does i18n stand for in software development?

Answer: i18n stands for Internationalization.

Q97. How many letters are between the first and last letters in “Internationalization” that form i18n?

Answer: There are 18 letters between the first and last letters, hence i18n.

Q98. Which React library is most commonly used for i18n support?

Answer: The react-i18next library is commonly used for i18n in React.

Q99. In react-i18next, what is the hook used to access translation functions?

Answer: The useTranslation hook provides access to translation functions.

Q100. Which JSON structure is typically used for storing translations?

Answer: Translations are usually stored as key-value pairs in JSON.

Q101. In i18n, what is “l10n”?

Answer: l10n refers to Localization.

Q102. Which of the following is NOT an i18n challenge?

Answer: Component state management is not considered an i18n challenge.

Q103. In i18next, what option allows you to fallback to a default language?

Answer: The fallbackLng option is used for fallback languages.

Q104. Which of these locales is valid for US English?

Answer: The correct locale code for US English is en-us.

Q105. What is the purpose of ICU message formatting in i18n?

Answer: ICU message formatting supports pluralization and gender rules.

Q106. What does the Trans component in react-i18next help with?

Answer: The Trans component allows translated strings to include React elements.

Q107. Which of the following is NOT a feature of react-i18next?

Answer: Automatic code-splitting is not provided by react-i18next.

Q108. If a translation key is missing, what will i18next usually display?

Answer: By default, i18next displays the key itself when a translation is missing.

Q109. Which React feature can help with dynamic text changes in multiple languages?

Answer: The Context API can help manage dynamic language changes.

Q110. Which statement is TRUE about localization?

Answer: Localization adapts content for specific regions and cultures, not just language.