Unit Testing ReactJS Apps (Jest & Enzyme)

Q1. What is the main purpose of Jest in React projects?
a) State management
b) Unit testing framework
c) CSS styling tool
d) Routing library
Answer: b

Q2. Which command is commonly used to run Jest tests?
a) npm run start
b) npm run test
c) npm run build
d) npm run lint
Answer: b

Q3. Enzyme is primarily used for:
a) API requests
b) Component testing
c) Redux store management
d) Internationalization
Answer: b

Q4. Which Jest function is used to group related test cases?
a) test()
b) expect()
c) describe()
d) it()
Answer: c

Q5. In Jest, which function is used to make assertions?
a) test()
b) expect()
c) render()
d) shallow()
Answer: b

Q6. What does the shallow() method from Enzyme do?
a) Fully renders the component tree

b) Renders only the given component without its children
c) Tests API endpoints
d) Creates Redux store mocks
Answer: b

Q7. Which function is used to render a component with full DOM in Enzyme?
a) shallow()
b) render()
c) mount()
d) snapshot()
Answer: c

Q8. Which Jest matcher is used to check strict equality?
a) toBe()
b) toEqual()
c) toMatch()
d) toContain()
Answer: a

Q9. What does Jest's toEqual() check for?
a) Reference equality
b) Deep equality of objects/arrays
c) String matching
d) Boolean comparison
Answer: b

Q10. Which Enzyme method simulates user interactions?
a) simulate()
b) trigger()
c) act()
d) mockEvent()
Answer: a

Q11. Jest's beforeEach() function is used to:
a) Run once before all tests
b) Run before every test case
c) Run once after all tests
d) Run only if tests fail

Answer: b

Q12. Snapshot testing in Jest ensures that:
a) API responses match schema
b) Component output hasn't unexpectedly changed
c) Redux store is updated
d) Database schema is consistent
Answer: b

Q13. Which file extension is commonly used for Jest test files?
a) .jsx
b) .test.js
c) .snap.js
d) .mock.js
Answer: b

Q14. Which Jest function is used to mock dependencies?
a) jest.fn()
b) jest.mock()
c) jest.spyOn()
d) All of the above
Answer: d

Q15. What does enzyme-adapter-react-16 (or latest) do?
a) Adds Redux support
b) Bridges Enzyme with React version
c) Enables Jest matchers
d) Creates snapshots
Answer: b

Q16. In Jest, which function is used to run a single test file?
a) npm test fileName.test.js
b) jest --runSingle
c) npm run build fileName
d) testOnly fileName
Answer: a

Q17. Enzyme's find() method is used to:
a) Locate a DOM node or component

b) Simulate user clicks
c) Create a snapshot
d) Mount the component
Answer: a

Q18. Which Jest configuration file is commonly used?
a) jest.config.json
b) jestfile.json
c) test.config.js
d) enzyme.config.js
Answer: a

Q19. Which Jest function is used to check if a function is called?
a) toBeCalled()
b) toHaveBeenCalled()
c) wasCalled()
d) fnCalled()
Answer: b

Q20. In Jest, which option enables coverage reports?
a) npm test --verbose
b) npm test --coverage
c) npm test --watch
d) npm test --report
Answer: b

E2E Testing using Cypress

Q21. Cypress is mainly used for:
a) Backend testing
b) API testing
c) End-to-End testing
d) Database testing
Answer: c) End-to-End testing

Q22. Cypress tests run directly inside the:
a) Browser
b) Server
c) Database

d) IDE
Answer: a) Browser

Q23. Which Cypress command is used to visit a page?
a) cy.load()
b) cy.goto()
c) cy.visit()
d) cy.open()
Answer: c) cy.visit()

Q24. Cypress test files are usually written in:
a) HTML
b) JavaScript/TypeScript
c) Python
d) C#
Answer: b) JavaScript/TypeScript

Q25. Cypress default folder for integration tests is:
a) /cypress/tests/
b) /cypress/integration/
c) /tests/integration/
d) /src/tests/
Answer: b) /cypress/integration/

Q26. Cypress supports which type of testing?
a) Unit
b) Component
c) Integration
d) All of the above
Answer: d) All of the above

Q27. To get an element by ID in Cypress:
a) cy.getElement('#id')
b) cy.find('#id')
c) cy.get('#id')
d) cy.locate('#id')
Answer: c) cy.get('#id')

Q28. Which Cypress command is used to simulate a click?
a) cy.press()
b) cy.tap()
c) cy.click()
d) cy.triggerClick()
Answer: c) cy.click()

Q29. Cypress provides:
a) Real browser environment
b) Virtual DOM only
c) API mocks only
d) Only CLI support
Answer: a) Real browser environment

Q30. Cypress runs asynchronously but provides:
a) Callbacks
b) Automatic waiting
c) Promises only
d) Manual waits
Answer: b) Automatic waiting

Q31. Cypress command to type text into an input:
a) cy.enter('text')
b) cy.type('text')
c) cy.write('text')
d) cy.input('text')
Answer: b) cy.type('text')

Q32. Cypress command to assert visibility:
a) cy.get().visible()
b) cy.should('be.visible')
c) cy.expect('visible')
d) cy.assert('visible')
Answer: b) cy.should('be.visible')

Q33. Cypress supports test retries with:
a) Retry block
b) Cypress.config({ retries })
c) cy.retry()
d) cy.loop()
Answer: b) Cypress.config({ retries })

Q34. Cypress dashboard is used for:
a) Real-time test results & analytics
b) Code compilation
c) Storing test code
d) Database queries
Answer: a) Real-time test results & analytics

Q35. Cypress automatically handles:
a) Async code & waits
b) SQL queries

c) Cloud deployments
d) Webpack builds
Answer: a) Async code & waits

Redux Data, Async and Data Fetching

Q36. In Redux, the state must always be treated as:
a) Mutable
b) Immutable
c) Temporary
d) None of the above
Answer: b) Immutable

Q37. Which function in Redux is used to combine multiple reducers into one?
a) createReducer
b) applyMiddleware
c) combineReducers
d) rootReducer
Answer: c) combineReducers

Q38. Which Redux middleware is commonly used for handling asynchronous operations?
a) redux-logger
b) redux-thunk
c) redux-toolkit
d) redux-persist
Answer: b) redux-thunk

Q39. In Redux, what does dispatch() do?
a) Updates the reducer directly
b) Sends an action to the store
c) Returns the current state
d) Initializes the store
Answer: b) Sends an action to the store

Q40. What is the correct order of Redux data flow?
a) Store → Reducer → Action → UI
b) UI → Action → Reducer → Store → UI

c) Reducer → Store → Action → UI
d) Action → UI → Reducer → Store
Answer: b) UI → Action → Reducer → Store → UI

Q41. When fetching data in Redux, where should the API call usually be placed?
a) Inside the reducer
b) Inside the action creator (with middleware)
c) Inside the store directly
d) Inside the component only
Answer: b) Inside the action creator (with middleware)

Q42. Which hook is often used in React-Redux for accessing state from the store?
a) useEffect
b) useReducer
c) useSelector
d) useContext
Answer: c) useSelector

Q43. Which hook is used in React-Redux to dispatch actions?
a) useDispatch
b) useSelector
c) useReducer
d) useAction
Answer: a) useDispatch

Q44. What does an action in Redux contain?
a) Reducers and state
b) State and middleware
c) Type and payload
d) Store and UI
Answer: c) Type and payload

Q45. What happens if you try to mutate Redux state directly?
a) Redux automatically corrects it
b) State updates without problems
c) It breaks time-travel debugging and immutability principles
d) Nothing happens
Answer: c) It breaks time-travel debugging and immutability principles

Q46. Which of the following is NOT true about Redux store?
a) It holds the application state
b) It allows direct modification of state
c) It is updated only through dispatching actions
d) It is created using createStore or configureStore
Answer: b) It allows direct modification of state

Q47. What is the main purpose of Redux middleware?
a) To directly update the reducer
b) To log only errors
c) To intercept actions before they reach the reducer
d) To remove boilerplate code
Answer: c) To intercept actions before they reach the reducer

Q48. Which library is recommended by the Redux team for writing Redux logic?
a) redux-observables
b) redux-toolkit
c) redux-thunk
d) redux-persist
Answer: b) redux-toolkit

Q49. What does the mapStateToProps function do in React-Redux?
a) Maps component props to reducer
b) Maps Redux state to React component props
c) Maps React component props to Redux store

d) Maps dispatch to reducers
Answer: b) Maps Redux state to React component props

Q50. If you want to initialize state in Redux with server data, which lifecycle stage is best?
a) Before rendering the component (useEffect)
b) Inside the reducer directly
c) Only during store creation
d) Inside UI event handlers
Answer: a) Before rendering the component (useEffect)

RxJS & Redux-Observables, Reducers & Actions

Q51. What does RxJS primarily help with in React applications?
a) Managing UI components
b) Handling asynchronous data streams
c) Styling components
d) Building server-side apps
Answer: b) Handling asynchronous data streams

Q52. What is a Redux-Observable?
a) A middleware for handling async logic with RxJS
b) A library for styling React apps
c) A testing framework for React
d) A server-side rendering library
Answer: a) A middleware for handling async logic with RxJS

Q53. Which RxJS operator is commonly used in Redux-Observable epics for mapping actions?
a) switchMap
b) mapState
c) reduce
d) dispatchMap
Answer: a) switchMap

Q54. In Redux, what is the role of a reducer?
a) Dispatching actions
b) Describing state changes based on actions
c) Fetching data from API

d) Handling authentication
Answer: b) Describing state changes based on actions

Q55. Which of the following best describes Redux actions?
a) Functions that modify state directly
b) Pure functions
c) Plain JavaScript objects describing changes
d) Observables that emit state
Answer: c) Plain JavaScript objects describing changes

Q56. Which operator cancels the previous observable when a new one is emitted?
a) mergeMap
b) switchMap
c) concatMap
d) map
Answer: b) switchMap

Q57. What is the purpose of implementing Undo History in Redux?
a) To support rollback of UI state
b) To manage API requests
c) To improve performance
d) To simplify reducers
Answer: a) To support rollback of UI state

Q58. Which Redux concept is most important for supporting undo/redo?
a) Middleware
b) Reducer composition
c) Immutable state
d) Async actions
Answer: c) Immutable state

Q59. What does ImmutableJS provide?
a) Fast rendering
b) Persistent, immutable data structures
c) Authentication management
d) A testing framework
Answer: b) Persistent, immutable data structures

Q60. Which of the following is a key advantage of ImmutableJS in Redux apps?
a) Larger bundle size
b) Direct state mutation
c) Performance optimization with structural sharing
d) Automatic API fetching
Answer: c) Performance optimization with structural sharing

Q61. In RxJS, which operator is best for handling multiple values sequentially without cancellation?
a) switchMap
b) concatMap
c) mergeMap
d) flatMap
Answer: b) concatMap

Q62. Which function is used in Redux to combine multiple reducers?
a) mergeReducers()
b) joinReducers()
c) combineReducers()
d) useReducers()
Answer: c) combineReducers()

Q63. In Redux-Observable, what is an Epic?
a) A reducer that handles async data
b) A middleware function that listens for actions and returns new actions
c) A UI component that renders state
d) A Redux store enhancer
Answer: b) A middleware function that listens for actions and returns new actions

Q64. ImmutableJS provides which method to update deeply nested data without mutation?
a) setDeep()
b) updateIn()
c) pushState()
d) changeIn()
Answer: b) updateIn()

Q65. In Undo History implementation, which Redux principle ensures previous states can be restored?
a) Single source of truth
b) Actions must be pure
c) State is immutable
d) Store is asynchronous
Answer: c) State is immutable

Redux-Thunk & Redux-Saga MCQs

66. What is the primary purpose of Redux-Thunk?
a) To handle routing in React apps
b) To write action creators that return functions instead of actions
c) To test Redux reducers
d) To optimize React rendering

Answer:b

67. In Redux-Thunk, the function returned by an action creator receives:
a) Only dispatch
b) Only getState
c) Both dispatch and getState
d) Neither
Answer:c

68. Which of the following is a use case for Redux-Thunk?
a) Testing React components
b) Managing asynchronous API calls
c) Writing reducers
d) Handling routing
Answer: b

69. Redux-Saga is based on which JavaScript feature?
a) Promises
b) Generators
c) Callbacks
d) Async/Await
Answer : b

70. In Redux-Saga, which effect is used to call asynchronous functions?
a) put
b) call
c) take
d) delay
Answer: b

71. Which Redux middleware is best for complex async workflows like cancellation and sequencing?
a) Redux-Thunk
b) Redux-Saga
c) Redux-Logger
d) Redux-Observable
Answer: b

72. The put effect in Redux-Saga is used to:
a) Dispatch an action
b) Call an API
c) Cancel a task
d) Create a reducer
Answer: a

73. In Redux-Saga, the takeEvery effect does what?
a) Cancels previous sagas
b) Runs a saga for every matched action
c) Runs only the latest action saga
d) Runs sagas sequentially
Answer:b

74. Which effect in Redux-Saga runs only the latest task and cancels previous ones?
a) takeLatest
b) takeEvery
c) fork
d) join
Answer :a

75. What is a key difference between Redux-Thunk and Redux-Saga?
a) Thunk uses Promises, Saga uses Generators
b) Thunk is synchronous, Saga is asynchronous
c) Thunk is faster than Saga
d) Saga does not support async
Answer:a

76. Which middleware allows you to retry failed API calls automatically?
a) Redux-Thunk
b) Redux-Saga
c) Redux-Persist
d) Redux-Logger
Answer:b

77. In Redux-Saga, yield call(apiFunction) ensures:
a) The function is executed asynchronously
b) The action is dispatched
c) The reducer is updated
d) The store is replaced
Answer:a

78. Which of these is NOT true about Redux-Thunk?
a) It is simpler to learn than Redux-Saga
b) It uses functions instead of plain actions
c) It is good for small to medium apps
d) It requires generator functions
Answer:d

79. Which effect is used to pause execution in Redux-Saga for a given time?
a) wait
b) delay
c) timeout
d) sleep
Answer: b

80. Which scenario would benefit more from Redux-Saga than Redux-Thunk?
a) Simple API calls
b) Complex async workflows with cancellation
c) Dispatching synchronous actions
d) Static state
Answer: b

81. What is the main purpose of Redux-Thunk?
a) Handle asynchronous logic in Redux
b) Manage routing in Redux apps
c) Optimize rendering performance
d) Create reducers automatically
Answer: a) Handle asynchronous logic in Redux

82. Redux-Thunk allows dispatching of:
a) Only objects
b) Functions and objects
c) Only strings
d) Reducers directly
Answer: b) Functions and objects

83. Which middleware is required to enable Redux-Thunk?
a) redux-saga
b) redux-devtools
c) redux-thunk
d) redux-persist
Answer: c) redux-thunk

84. In Redux-Saga, which effect is used to call an asynchronous function?
a) take
b) call
c) put
d) select
Answer: b) call

85. What does the put effect in Redux-Saga do?
a) Calls an API
b) Dispatches an action
c) Waits for an action
d) Cancels a task
Answer: b) Dispatches an action

86. Redux-Saga is built on top of:
a) Async/Await
b) Observables
c) Generator functions
d) Promises only
Answer: c) Generator functions

87. In Redux-Saga, the takeLatest effect is used to:
a) Run all actions in sequence
b) Cancel previous tasks and run only the latest one

c) Queue all actions without canceling
d) Run actions in parallel
Answer: b) Cancel previous tasks and run only the latest one

88. Redux-Thunk is best suited for:
a) Complex async flows with cancellation
b) Simple async logic like API calls
c) State immutability handling
d) Reducer composition
Answer: b) Simple async logic like API calls

89. Which effect in Redux-Saga is used to watch for dispatched actions?
a) take
b) call
c) race
d) put
Answer: a) take

90. Redux-Saga can handle:
a) Only synchronous logic
b) Complex asynchronous workflows
c) CSS styling
d) DOM rendering
Answer: b) Complex asynchronous workflows

91. Which of the following is TRUE about Redux-Thunk?
a) It uses generator functions
b) It allows writing action creators that return functions
c) It replaces reducers
d) It is built on RxJS
Answer: b) It allows writing action creators that return functions

92. In Redux-Saga, the select effect is used to:
a) Fetch data from an API
b) Access the Redux store state

c) Dispatch an action
d) Cancel an effect
Answer: b) Access the Redux store state

93. Which is an advantage of Redux-Saga over Redux-Thunk?
a) Simpler for beginners
b) Handles side effects more declaratively
c) No need for middleware
d) Faster rendering always
Answer: b) Handles side effects more declaratively

94. Which effect in Redux-Saga allows running multiple effects in parallel?
a) race
b) all
c) call
d) fork
Answer: b) all

95. In Redux-Saga, what does the fork effect do?
a) Run a blocking call
b) Spawn a non-blocking task
c) Cancel all tasks
d) Dispatch a reducer
Answer: b) Spawn a non-blocking task

React i18n (15 MCQs)

Q96. What does i18n stand for in software development?
a) Integration
b) Internationalization
c) Interaction
d) Initialization
Answer: b) Internationalization

Q97. How many letters are between the first and last letters in “Internationalization” that form i18n?
a) 15
b) 17
c) 18
d) 19
Answer: c) 18

Q98. Which React library is most commonly used for i18n support?
a) react-intl
b) react-localize-redux
c) react-i18next
d) next-translate
Answer: c) react-i18next

Q99. In react-i18next, what is the hook used to access translation functions?
a) useLocale
b) useLang
c) useI18n
d) useTranslation
Answer: d) useTranslation

Q100. Which JSON structure is typically used for storing translations?
a) Array of strings
b) Key-value pairs
c) CSV format
d) XML format
Answer: b) Key-value pairs

Q101. In i18n, what is “l10n”?
a) Localization
b) Lexicalization
c) Legalization
d) Language-switching
Answer: a) Localization

Q102. Which of the following is NOT an i18n challenge?
a) Different date formats
b) Currency conversion
c) String interpolation
d) Component state management
Answer: d) Component state management

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

a) defaultLang
b) fallbackLng
c) baseLocale
d) backupLang
Answer: b) fallbackLng

Q104. Which of these locales is valid for US English?
a) en-uk
b) en-us
c) eng-us
d) en-english
Answer: b) en-us

Q105. What is the purpose of ICU message formatting in i18n?
a) To format database queries
b) To support pluralization and gender rules
c) To compress translation files
d) To sort translation keys
Answer: b) To support pluralization and gender rules

Q106. What does the Trans component in react-i18next help with?
a) Language detection
b) Rendering translated strings with React elements inside
c) Switching between locales
d) Storing translations in cookies
Answer: b) Rendering translated strings with React elements inside

Q107. Which of the following is NOT a feature of react-i18next?
a) Lazy loading translations
b) Context-based translations
c) Automatic pluralization
d) Automatic code-splitting
Answer: d) Automatic code-splitting

Q108. If a translation key is missing, what will i18next usually display?
a) Error message
b) Empty string
c) The key itself
d) Default English
Answer: c) The key itself

Q109. Which React feature can help with dynamic text changes in multiple languages?
a) Context API
b) useMemo

c) useEffect
d) Suspense
Answer: a) Context API

Q110. Which statement is TRUE about localization?
a) It always uses English as fallback
b) It adapts content to specific regions/cultures
c) It only changes the text language
d) It ignores formatting of dates and numbers
Answer: b) It adapts content to specific regions/cultures