Interview Q&A

Top 50 ReactJs Interview Questions for Experienced | 2022

🥵 By <u>Ronita Ghosh</u> | 🕓 May 31, 2020 | 🟳 <u>2 Comments</u>



After reviewing various resources such as articles and videos, we have compiled the Top 50 ReactJs Interview Questions for Senior Developers, which will help you easily pass any interview.

React is one such JavaScript (ReactJS) repository used for building user interfaces supported by the tech giant Facebook.



ReactJs Interview Questions for Senior Developer

In the context of the interactive user interface (UI) experience, React JS has garnered quite a good amount of popularity these days. It is known for its declarative, component-based, interactive scripting nature.



React JS and its importance in website or web application development

An interactive UI is a pivotal aspect of a successful website. Apparently, it is used by clients to develop websites (using specific programming and coding skills by web developers) for the website users to browse through declarative and efficient on-screen menus and buttons with ease.

Are you looking for a career as a Frontend Developer?

React certification is a must-have. As you must be preparing for the interviews, we've gathered something for you. Below are some important **react interview questions and answers** to assist you with your interview preparations.

Salary of FullStack Developer in India

1. What is React? And, What are the key features of Reactjs?

It can be defined as a component-based JavaScript Library that helps build interactive and agile UIs (user interfaces).

The major features of React include -

1. JSX,

- 2. One-way Data Binding,
- 3. Components,
- 4. Simplicity,
- 5. Virtual DOM,
- 6. Performance

2. React version history.

The first React native was released by Facebook as its News Feed in 2011. It was initially known as Facebook's React Conf and was open-sourced in March 2015.

3. How to Create React components?

The steps to create React components include:

1. Creation of class components by first naming it like "Example".

- 2. A state of the class component is added. This is done by adding a constructor to the class component.
- 3. This step is followed by the introduction of the Render method to return the JSX. It must be noted that in this case the JSX must be wrapped in an outer element like <div>.
- 4. Now, the 'this' keyword must be bound inside the constructor, for it to be used anywhere.
- 5. In order for us to use the component, it must be imported into a file (in case the components are in separate files). Once imported, use the assigned name to be added to the JSX.
- 6. This is followed by passing specific 'Props' into the component. Each prop must be given a name.
- 7. In order for the components to receive props, the props must be put into the arguments. After they are received by the components, they can be accessed in the JSX by using 'this.props.NameOfProp'.

8. Once all the above steps are performed, the code can be completed.

This can be considered one of the most frequently asked **react interview questions**.

4. **Top 10 Advantages of Using React**? (Amongst the best ReactJs Interview Questions for Senior Developer)

- 1. With the help of JSX syntax extension, the overall process of writing components becomes very easy. This accepts HTML quoting.
- 2. Enables a faster rendering for even high-load applications with the help of Virtual DOM (document object model) designed by the Facebook development team.
- 3. Ameliorates productivity and facilitates further maintenance of application by reusing the system components.
- 4. It is SEO-friendly.
- 5. The data binding aspect of ReactJS guarantees code stability and seamless app performance.
- 6. It is one of the best JS libraries that follows the same design pattern in both web and mobile app development. It is compatible with both iOS and Android platforms.
- 7. Supported and backed by strong engineering communities of Facebook and Instagram engineering teams as well as external experts. To be precise, the React GitHub repository has over 1100 contributors.
- 8. It is easy to learn.
- 9. It offers the user a distinctive range of debugging and design toolsets.
- 10. Due to its faster rendering and ease of accessibility, Fortune 500 companies and startups use ReactJs alike.

5. What are the differences between Angularjs and Reactjs?

<u>AngularJS</u> is an open-source MVC (Model-Viewer-Controller) web framework, a structural component for today's robust web apps framework, and a structural component for today's robust web apps.



The fundamental difference between the two can be listed as follows:

1. AngularJS is a JavaScript framework while the latter (ReactJS) is a library.

- 2. AngularJS is designed and controlled by Google, while ReactJS is maintained by the Facebook community.
- 3. While AngularJS uses a two-way data binding technique to connect DOM values to model data, ReactJS can support a one-way binding process.
- 4. ReactJS's one-way data flow model allows it to manage complexities efficiently. Hence, it is easy to debug large components of ReactJS as compared to that <u>AngularJS</u>.
- 5. Controllers like Redux and Flux play a pivotal role in operating the ReactJS libraries. AngularJS, on the other hand, is a framework with integrated features and tools and operates on the MVVM (Model-View-View-Model) approach.
- 6. The virtual DOM created by ReactJS boosts the web navigation performance tremendously when compared to AngularJS apps.

6. What is the difference between the Class and Function Components of React?

Class components use ES6 syntax and have lifecycle hooks to allow React to create instances and can maintain their internal state themselves.

On the other hand, Functional components also referred to as "stateless components", are functions that represent pure old JavaScripts that return React elements.

7. What are the Pure components in ReactJS?

Pure components have a significant role to play to re-render a component. With the help of a pure component, the lifecycle method is used for comparing the states or props for any changes in the properties before the re-rendering of the component is attempted.

To be precise, pure components analyze and compare the properties of the current state or prop with the next state or prop and how the parent component is different from the child component in the set state hierarchy.

8. What are the High-order components of ReactJS?

A High-order component is known for augmenting and returning an original component to the React composition. HOC or a High-order component is a function that when implemented in a React state can create a pattern without the need for code repetition.

A few examples of HOCs are Relay's 'create container' and Redux's connect, respectively.

9. How can you differentiate between the state and props in ReactJS?

State: These datasets belong to a single component and change their output while responding to certain events over time. The states are managed by the component in its entirety.

Prop: Props, on the other hand, are read-only functions. They display a fixed character throughout the component's lifecycle. Props are stateless in nature and are used for customizing components at the time of their creation.

10. Can you update the state directly in ReactJS? If not, then why?

Mutating a state directly in React is not recommended. The five primitives in any JavaScript (including React) include strings, booleans, numbers, null, and undefined. These 5 primitives cannot participate in data mutations in JavaScript. The state is a boolean primitive and cannot be used unless it is in the form "this.setState".

11. Explain the difference between the HTML and React event handling process.

1. HTML events can be written in lowercase, whereas React events are written using camelCase,

2. Unlike HTML events, the coder cannot stop or prevent a default behavior by returning false. It must be stated as 'preventDefault'.

12. What is the significance of .bind in React?

To handle event components in React, one must use binding. While running a function in React to change 'this' value using the .bind() method, or else 'this' won't be able to determine the context.

13. What are the keys in React?

The 'keys' props help the user to identify React components that have been rendered with a function of 'editing' or 'deleting' an event.

14. What are refs in ReactJS?

'Refs' or 'References' are used to access the underlying properties of a **DOM element** and **Class component** in React.

Uses of Refs:

- To set or unset a ref attribute. In the latter case, the ref attribute can be given a pass.
- To increment or decrement input values
- Integrate with 3rd party DOM libraries, e.g., jQuery File uploader, D3
- For imperative animations, i.e. by nesting one element into another component while trying to trigger a style update in React.
- To manage focus, select text or playback media.

15. How to remove duplicate values from dropdown in React?

The different ways of removing duplicate values from the dropdown list in React are:

- Set, i.e., creating a new set in the array. We can also convert a 'set' to an array via "Array.from"
- Filter This can be done via the "indexOf" method. In this case, if the item is repeated in the index, it will be removed.
- Reduce, i.e., by reducing the array of elements.

16. What is the preferred option in the case of callback refs and findDOMNode()?

Callback refs.

17. What is the difference between Real DOM and Virtual DOM?

The DOM or 'Document Object Model' is a hierarchical programming interface where the entire HTML code is represented as a tree structure. It is used to modify the content of a web page and it can be classified into:

i) 'Real' or HTML DOM

ii) Virtual DOM.

The virtual DOM is a modified version or copy of the Real DOM and is much lighter in weight. It is implemented to boost the browser-specific DOM computations, which, otherwise can be slowed down with the Real DOM system.

Note: Even though virtual DOM was not invented by React, it is still used by React and is freely available to React.js users.

18. Define JSX. What is the correlation between **React and JSX?**

While working with React, the components are preferably written in JSX. JSX helps convert HTML syntax to a better-performing coding script. It is also type-safe when compared to JavaScript and is supported by TypeScript. JSX provides a safe environment for user inputs (UI). The errors in the script are usually caught during the compilation phase itself.

19. How can you differentiate between ReactJS and Ember JS?

While React.js (an open-source JavaScript Library) is used for managing user interfaces (APIs) for web and mobile applications, Ember.js (an open-source Javascript framework) is used for creating web applications.

20. Commonly Asked React Js Interview Questions, **Explain how Virtual DOM works.**

Even though a real DOM node can be easily updated, the re-rendering of the UI updates is difficult to achieve. This calls for a system that is quick enough to render the updates on a new DOM tree, hence the introduction of virtual DOM. Virtual DOM enables users with optimized operations while reducing performance costs simultaneously.

21. How to mock a method from a functional React component using jest and Enzyme? What are the benefits of using Jest of testing React components?

When you have a functional unit component, you may want to test run it to fix any underlying bugs or functionality issues. It can be done by creating a mock function to trigger the simulated response (or Enzyme) using *jest.fn* (from Jest).

Significance of Jest for Testing React Components:

- Easy to configure.
- The run time was reduced to instant runs.
- Comes with a snapshot testing feature (i.e., the test run happens in JSON config)
- Offers a code coverage report
- Has a built-in interactive watch mode.

22. What is CRA in React?

CRA or "**create-react-app**" is a framework that helps build React applications seamlessly. The essential commodities of CRA include a module builder (or **webpack**) and a transpiler (also known as '**Babel**').

23. What is the difference between React's ES6 syntax and ES5?

Even though both ES5 and ES6 are JavaScripts, the latter is the latest syntax version and usually is unsupported by many browsers. Hence, to make ECMAScript 2015 (or ES6) work, one must convert it to ES5 (the standard syntax) with the help of transpilers like *Traceur* or *Babel*.

24. Are there any life-cycle methods in React?

What are they?

Yes! Even though lifecycle methods are changing in React, they can be classified into the following:

Phase 1 – Render Phase –

- *Mounting* or Constructor (used to set the initial state of JavaScript)
- Updating or Static or getDerivedStateFromProps (previously known as componentwillreceiveprops) => this method is used both during the initial render and rerender phase.
- Render (a mandatory method in the lifecycle)
- **Unmounting** or componentDidMount

Phase 2 – Re-render Phase

- Updating or Static getDerivedStateFromProps
- shouldComponentUpdate (if updated as "true" = render, or "false" = rendering stops)
- Render
- getSnapshotBeforeUpdate (or pre-commit phase)
- componentDidUpdate

React components' lifecycle can be classified into 3 categories. These are *Mounting*, *Updating*, *Unmounting*, and finally *Error-handling* phases respectively.

25. Define 'context' in React.

In the case of component-based UI frameworks, there may arise a necessity of passing data from one component to another (i.e., parent-child). This process can overcrowd the component tree with prop-drilling. 'Contexts' help solve this issue by sharing the data values by avoiding the prop-drilling process.

26. Define and differentiate between the controlled and uncontrolled components in React.

While creating web pages in Reactjs, React components handle form data using 'controlled' elements.



Note: Refs=uncontrolled elements, state= controlled elements

27. How to handle a Router Redirect after a user has logged in?

To help define web page visits (using the 'routing' analogy) by users (*react-router-dom* in case of web applications), the use of React hooks plays a pivotal role. While a user is logged in to a web app, he should be able to get redirected to the homepage.

Hence first, a protected component must be set up before it is rendered to match a route. The creation of a 'ProtectedRoute' helps the user pass the 'logged' state to the desired state object

path defined by the creator.

Every login string must be given a 'pathname' of '/' so that when the NavLink is actioned upon it can redirect the user to the login page (in case of correctly entered credentials) or homepage with an error description like 'Authorization Failed' (in case of incorrect credentials).

28. How can you validate props in React?

Prop validation forces React components used in the development of apps to be rendered into easily formattable codes and bug-free systems for future usage. This process helps improve React components. To achieve prop validation, running 'Type Checking' with 'PropTypes' is recommended. E.g.,

<pre>Import React, { Component } from `react';</pre>
<pre>Import `./App.css';</pre>
<pre>Import PropTypes from `prop-types';</pre>
Const Test = (props) => {
return(<h1></h1>



To check and validate a property in the above snippet,

Test.propTypes = { str:PropTypes.string }

Now, change the string type to boolean, e.g.,

Test.propTypes = { str:PropTypes.bool }

once run this comes up with an error message, for the prop supplied and expected are different.

To validate the PropType and avoid any warning or error messages, the following code snippet can be rendered:

```
Const Test = (props) => {
    return(
          <div>
         <h1>{props.str}<h1>
         <h1>{ (props.bool ? 'bool' : 'no bool') }<h1>
         </div>
     str:PropTypes.string,
         bool: PropTypes.bool
10. }
11. Class App extends Component {
   render() {
13.Iecuin14.<div className="App">
        <Test
str={`xyz'} => (str can be any pathname)
          bool={true}
            />
20. Const Test = (props) => {
21. return(
         <h1>{props.str}<h1>
         <h1>{(props.bool ? 'bool' : 'no bool')}<h1>
           </div>
       props.ary.map((val)=> {
28. return({val}
                                       =>(a list must be accompanied with a
29. Test.propTypes = {
30. str:PropTypes.string,
        bool: PropTypes.bool
         strOrNum: PropTypes.oneOfType( [PropTypes.string, PropTypes.number] )
         ary:PropTypes.arrayOf(PropTypes,number)
35. Const Test = (props) => {
36. return(
          <div>
         <h1>{props.str}<h1>
          <h1>{(props.bool ? 'bool' : 'no bool')}<h1>
           </div>
        props.ary0f0bj.map((val)=> {
43. return({val.name}
44. Class App extends Component {
    render() {
        return (
47. <div className="App">
```

<test< th=""></test<>
<pre>str={`xyz'} => (str can be any pathname)</pre>
bool
<pre>strOrNum={10} => (if we denote a number, the rendering will pass th</pre>
But, strOrNum={ true } => (will fail, for it is not a number)
Let's introduce a number array after this , i.e.,
ary={[1,2,3]
<pre>ary0f0bj={ [{name:'max', age:7}, {name:'hugh', age:'10'] }</pre>
/>
To print the above array on the web page, the following snippet can be used:
Test.propTypes = {
<pre>str:PropTypes.string,</pre>
bool: PropTypes.bool
<pre>strOrNum: PropTypes.oneOfType([PropTypes.string, PropTypes.number])</pre>
<pre>ary:PropTypes.arrayOf(PropTypes,number)</pre>
aryOfObj:PropTypes.arrayOf(PropTypes.shape(
{
name:PropTypes.string => (it will pass, as the name prop is a string, but
age:PropTypes.number



29. Explain how mixins can be incorporated into React components and enforce modularity (extend, createClass and mixins, HOCs).

Similar to how an ice cream vendor adds different flavors to your ice cream to accentuate its taste, mixins in JavaScript help extend the functionalities of various objects (classes) using codes.

HOCs (or High Order Components) replace Mixins to create a class component and return it to another class.

30. Define Redux and elucidate on Redux thunk.

Redux is a tool used for state management of React components. In react, the UI is broken down into various components (both main components and their further smaller versions). With the help of Redux, the data for various react components are stored at one location for it to be retrieved during the rendering or execution phase.

Redux thunk is an asynchronous action of executing a statement wrapped up in a function that is returned.

Applying Thunk to Redux:

Redux is usually operated by a group of abstractions, namely:

- Actions
- Action Creators
- Reducers
- Middleware

Without introducing an 'action creator' function, the process of taking action on a property becomes overwhelming and error-prone. This elaborates the use of Redux in defining functions to actions. Thunk helps reduce a function into a bundle instead of an action object. In Redux, new actions are rendered using the 'dispatch' action, while the current state is retained via the 'getState' action. In Redux thunk, every action is passed through functions by applying middleware action.

E.g.,

```
2. Return ( {dispatch, getState} ) => next =>action => {
3. // This gets called for every action you dispatch.
4. // If it's a function, call it.
5. If ( typeOf action === `function' ) {
6. Return action (dispatch, getState, extraArgument);
7. }
8. // Otherwise, just continue processing this action as usual.
9. Return next (action);
10. };
11. }
12. Const thunk = createThunkMiddleware ();
13. Thunk. withExtraArgument = createThunkMiddleware;
14. Export default thunk;
```

Function createThunkMiddleware (extraArgument) {

31. Is there a Repository URL of ReactJS? If yes, name one.

https://github.com/reactjs/reactjs.org

32. What is the current stable version of ReactJS?

The latest stable version of ReactJS is 16.3.1, released on March 19, 2020.

33. Name a few drawbacks of using ReactJS.

Besides having several advantages like allowing the use or reuse of customizable components and the efficiencies laid out by the Virtual DOM, ReactJS also poses several inadequacies to the user. E.g.,

- Old ReactJS features are not compatible with new updates
- Lack of documentation and conventions to provide resourceful updates on ReactJS.

34. Explain the process of embedding two components into one.

Here's the code snippet for embedding two react components into one:



35. What is Flux? Compare MVC with Flux.

Flux is an application architecture with a unidirectional data flow developed and used by Facebook to build client-side web applications. It can be embedded in JS frameworks. The source of flux implementation is available in Redux libraries.

It has 4 major components. These include:

- i) Action
- ii) Dispatcher
- iii) Store
- iv) View

MVC vs. Flux.

- MVC (or Model-View-Controller) is also an application architecture similar to Flux but is used to develop UI unlike just the client-side web app (in case of Flux).
- Unlike Flux, MVC is a bi-directional communication pattern.
- In MVC the data binding during the controller phase plays a significant role in building UIs. However, Flux components rely on the actions or events phase to provide data to dispatch.
- MVC corresponds to a synchronous mechanism while Flux is an asynchronous system.
- On the other hand, Flux architecture is easy to maintain, scale, and test when compared with MVC.

36. In which ReactJS lifecycle event is AJAX request(s) made?

In the 'componentDidMount' lifecycle phase.

37. List out at least three reasons behind the success of ReactJS.

Some of the best-known advantages of using Reactjs:

i) Easy to manage state changes (from Parent to child components). Due to its uni-directional flow of data, once a state is changed for a parent component, the changes are reflected on the child component as well.

ii) Enables the reuse of components easy, highly recommended for future web development requirements

iii) Quick to learn API that can be used to build VRs as well as Mobile Apps

38. Explain what is meant by event handling?

In ReactJS, the need for event handling arises while calling a function and rendering an action when an element is clicked, React events are named using *camelCase* and not lower case. In JSX, a function is passed as an event handler instead of a string.

39. Why is JSX unreadable to browsers?

JSX (or JavaScript Extension) is a syntax that combines JavaScript and HTML. Since the accepted readable syntax format is JavaScript and in JSX it appears as HTML codes, it becomes unreadable to web browsers. To convert it to a readable format, it must be passed through a transpiler like Babel.

40. What if the expressions contain more than one line? Explain how you will perform a task in this case?

It is true that in Reactjs rendering more than one expression is not possible until the components are embedded and wrapped within curly braces. One of the many methods that can be applied to embed multiple expressions into one is the 'inline if/else method with conditional operator'.



41. Define a React Router.

Routing enables a user to navigate from one link to another on a web browser, i.e., through the browser to address bar and navigation buttons. Hence, in simple terms, React Router (latest release – version 4) is an API and a 3rd party library that helps the user switch between the pages.

42. Compare how a React Router and its techniques differ from conventional JS routers such as Backbone Router.

Backbone routing corresponds to the conventional routing mechanism. Unlike Backbone Routing, in react routing one user gets to enjoy a single page application (SPA) view and use and gets directed to the homepage while toggling between log out and login buttons. Also, backbone routers cannot process query params when compared to react-router-dom which can help the user access query params or complete route strings.

43. What are stateless components?

Also known as functional components are plain JavaScript functions and are used as mediums to store a state in an external Redux store. These are adopted by presentational components and are relevant to the UI instead of state behavior.

44. What is meant by DOM diffing?

After a JSX element has been rendered the updates are visible on the real DOM objects. Manipulating real DOM objects is a time-consuming process and these changes are first applied to a virtual DOM (a faster and easier process). The changes are then viewed on a preupdate version and only the necessary changes are then applied to the real DOM object. This process of viewing changes on a virtual DOM pre-update version is known as "**diffing**".

45. Can React restrict certain types of Props from existing while helping other Props to survive?

In Reactjs, a child component cannot pass data (or props) to a parent component since it is a unidirectional flow of data from parent to child component. Hence, React only allows the props (or functions) from a parent component to the child, preventing the latter to pass data to a parent component.

46. **Define Prop drilling and can it be avoided?**

The process of passing data (props) from one component to the other is a top-down approach to React. This process is called '**prop-drilling**'. In a normal scenario, prop-drilling doesn't allow the user to skip the middle components to pass the props to the required component.

However, the help of **'ContextAPI'** lets you bypass a few components in the component tree and pass the data to the appropriate component directly. Another solution that can be used to avoid prop-drilling is '**Render Props**'. In this process, an item component is declared in another item component enabling the data to pass directly from one component to another.

47. Explain how you will choose between Angular (JavaScript Framework) and React (JavaScript Library) while working on a project.

Both platforms have several advantages and a few disadvantages. When on one hand, **Angular** (by Google) is a preferred choice for the Enterprise App Development Services and has upgraded error-handling features, **React** library (by Facebook) offers the user the flexibility and freedom to choose tools, libraries as well the architecture to help build an app. Another advantage of the React library is the virtual DOM feature that allows the user to track and change components in different nodes on a virtual mode without affecting the entire component tree of a real DOM.

Hence, the React library can opt over Angular while working on a project (this is an opinionbased reaction. Depending on the project complexity a different framework or library can be chosen).

48. What is Automated tooling and how can it be used to boost the accessibility of ReactJS?

There is a misnomer that Reacts apps are inaccessible from the engineering perspective. Of course! It is untrue. While designing web apps, the use of assistive technology tops the list for ease of interpretation by the users. With the help of automated accessibility tools, web app developers run tests and audit to boost the accessibility of these apps.

Some examples of automated tooling are used.

Testing and boosting browser accessibility include **aXe** (or the Accessibility Engine), **aXe-core** (for end-to-end accessibility tests), **react-ax** (used during debugging, or developing the apps as the accessibility findings are introduced to the console), **WebAIM WAVE** (web accessibility evaluation tool for extension browsers).

49. How will you fix the performance of React applications if the speed at which a page is loading is very slow?

Even though React is better known for its performance, it may experience certain bottlenecks and performance issues. React app's performance hiccups can occur

- During the production stage, the app's performance is much slower than the anticipated run after the app's launch.
- When the CPU throttling option under the 'Performance' tab is not set to 4x.
- Also, the 'checkbox' option in React DevTools is not checked and some of the components undergo re-rendering
- When the latest React version ≥ 16.5 is not being used, etc.

50. Why class methods must be bound to a class instance. How can you avoid binding the two?

Binding class components help developers access state, props, or other class members.

We can skip the use of ".bind" for binding a function or class component to React. It can be done by implementing ".this". Also, there have been instances where arrow functions fail to bind a class function to class instances and throw errors. This can be avoided by passing the class function using the "SomeClass.handleInputChange" for binding event handlers.

React Js Interview Questions PDF Conclusion

Planning and preparation have a huge role in how well an interviewee performs. And if that is combined with the appropriate queries, such as these react interview questions for the senior developer, then the likelihood of passing the interview increases.

Researching about the company or industry along with a thorough knowledge of the topic in question adds value as well.

ReactJS is quite an intriguing concept, a fundamental prerequisite for students to learn and practice and some of these ideal **react interview questions** can certainly help them to fetch a career in front-end development.

What other topics or trends would you like to learn about? We'd like to hear all about them in the comments!



🖉 Last updated on August 3, 2022