TekSlate
Watch Learn & Excel

• Interview Questions Article

ReactJS Interview Questions

By admin November 29, 2017 Total Views: 10,203

Enroll Now

LEAVE A MESSAGE

ReactJS Interview Questions And Answers

Q: What is React?

React is a front end JavaScript library developed by Facebook in 2011. It follows the component based approach which helps in building reusable UI components. It is used for developing complex and interactive web and mobile UI. Even though, it was open-sourced only in 2015, it has one of the largest communities supporting it.

Q: What are the features of React?

JSX: JSX is JavaScript syntax extension.
Components : React is all about components.
One direction flow: React implements one way data flow which makes it easy to reason about your app

Q: List some of the major advantages of React.

Some of the major advantages of React are:

- It increases the application's performance
- It can be used on client and server side
- Code's readability increases, because of JSX.
- It is easy to integrate with other frameworks such as Angular, Meteor etc
- Using React, writing UI test cases become extremely easy
- React uses virtual DOM which is JavaScript object.
- This will improve apps performance
- Component and Data patterns improve readability.

Q: What are the limitations of React?

Limitations of React are listed below:

- React is just a library, not a full-blown framework
- Its library is very large and takes time to understand
- It can be little difficult for the novice programmers to understand
- Coding gets complex as it uses inline templating and JSX

Q: What is JSX?

JSX is a shorthand for JavaScript XML. This is a type of file used by React which utilizes the expressiveness of JavaScript along with HTML like template syntax. This mak

easy to understand. This file makes applications robust and boosts its performance. Below is an example of JSX:

```
render(){
    return(
        <div>
            <h1> Welcome To Tekslate!!</h1>
        </div>
    );
}
```

Q: Differentiate between Real DOM and Virtual DOM.

Real DOM	**Virtual DOM**
It updates slowly.	It updates faster.
Can directly update HTML.	Can't directly update HTML.
Creates a new DOM if element updates.	Updates the JSX if element updates.
DOM manipulation is very expensive.	DOM manipulation is very easy.
Too much of memory wastage.	No memory wastage.

Q. Real DOM vs Virtual DOM

Browsers can only read JavaScript objects but JSX in not a regular JavaScript object. Thus to enable a browser to read JSX, first, we need to transform JSX file into a JavaScript object using JSX transformers like Babel and then pass it to the browser.

Q: How is React different from Angular?

React vs Angular

TOPIC	**REACT**	**ANGULAR**
ARCHITECTURE	Only the View of MVC	Complete MVC
RENDERING	Server side rendering	Client side rendering
DOM	Uses virtual DOM	Uses real DOM
DATA BINDING	One-way data binding	Two-way data binding
DEBUGGING	Compile time debugging	Run time debug

AUTHOR	Facebook	Google

Q: Why can't browsers read JSX?

Components are the building blocks of a React application's UI. These components split up the entire UI into small independent and reusable pieces. Then it renders each of these components independent of each other without affecting the rest of the UI.

Q: Explain the purpose of render() in React.

Each React component must have a render() compulsory. If more than one HTML elements needs to be rendered, then they must be grouped together inside one enclosing tag such as <form>, <group>, etc. It returns to the single react element which is the presentation of native DOM Component. This function must be kept pure i.e., it must return the same result each time it is invoked.

Q: What is Props?

Props are short hand for Properties in React. They are read-only components which must be kept pure i.e. immutable. They are always passed down from the parent to the child components through out the application. A child component can never send a prop back to the parent component. This help in maintaining the unidirectional data flow and are generally used to render the dynamically generated data.

Q: What is a state in React and how is it used?

States are the heart of React components. States are the source of data and must be kept as simple as possible. Basically, states are the objects which determine components rendering and behavior. They are mutable unlike the props and create dynamic and interactive components. They are accessed via this.state().

React JS Interview Questions And Answers

Q: Differentiate states and props.

States vs Props

Conditions	State	Props
1. Receive initial value from parent component	Yes	Yes
2. Parent component can change value	No	Yes

3. Set default values inside component	Yes	Yes
4. Changes inside component	Yes	No
5. Set initial value for child components	Yes	Yes
6. Changes inside child components	No	Yes

Q: What is arrow function in React? How is it used?

Arrow functions are more of brief syntax for writing the function expression. They are also called 'fat arrow' (=>) the functions. These functions allow to bind the context of the components properly since in ES6 auto binding is not available by default. Arrow functions are mostly useful while working with the higher order functions.

//General way

```
render() {
    return(
        <MyInput onChange={this.handleChange.bind(this) } />
    );
}
```

//With Arrow Function

```
render() {
    return(
        <MyInput onChange={ (e) => this.handleOnChange(e) } />
    );
}
```

Q: Differentiate between stateful and stateless components.

Stateful vs Stateless Components React

Stateful Component	Stateful Component
Stores info about component's state change in memory	Stores info about component's state change in memory
Have authority to change state	Do not have the authority to change state
Contains the knowledge of past, current and possible	Contains no knowledge of

future changes in state	future state changes
Stateless components notify them about the requirement of the state change, then they send down the props to them.	They receive the props from the Stateful components and treat them as callback functions.

Q: What are the different phases of React component's lifecycle?

There are three different phases of React component's lifecycle:

Initial Rendering Phase: This is the phase when the component is about to start its life journey and make its way to the DOM.

Updating Phase: Once the component gets added to the DOM, it can potentially update and re-render only when a prop or state change occurs. That happens only in this phase.

Unmounting Phase: This is the final phase of a component's life cycle in which the component is destroyed and removed from the DOM.

Q: Explain the lifecycle methods of React components in detail.

Some of the most important lifecycle methods are:

componentWillMount() – Executed just before rendering takes place both on the client as well as server-side.

componentDidMount() – Executed on the client side only after the first render.

componentWillReceiveProps() – Invoked as soon as the props are received from the parent class and before another render is called.

shouldComponentUpdate() – Returns true or false value based on certain conditions. If you want your component to update, return true else return false. By default, it returns false.

componentWillUpdate() – Called just before rendering takes place in the DOM.

componentDidUpdate() – Called immediately after rendering takes place.

componentWillUnmount() – Called after the component is unmounted from the DOM. It is used to clear up the memory spaces.

Q: What is an event in React?

In React, events are the triggered reactions to specific actions like mouse hover, mouse click, key press, etc. Handling these events are similar to handling events in DOM elements. But there are some syntactical differences like:

Events are named using camel case instead of just using the lowercase.

Events are passed as functions instead of strings.

The event argument contains a set of properties, which are specific to an event. Each event type contains its own properties and behavior which can be accessed via its event handler only.

Q: What are synthetic events in React?

Synthetic events are the objects which act as a cross-browser wrapper around the browser's native event. They combine the behavior of different browsers into one API. This is done to make sure that the events show consistent properties across different browsers.

Q: List some of the cases when you should use Refs.

Following are the cases when refs should be used:

When you need to manage focus, select text or media playback

To trigger imperative animations

Integrate with third-party DOM libraries

Q: What do you know about controlled and uncontrolled components?

Controlled vs Uncontrolled Components

Controlled Components	Uncontrolled Components
They do not maintain their own state	They maintain their own state
Data is controlled by the parent component	Data is controlled by the DOM
They take in the current values through props and then notify the changes via callbacks.	Refs are used to get their current values

Q: What are Higher Order Components(HOC)?

Higher Order Component is an advanced way of reusing the component logic. Basically, it's a pattern that is derived from React's compositional nature. HOC are custom components which wraps another component within it. They can accept any dynamically provided child component but they won't modify or copy any behavior from their input components. You can say that HOC are 'pure' components.

Q: What can you do with HOC?

HOC can be used for many tasks like:

1. Code reuse, logic and bootstrap abstraction
2. Render High jacking
3. State abstraction and manipulation
4. Props manipulation

Q: What are Pure Components?

Pure components are the simplest and fastest components which can be written. They can replace any component which only has a render(). These components enhance the simplicity of the code and performance of the application.

Q: What is the significance of keys in React?

Keys are used for identifying unique Virtual DOM Elements with their corresponding data driving the UI. They help React to optimize the rendering by recycling all the existing elements in the DOM. These keys must be a unique number or string, using which React just reorders the elements instead of re-rendering them. This leads to increase in application's performance.

Reactjs Interview Questions For Experienced

Q: What were the major problems with MVC framework?

Following are some of the major problems with MVC framework:

- DOM manipulation was very expensive
- Applications were slow and inefficient
- There was huge memory wastage
- Because of circular dependencies, complicated model was created around models and views

Q: What is Redux?

Redux is one of the hottest libraries for front end development in today's marketplace. It is a predictable state container for JavaScript applications and is used for the entire applications state management. Applications developed with Redux are easy to test and can run in different environments showing consistent behavior.

Q: What do you understand by "Single source of truth"?

Redux uses 'Store' for storing the application's entire state at one place. So all the component's state are stored in the Store and they receive updates from the Store itself. The single state tree makes it easier to keep track of changes over time and debug or inspect the application.

Q: List down the components of Redux.

Redux is composed of the following components:

Action – It's an object that describes what happened.

Reducer – It is a place to determine how the state will change.

Store – State/ Object tree of the entire application is saved in the Store.

View – Simply displays the data provided by the Store.

Q: How are Actions defined in Redux?

Actions in React must have a type property that indicates the type of ACTION being performed. They must be defined as a String constant and you can add more properties to it as well. In Redux, actions are created using the functions called Action Creators.

Below is an example of Action and Action Creator:

```
function addTodo(text) {
    return {
        type: ADD_TODO,
        text
    }
}
```

Q: Explain the role of Reducer.

Reducers are pure functions which specify how the application's state changes in response to an ACTION. Reducers work by taking in the previous state and action, and then it returns a new state. It determines what sort of update needs to be done based on the type of the action, and then returns new values. It returns the previous state as it is, if no work needs to be done.

Q: What is the significance of Store in Redux?

A store is a JavaScript object which can hold the application's state and provide a few helper methods to access the state, dispatch actions and register listeners. The entire state/ object tree of an application is saved in a single store. As a result of this, Redux is very simple and predictable. We can pass middleware to the store to handle processing of data as well as to keep a log of various actions that change the state of stores. All the actions return a new state via reducers.

Q: How is Redux different from Flux?

Flux vs Redux

Flux	Redux

Flux	Redux
The Store contains state and change logic	Store and change logic are separate
There are multiple stores	There is only one store
All the stores are disconnected and flat	Single store with hierarchical reducers
Has singleton dispatcher	No concept of dispatcher
React components subscribe to the store	Container components utilize connect
State is mutable	State is immutable

Q: What are the advantages of Redux?

Advantages of Redux are listed below:

Predictability of outcome: Since there is always one source of truth, i.e. the store, there is no confusion about how to sync the current state with actions and other parts of the application.

Maintainability: It is simple to maintain with a strict structure and predictable outcome.

Server side rendering: To the client side, You just need to pass the store created on the server. This is helpful for initial render and provides a high quality user experience as it optimizes the application performance.

Developer tools: From actions to state changes, developers can track everything going on in the application in real time.

Community and ecosystem: Redux has a huge community behind it which makes it even more captivating to use. A large community of talented individuals contribute to the betterment of the library and develop various applications with it.

Ease of testing: Redux's code is mostly functions which are small, pure and isolated. This makes the code testable and independent.

Organization: Redux is precise about how code should be organized, this makes the code more consistent and easier when a team works with it.

Q: What is React Router?

React Router is a powerful routing library built on top of React, which helps in adding new screens and flows to the application. This keeps the URL in sync with data that's being displayed on the web page. It maintains a standardized structure and behavior and is used for developing single page web applications. React Router has a simple API.

Q: Why is switch keyword used in React Router v4?

Although a is used to encapsulate multiple routes inside the Router. The 'switch' keyword is used when you want to display only a single route to be rendered amongst the several defined routes. The <switch> tag when in use matches the typed URL with the defined routes in sequential order. When the first match is found, it renders the specified route. Thereby bypassing the remaining routes.

Q: Why do we need a Router in React?

A Router is used to define multiple routes and when a user types a specific URL, if this URL matches the path of any 'route' defined inside the router, then the user is redirected to that particular route. So basically, we need to add a Router library to our app that allows creating multiple routes with each leading to us a unique view.

```
1    <switch>
2      <route exact path='/' component={Home}/>
3      <route path='/posts/:id' component={Newpost}/>
4      <route path='/posts'   component={Post}/>
5    </switch>
```

Q: List down the advantages of React Router.

Few advantages are:

Just like how React is based on components, in React Router v4, the API is 'All About Components'. A Router can be visualized as a single root component (<BrowserRouter>) in which we enclose the specific child routes (<route>).

No need to manually set History value: In React Router v4, all we need to do is wrap our routes within the<BrowserRouter> component.

The packages are split: Three packages one each for Web, Native and Core. This supports the compact size of our application. It's easy to switch over based on similar coding style.

Q: How is React Router different from conventional routing?

Topic	Conventional Routing	React Routing
PAGES INVOLVED	Each view corresponds to a new file	Only single HTML page is involved
URL CHANGES	A HTTP request is sent to server and	Only the History att

CHANGES	corresponding HTML page is received	
FEEL	User actually navigates across different pages for each view	User is duped thinking he is navigating across different pages

Summary	
Reviewer	Jim Heck
Review Date	2017-11-29
Reviewed Item	React JS Interview Questions and Answers
Author Rating	★★★★★

INTERVIEW QUESTIONS

Meet the author

ADMIN

"At TekSlate, we are trying to create high quality tutorials and articles, if you think any information is incorrect or want to add anything to the article, please feel free to get in touch with us at info@tekslate.com, we will update the article in 24 hours."

2 Responses on ReactJS Interview Questions"

Anil Yadav says: **REPLY**

MAY 18, 2018 AT 9:23 AM

This article is really good.
I would suggest adding topics/questions on testing React Applications using Kar

Kiran says:

REPLY

JUNE 7, 2018 AT 10:52 PM

Very well descriptive Answers , Thanks !

Leave a Message

YOUR EMAIL ADDRESS WILL NOT BE PUBLISHED. Required fields are marked *

Comment

Name*

Email*

Website

POST COMMENT

Latest Tutorials >

Browse all 157 Tutorials >

JSF Tutorials

JSF Tutorials Overview Welcome to JSF Tutorials. The intent of these tutorials is to gain in

Oracle Applications Administrator Tutorials

Welcome to the free Oracle Applications Administrator Tutorials. The intent of these tutorials is to

Enroll Now

Workday Integration Tutorials

LEAVE A MESSAGE

Welcome to Workday Integration Tutorials. The objective of these tutorials is to provide in depth

MORE

Home

About

Become Instructor

Disclaimer

COURSES

Blue Prism Training

Hybris Training

Blockchain Training

VIEW ALL COURSES

TUTORIALS

Informatica Tutorials & Interview Questions

Tableau Tutorials & Interview Questions

Salesforce CRM Tutorials

VIEW ALL TUTORIALS

INTERVIEW QUESTIONS

Tableau Interview Questions And Answers

JMeter Interview Question And Answers

Maven Interview Questions

VIEW ALL QUESTIONS