Frontend Test Questions - ReactJS / NodeJS

ReactJS

1) What is React? How is it different from other JS frameworks?

Ans: 1. ReactJS is used as the V in MVC framework.

2. It breaks down parts of the view in the Components. These components have both the logic to handle the display of view and the view itself.

3. When the state changes, React conceptually hits the refresh button and knows to only update the changed parts.

How is it different from other JS frameworks:

1. It uses virtual DOM to render to instead of the actual DOM. It finds difference between the real DOM and virtual DOM and conducts the minimum number of DOM operations required to achieve the new state.

2) What is JSX?

Ans: JSX is an XML-like syntax extension it uses familiar syntax with plain HTML.

3) What is Redux? How to use it for?

Ans: Redux maintains a single state tree for the complete application and add reducers to enhance the state of the store.

How to use it for:

Step 1. Store(all applications state/data as one big JavaScript Object)-->

Step 2. Provider(makes store available to all containers)--whenever state changes components are re-rendered-->

Step 3. Components--(user interacting with app)-->

Step 4. Actions( are object which have 2 properties type and data)-->

Step 5. Reducers (take previous state nad action and returns new state)-->

Store-->step 1

4) How would you create Higher Order Components (HOCs) in React?

Ans: A higher order function is a function that when called, returns another function.

Simmilarly in React HOCs are React Component that wraps another one.

import React from 'react';

var newData = {

data: 'Data from HOC...',

}

var MyHOC = ComposedComponent ⇒ class extends React.Component {

componentDidMount() {

this.setState({

data: newData.data

});

}

render() {

return <ComposedComponent {...this.props} {...this.state} />;

}

};

class MyComponent extends React.Component {

render() {

return (

<div>

<h1>{this.props.data}</h1>

</div>

)

}

}

5. Compare and contrast setState() and forceUpdate(). What is the significance of each, and when would you

use one or the other? In addition, how might you use any data layer with React (like Backbone, Ember, or

Redux)? What are the systemic requirements of doing so?

Ans:

forceState() : If render() method reads from something other than this.props or this.state, we will need to tell React when it needs to re-run render() by calling forceUpdate(). we will also need to call forceUpdate() if we mutate this.state directly.

setState() : when we need to re-render the component setState() is use to update the state.

6. How might React handle or restrict Props to certain types, or require certain Props to exist?

Ans:

Using propTypes React handle Props to certain type.

App.propTypes = {

propNumber: React.PropTypes.number,

propString: React.PropTypes.string,

propObject: React.PropTypes.object.isrequired

}

if propNumber is not Number or propString is not a String and propObject is not present then we will have error on console.

7) ReactJS is an MVC based framework?

☐ False (its a frontend "V" in MVC library).

8) Which of the following concepts is/are key to ReactJS?

☐ Component-oriented design

☐ Event delegation model

☐ Both of the above,

Ans Both of the above

9) ReactJS focuses on which of the following part when considering MVC?

☐ M (Model)

☐ V (View)

☐ C (Controller)

Ans : V (View)

10) Which of the following needs to be updated to achieve dynamic UI updates?

☐ props

☐ state

Ans : State

11) Which of the following API is a MUST for every ReactJS component?

☐ getInitialState

☐ render

☐ renderComponent

Ans: render

12) ‘div’ defined within render methods is an actual DOM div element?

☐ True

☐ False

Ans: false

13) Which of the following is used to pass the data from parent to child?

☐ state

☐ props

Ans: props

14) A component in ReactJS could be composed of one or more inner components?

☐ True

☐ False

Ans: true

15) JSX transformer is a MUST to work with ReactJS?

☐ True

☐ False

Ans : false

16) One can define default values for properties, props, using which of the following method?

☐ getDefaultProps

☐ getPropsValue

☐ getInitialState

Ans: getDefaultProps()

17) PropTypes is used for \_\_\_\_\_?

☐ Determine properties types

☐ Validation

Ans: validation

18) Life cycle methods of a components fall under following categories?

☐ Mounting, Unmounting

☐ Mounting, Updating

☐ Mounting, Updating, Unmounting

Ans: Mounting, Updating, Unmounting

19) Which of the following mounting methods is/are invoked before the component is inserted into DOM?

☐ getInitialState

☐ getInitialState, componentWillMount

☐ getInitialState, componentDidMount

☐ componentWillMount

Ans: getInitialState, componentWillMount

20) Which of the following methods change the state of the component?

☐ forceUpdate

☐ setState

☐ Both of the above

☐ None of the above

Ans: setState

21) To use native javascript as an attribute value, the expression should be wrapped within \_\_\_\_\_?

☐ curly-prances{}

☐ double-quotes

Ans: {}

22) Which of the following is used to trigger a UI update?

☐ props

☐ state

Ans: state

23) ReactJS renders HTML tags if the element is defined in \_\_\_\_\_\_

☐ Lowercase

☐ Uppercase

Ans: Lowercase

24) Data flow from owner to owned components using which of the following?

☐ state

☐ props

Ans: props

25) Change or update to large number of nodes is optimized using which of the following techniques?

☐ Batching

☐ Change detection

☐ Both of the above

Ans: Change detection

NodeJs:

26) What is your favourite Javascript Framework/Boilerplate and why?

Ans: React, It provides a component layer that makes it easier to make UI elements and combine them together. It abstracts the DOM away so that it optimizes rendering and allows to render React from node.js. Further, it implements a one-way reactive data flow that makes it easier to understand and use than other frameworks.

27) What is ‘Callback’ in NodeJS?

Ans: A callback function is simply a function you pass into another function so that function can call it at a later time.

28) Explain NPM and how you an you install a module in NodeJS?

Ans: NPM is Node Package Manager for node applications, That can be used to install dependencies and publish new modules.

Installing a module in node:

npm install <Module Name>

29) What are the popular modules of NPM in Node.js?

Ans: express, socket.io, react, nodemon

30) Explain Package.JSON?

Ans: A package.json file is meta data file about app or module.It includes the list of dependencies to install from npm when running npm install.

npm init

npm install --save

31) What is the purpose of Node.js module.export and how do you use it?

Ans: module.export is used to return code of a file or unit that can be used in other files using require or import.

32) Which of the following code gets length of a buffer buf?

☐ buf.length

☐ buf.size

☐ buf.length()

☐ buf.size()

Ans: buf.length

33) Which of the following is true about Piping streams?

☐ Piping is a mechanism where we provide output of one stream as the input to another stream

☐ Piping is normally used to get data from one stream and to pass output of that stream to another

stream

☐ There is no limit on piping operations

☐ All of the above

Ans: All of the above

34) Which of the following is true about RESTful webservices?

☐ Webservices based on REST Architecture are known as RESTful web services

☐ Webservices uses HTTP methods to implement the concept of REST architecture

☐ Both of the above

☐ None of the above

Ans: Both of the above

35) 1In which of the following areas, Node.js is perfect to use?

☐ I/O bound Applications

☐ Data Streaming Applications

☐ Data Intensive Realtime Applications (DIRT)

☐ All of the above

Ans: All of the above

36) By default, npm installs any dependency in the global mode.

☐ True

☐ False

Ans: False, we need to use -g flag

37) Which of the following code can make a request to a web server?

☐ http.request(options, callback)

☐ http.createRequest(options, callback)

☐ http.makeRequest(options, callback)

☐ None of the above

Ans: http.request(options, callback)

38) Why code written in Node.JS is pretty fast although being written in JavaScript?

☐ Node.JS internally converts JavaScript code to Java based code and then execute the same.

☐ Node.JS internally converts JavaScript code to C based code and then execute the same.

☐ Being built on Google Chrome's V8 JavaScript Engine.

Ans: Being built on Google Chrome's V8 JavaScript Engine.

39) Which method of fs module is used to write a file?

☐ fs.write(path, flags[, mode], callback)

☐ fs.writeFile(path, flags[, mode], callback)

☐ fs.writePath(path, flags[, mode], callback)

Ans: fs.writeFile(path, flags[, mode], callback)

40) Which method of fs module is used to delete a file?

☐ fs.delete(fd, len, callback)

☐ fs.remove(fd, len, callback)

☐ fs.unlink(path, callback)

Ans: fs.unlink(path, callback)

41) Which of the following module is required for exception handling in Node?

☐ web module

☐ net module

☐ domain module

☐ error module

Ans: error module

42) Node js is a single threaded application but supports concurrency.

☐ True

☐ False

Ans: True