



Full Name:	Kenneth Choi
Email:	kennethchoi@gmail.com
Test Name:	MBA: React/Redux
Taken On:	1 Aug 2019 17:40:06 PDT
Time Taken:	14 min 7 sec/ 25 min
Work Experience:	1 years
Invited by:	Jeff
Invited on:	1 Aug 2019 17:35:54 PDT
Tags Score:	<div>Advanced15/25</div> <div>Essential76.25/95</div> <div>Flux0/15</div> <div>JSX5/5</div> <div>Javascript15/15</div> <div>React Class Components47.5/55</div> <div>React Functional Components10/15</div> <div>React Lifecycle Methods15/15</div> <div>React Render20/30</div> <div>React State26.25/30</div> <div>ReactJS72.5/85</div> <div>Redux28.75/30</div> <div>Redux Actions5/5</div> <div>Redux Reducers5/5</div> <div>Redux Thunk5/5</div>

76%

91/120

scored in **MBA: React/Redux** in
14 min 7 sec on 1 Aug 2019
17:40:06 PDT


Recruiter/Team Comments:

No Comments.

	Question Description	Time Taken	Score	Status
Q1	JSX > Multiple Choice	38 sec	5/ 5	✓
Q2	React Rendering > Multiple Choice	42 sec	5/ 5	✓
Q3	React Debugging > Multiple Choice	1 min 1 sec	5/ 5	✓
Q4	React Lifecycle Methods > Multiple Choice	16 sec	5/ 5	✓
Q5	React Components > Multiple Choice	8 sec	5/ 5	✓
Q6	React State > Multiple Choice	20 sec	5/ 5	✓

Q7	React Lifecycle Methods > Multiple Choice	8 sec	5/ 5	✓
Q8	React Props > Multiple Choice	2 min 30 sec	0/ 5	✗
Q9	Redux Actions and Reducers > Multiple Choice	30 sec	5/ 5	✓
Q10	Redux Thunk > Multiple Choice	23 sec	5/ 5	✓
Q11	Redux > Multiple Choice	20 sec	5/ 5	✓
Q12	React Component Methods > Multiple Choice	35 sec	5/ 5	✓
Q13	React Diffing > Multiple Choice	1 min 8 sec	0/ 5	✗
Q14	React Lifecycle Methods > Multiple Choice	1 sec	5/ 5	✓
Q15	Flux > Sentence Completion	2 min 14 sec	0/ 15	✗
Q16	React Diffing > Multiple Choice	6 sec	5/ 5	✓
Q17	Redux Store > Multiple Choice	54 sec	3.75/ 5	⚠
Q18	Redux State Shape > Multiple Choice	38 sec	5/ 5	✓
Q19	React > Multiple Choice	42 sec	5/ 5	✓
Q20	React Best Practices > Multiple Choice	16 sec	5/ 5	✓
Q21	React Stateful Components > Multiple Choice	25 sec	2.5/ 5	⚠
Q22	Redux > Multiple Choice		5/ 5	✓

QUESTION 1

Correct Answer

Score 5

JSX > Multiple Choice

Javascript

ReactJS

JSX

React Render

Advanced

QUESTION DESCRIPTION

Given the following JSX snippet **that renders input from a user's form**:

```
// ...

render (
  <p>{userInput}</p>
);

// ...
```

This code is _____ .

CANDIDATE ANSWER


Options: (Expected answer indicated with a tick)

☐ unsafe, because a user can write malicious code, which makes the page vulnerable to XSS attacks

☒ safe, because JSX escapes any embedded values before rendering them

No Comments

QUESTION 2

Correct Answer

Score 5

React Rendering > Multiple Choice

ReactJS

React Render

Essential

QUESTION DESCRIPTION

Every time the **render** method in a react component is called, _____.

CANDIDATE ANSWER

Options: (Expected answer indicated with a tick)

☐ every DOM element on the entire page will be re-painted

☐ every DOM element referenced by the React component will be re-painted

☒ only DOM elements with updated content will be re-painted

☐ DOM elements with updated content and any DOM elements after the updated elements will be re-painted

No Comments

QUESTION 3



Correct Answer

Score 5

React Debugging > Multiple Choice

ReactJS

React Class Components

Advanced

QUESTION DESCRIPTION

The following component should show a number and a button. When a user clicks the button, the number should increment. In its current state, the component is throwing an error:

What is wrong with this component?

```
import React, { Component } from 'react';

class Counter extends Component {
  constructor() {
    super();
    this.state = {
      num: 0,
    };
  }

  handleClick() {
    const num = this.state.num + 1;
    this.setState({ num });
  }

  render() {
    return (
      <div>
        <h1>{ this.state.num }</h1>
        <button onClick={this.handleClick()}>Add 1</button>
      </div>
    );
  }
}

export default Counter;
```


CANDIDATE ANSWER

Options: (Expected answer indicated with a tick)

- ☐ A. There is no 'componentDidCatch()' to prevent errors in the component.
- ☐ B. 'handleClick()' should be updating props instead of state.
- ☐ C. 'handleClick()' is not bound to the instance of the component.
- ☐ D. 'handleClick()' is being called on every render, which triggers a re-render, resulting in an infinite loop.
- ☒ E. Both C and D need fixed before this component will work.

No Comments

QUESTION 4


Correct Answer

Score 5

React Lifecycle Methods > Multiple Choice

Javascript

ReactJS

React Class Components

React Lifecycle Methods

Essential

QUESTION DESCRIPTION

The React lifecycle method, `componentDidUpdate(oldProps, oldState, snapshot)` _____.


CANDIDATE ANSWER

Options: (Expected answer indicated with a tick)

- ☐ can be used to prevent a component from mounting.
- ☐ can be used to prevent a component from updating.
- ☐ is called immediately after updating occurs, including the initial render.
- ☒ is called immediately after updating occurs, but is not called on the initial render.

No Comments

QUESTION 5


Correct Answer

Score 5

React Components > Multiple Choice

ReactJS

React Class Components

React Functional Components

Essential

QUESTION DESCRIPTION

Functional React components are preferred over Class components when a component _____.

CANDIDATE ANSWER

Options: (Expected answer indicated with a tick)

- ☐ has user input which is tracked in its state
- ☒ is purely presentational
- ☐ needs to keep track of updating information in state
- ☐ needs to have access to React lifecycle methods

No Comments

QUESTION 6



Correct Answer

Score 5

React State > Multiple Choice

ReactJS

React Class Components

React State

Essential

QUESTION DESCRIPTION


Given the following react component, **what is wrong with the onClick function in the button?**

```
class Example extends React.Component {
  constructor() {
    super();
    this.state = {
      color: 'orange'
    };
  }

  render() {
    return (
      <div>
        <h1>Color: {this.state.color}</h1>
        <button
          onClick={() => (this.state.color = 'blue')}>
          Change color to blue
        </button>
      </div>
    );
  }
}
```

CANDIDATE ANSWER

Options: (Expected answer indicated with a tick)

- ☐ Nothing. The onClick function will update the state and trigger a re-render.
- ☐ Context. The onClick function needs to be bound to the instance of the class inside the constructor.
- ☐ Wrong variable. It should be `this.props.color = 'blue'`.
- ☒  Mutating state. It should call `this.setState` to update the state.

No Comments

QUESTION 7

Correct Answer

Score 5

React Lifecycle Methods > Multiple Choice

ReactJS

React Lifecycle Methods

React Class Components

Essential

QUESTION DESCRIPTION

The lifecycle method, componentDidMount(), _____.

CANDIDATE ANSWER

Options: (Expected answer indicated with a tick)

- ☐ will be called every time the component re-renders.
- ☒ is only called after a component has finished mounting, but not when it re-renders.
- ☐ is called after a component has finished mounting AND when it re-renders.
- ☐ is called when updated props are passed down to a component.

No Comments

QUESTION 8



Wrong Answer

Score 0

React Props > Multiple Choice

ReactJS

React Render

React Class Components

Advanced

QUESTION DESCRIPTION

Given the following React components:

```
// color-title.jsx

import React from 'react';

class ColorTitle extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
      color: this.props.color
    };
  }

  render() {
    return (
      <h1>Color: {this.state.color}</h1>
    );
  }
}

export default ColorTitle;
```

Upon the initial render, the parent container passes down the following props: **{ color: 'blue' }**, and the **<h1>** inside of the ColorTitle component will have the text "Color: blue". Assuming that the ColorTitle component is **NOT** unmounted, **what color will the <h1> text be after the new props: { color: 'yellow' } are passed down, and why?**

CANDIDATE ANSWER

Options: (Expected answer indicated with a tick)

- ☐ "Color: yellow", because changing the state of the parent component will re-render its children, causing the ColorTitle to un-mount and re-mount with the updated props.
- ☒ "Color: yellow", because passing down new props into the ColorTitle component will update its state.
- ☐ "Color: blue", because 'this.changeColor' needs to be bound to the class instance in the constructor.
- ☒ "Color: blue", because passing down new props into the ColorTitle component will not update its state.

No Comments

QUESTION 9

Correct Answer

Score 5

Redux Actions and Reducers > Multiple Choice

Redux

Redux Actions

Redux Reducers

Essential

QUESTION DESCRIPTION


When an action is dispatched, what happens in regards to the reducers?

```
// Example Action

const exampleAction = {
  type: "DO_THINGS",
  payload: { key: 'val' },
};
```


CANDIDATE ANSWER

Options: (Expected answer indicated with a tick)

- ☐ The action is deconstructed and compared with each reducer's state until there is a match for the data contained in the payload.
- ☐ The action is sent to the components' `this.setState()` method.
- ☐ The action is sent **ONLY** to the reducer with a case that matches the action's ``type``.
- ☒  The action is sent to **EVERY** reducer, but new slices of state are only returned for reducers that are designed to match the action's ``type``.

No Comments

QUESTION 10

Correct Answer

Score 5

Redux Thunk > Multiple Choice

ReduxRedux ThunkEssential

QUESTION DESCRIPTION

The **Thunk Middleware** for redux makes it easier for developers to handle asynchronous requests for data. At it's core, **how does the Thunk Middleware work?**

CANDIDATE ANSWER

Options: (Expected answer indicated with a tick)

☒

☐

The Thunk Middleware intercepts everything that is dispatched. If the data that is being dispatched is a function, that function is invoked and passed the 'dispatch' method (to be called when that function resolves). If the data is a plain JavaScript object, that object is sent straight to the reducer.

☐

The Thunk Middleware sends out a 'thunk' event listener to the middleware handler of the Redux store. When the event listener is triggered, the AJAX request is sent to the reducer.

☐


The Thunk Middleware automatically creates and dispatches AJAX requests.

☐

The Thunk pushes the middleware into the event loop, which executes the action asynchronously. When the request returns, the Thunk logs the response and updates the Redux store.

No Comments

QUESTION 11

Correct Answer

Score 5

Redux > Multiple Choice

ReactJSReduxEssential

QUESTION DESCRIPTION

How does a component get access to data in the Redux store?

CANDIDATE ANSWER

Options: (Expected answer indicated with a tick)

☒

☐

The component should be turned into a 'connected component' using a Redux container.

☐

The component automatically has access to the entire Redux store.

☐

The component needs to be wrapped in a 'Provider'.

☐

A parent component needs to pass data down from its local state as props.

No Comments

QUESTION 12



Correct Answer

Score 5

React Component Methods > Multiple Choice

ReactJS

React Class Components

React State

Essential

QUESTION DESCRIPTION

The React component method, **this.setState()**, _____.

CANDIDATE ANSWER

Options: (Expected answer indicated with a tick)

- ☐ immediately changes a components state and forces the component to re-render.
- ☐ is a synchronous method.
- ☒ enqueues a request to update the state and re-render the component, which may be delayed and batched with other requests.
- ☐ will not cause a re-render if the new state is the same as the old state.

No Comments

QUESTION 13



Wrong Answer

Score 0

React Diffing > Multiple Choice

ReactJS

React Functional Components

React Render

Advanced

QUESTION DESCRIPTION

Given the following react components:

```
// parent.jsx

import React from 'react';
import MyFootercomponent from './my_footer_component';

const Parent = ({ num }) => (
  <article>
    <heading>
      <h1>Title: </h1>
      <p>{num}</p>
    </heading>
    <p>
      Numbers are very important....
    </p>
    <MyFooterComponent />
  </article>
)

export default Parent;
```

```
// my_footer_component.jsx

export default () => (
  <p>Words are also important...</p>
);
```

When a new **num** is passed down as props to the Parent component, **what happens to the <MyFooterComponent /> component?**


CANDIDATE ANSWER

Options: (Expected answer indicated with a tick)

- ☐ <MyFooterComponent /> WILL unmount, re-mount, re-render, and the coinciding DOM element WILL be re-painted.
- ☐ <MyFooterComponent /> WILL re-render and the coinciding DOM element WILL be re-painted.
- ☒ <MyFooterComponent /> WILL re-render, but the coinciding DOM element WILL NOT be re-painted.
- ☐ <MyFooterComponent /> WILL NOT re-render, and the coinciding DOM element WILL NOT be re-painted.

No Comments

QUESTION 14



Correct Answer

Score 5

React Lifecycle Methods > Multiple Choice

ReactJSReact Lifecycle Methods


React Class ComponentsEssential

QUESTION DESCRIPTION

The React lifecycle method, `componentWillUpdate(nextProps, nextState)`, _____.

CANDIDATE ANSWER

Options: (Expected answer indicated with a tick)

☒ is called just before rendering after new props or state are received.


☐ is used to selectively decide when a component should re-render.

☐ is called on the initial render of a component.

☐ is primarily used for setting elements of the state, using `this.setState()`.

No Comments

QUESTION 15



Wrong Answer

Score 0

Flux > Sentence Completion

FluxEssential

QUESTION DESCRIPTION

Problem Statement

Flux is a pattern used to structure an application. Redux is an implementation of the Flux pattern but there are some major differences. Which of the following are the **differences between Redux and Flux?** (Enter only the letters below).

A. There are multiple stores

B. Actions are simple POJOs

C. The store represents the entire state of the application.

D. State can be mutated

Complete String

The first way they differ is{blank} and the second way is {blank}.

INTERNAL NOTES

15

CANDIDATE ANSWER

The first way they differ is `b` and the second way is `c`.

No Comments

QUESTION 16



Correct Answer

Score 5

React Diffing > Multiple Choice

React Render

ReactJS

Advanced

QUESTION DESCRIPTION

Given the following React component:

```
import React from 'react';
import MyFootercomponent from './MyFooterComponent';

export default ({num}) => (
  <article>
    <h2>{num}</h2>
    <p>
      Numbers are very important....
    </p>
    <MyFooterComponent />
  </article>
)
```

Every time a new **num** is passed down in the props, which elements/components will be **re-painted** in the **DOM**?

CANDIDATE ANSWER

Options: (Expected answer indicated with a tick)

- ☐ every DOM element (the <article> and its children) will be re-painted on the page
- ☐ the <h2> element and every element/component after the <h2> (<p> and <MyFooterComponent />) will be re-painted on the page
- ☒ only the <h2> element will be repainted
- ☐ the <h2> element and the <MyFooterComponent /> child component will be re-painted

No Comments

QUESTION 17

Correct Answer

Score 3.75

Redux Store > Multiple Choice

Redux

React State

Essential

QUESTION DESCRIPTION

When is it appropriate to store data in the Redux Store? (select multiple)

CANDIDATE ANSWER**Options:** (Expected answer indicated with a tick)

- ☒ ☐ When we need access to that data in multiple components on varying levels of nesting
- ☒ ☐ When we need to trigger a change in a parent component from a child component
- ☐ When handling user input from a text field
- ☒ ☐ When a pre-loaded set of information needs to be bootstrapped to multiple components
- ☒ ☐ When it is necessary to save or restore a snapshot of the all the data in the app

No Comments

QUESTION 18



Correct Answer

Score 5

Redux State Shape > Multiple Choice

Redux

React State

Essential

QUESTION DESCRIPTION

What about the below Redux state shape **does not follow best practices?** (Select Multiple)

```
{
  posts: {
    1 : {
      id: 1,
      author: { username: 'user1', name: 'User 1' },
      body: 'post1',
      comments: [
        {
          id: 1,
          author: { username: 'user2', name: 'User 2' },
          comment: 'comment1'
        },
        {
          id: 2,
          author: { username: 'user1', name: 'User 1' },
          comment: 'comment2'
        }
      ]
    }
  }
}
```

CANDIDATE ANSWER

Options: (Expected answer indicated with a tick)

- ☒ ☐ The state shape isn't flat.
- ☐ The 'posts' slice of state should return an array containing post objects.
- ☒ ☐ The 'posts' slice of state should be nested under an 'entities' key to mark it as a table in the database.
- ☒ ☐ There should additionally be a 'users' slice of state and a 'comments' slice of state to prevent duplicating data.

No Comments

QUESTION 19



Correct Answer

Score 5

React > Multiple Choice

ReactJS

Javascript

React Class Components

React Render

Essential

QUESTION DESCRIPTION

Below is a React component that displays a number and a button. Every time the button is clicked, the number should increase by 1. In its current state, it does not work as intended.

What needs to be changed in order for this component to work as planned?

```
import React, { Component } from 'react';

class Counter extends Component {
  constructor() {
    super();
    this.state = {
      num: 0,
    };
  }

  handleClick() {
    this.setState({ num: this.state.num + 1 });
  }

  render() {
    return (
      <div>
        <h1>{ this.state.num }</h1>
        <button onClick={this.handleClick}>Add 1</button>
      </div>
    );
  }
}

export default Counter;
```


CANDIDATE ANSWER

Options: (Expected answer indicated with a tick)

- ☐ handleClick() should be updating props instead of updating state.
- ☒ handleClick() needs to be bound to the instance of the class.
- ☐ Inside of handleClick(), 'this.state.num + 1' needs to be performed outside of the object being passed into 'this.setState()'
- ☐ 'onClick' should be 'onMouseDown'.

No Comments

QUESTION 20



Correct Answer

Score 5

React Best Practices > Multiple Choice

ReactJSReact Class Components


React Functional ComponentsEssential

QUESTION DESCRIPTION


In general, **why are *stateless* (or *stateless functional components*) preferred over *stateful components* for presentational components?** (Select multiple)


CANDIDATE ANSWER

Options: (Expected answer indicated with a tick)

☒ Stateless components are easier to test than stateful components.


☐ Stateless components allow for the use of lifecycle methods.

☒ Stateless components are easier to understand.

☒ Stateless components are easier to reuse.

No Comments

QUESTION 21



Correct Answer

Score 2.5

React Stateful Components > Multiple Choice

ReactJSReact Class Components


React StateEssential

QUESTION DESCRIPTION


In a stateful React component, **why is it considered *bad practice* to mutate state directly?** (Select multiple)

CANDIDATE ANSWER

Options: (Expected answer indicated with a tick)

☒ Mutating state will not trigger a re-render in the component, even if it is needed

☐ Mutating state will not change the state at all.

☐ Changes to state could be overwritten by later calls to `this.setState()`.

☐ Mutating the state resets the context of `this`.

No Comments

QUESTION 22

Correct Answer

Score 5

Redux > Multiple Choice

ReactJS

Redux

React State

Essential

QUESTION DESCRIPTION

What is the purpose of the 'connect' function in Redux?

CANDIDATE ANSWER**Options:** (Expected answer indicated with a tick)

It works with the Provider to give components access to selective parts of the Redux store and to give access to the dispatch function.



It connects different components together and allows them to trade slices of state.



It connects reducers that share common action responses.



It connects the backend to the Redux state.

No Comments