1. Stub?

Ans,

1. angular vs react

Ans.

3-bower, grunt,npm dufference

Ans,

1. virtual dom

Ans.

1. component life cycle

Ans.

1. ref

Ans.

6-senthitec event

Ans. Inside React event handlers, the event object is wrapped in a SyntheticEvent object. These objects are pooled, which means that the objects received at an event handler will be reused for other events to increase performance. This also means that accessing the event object’s properties asynchronously will be impossible since the event’s properties have been reset due to reuse.

function handleClick(event) {

setTimeout(function () {

console.log(event.target.name);

}, 1000);

}

1. redux vs file

Ans.

1. -chainning jqery

Ans.

1. webstorage in html

Ans.

8-differt loops

Ans.

1. third param reducer

Ans.

1. set timeout inside loop setstate

Ans.

1. state in render,

Ans,

1. modular js

Ans.

1. imediate executable function

Ans.

1. force update

Ans.

1. selective in redux

Ans.

15 -web worker,

Ans,

16 -pure component

Ans- class Welcome extends React.PureComponent {

render() {

return <h1>Welcome</h1>

}

}

Hello = () => {

return <h1>Hello</h1>;

}

Groww Engineering

Go to the profile of Vikas Singh

Vikas Singh

I code to bring wow experience for users

Apr 13

Stateless Component vs Pure Component

As websites getting increasingly complex and richer making use of all the improvements made over in last few years in UI technologies, we are seeing the emergence of various components. Some focusing on stability, some on readability while others have performance benefits like we use Higher Order Component(HOC) which focuses on reusability of existing component.

You will get a lot of theoretical aspect of both the Stateless and Pure components with single quick search. I will be discussing how both components compare performance wise, developer experience and how both will fit into your production project.

Just for the sake of completion, let’s see for a sec what is a Stateless and Pure component.

STATELESS COMPONENT declared as a function that has no state and returns the same markup given the same props.

A quote from the React documentation:

These components must not retain internal state, do not have backing instances, and do not have the component lifecycle methods. They are pure functional transforms of their input, with zero boilerplate. However, you may still specify .propTypes and .defaultProps by setting them as properties on the function, just as you would set them on an ES6 class.

PURE COMPONENT is one of the most significant ways to optimize React applications. The usage of Pure Component gives a considerable increase in performance because it reduces the number of render operation in the application.

Let’s look at the performance aspect of both.

class Welcome extends React.PureComponent {

render() {

return <h1>Welcome</h1>

}

}

Hello = () => {

return <h1>Hello</h1>;

}

So above there is an example of a very simple Welcome Pure Component and Hello Stateless Component. When you use these two in your Parent Component, you will see Hello will re-render whenever Parent Component will re-render but Welcome Component will not.

This is because PureComponent changes the life-cycle method shouldComponentUpdate and adds some logic to automatically check whether a re-render is required for the component. This allows a PureComponent to call method render only if it detects changes in state or props.

17 -redux middle ware

Ans.

18- pure function –

Ans. A pure function doesn’t depend on and doesn’t modify the states of variables out of its scope.

Concretely, that means a pure function always returns the same result given same parameters. Its execution doesn’t depend on the state of the system.

var values = { a: 1 };

function impureFunction ( items ) {

var b = 1;

items.a = items.a \* b + 2;

return items.a;

}

var c = impureFunction( values );

// Now `values.a` is 3, the impure function modifies it.

var values = { a: 1 };

function pureFunction ( a ) {

var b = 1;

a = a \* b + 2;

return a;

}

var c = pureFunction( values.a );

// `values.a` has not been modified, it's still 1

19- redux architecture

Ans.

20- axios, fetchGet,fetchPost

Ans.

21- about Webpack build

Ans.

22- about testing tools enzyme - explained test cases

Ans.

23- keys in reactjs - used for performance

Ans.

24- React code handling mechanism- i told about jslint

Ans.

25-