Using ES6 Spread Operator in React.

The Spread operator lets you expand an iterable like a string, object or array into its elements while the Rest operator does the inverse by reducing a set of elements into one array.

Both operators have many use cases and are used nowadays by most JavaScript developers to achieve the old JS tricks without getting things complex. The dots syntax is supported in all modern browsers which means you can use it with React with no worries. In fact, since most React developers make use of build systems and transpilers such as Babel to support latest JS features, you can safely use the dots syntax in your React code.

–Passing React Props Using the Spread Operator in JSX

You can use the ES6 Spread operator to pass props to a React component. Let's take an example for a component that expects two props.

Normaly we have:

```
function App() {
  return <Hello firstName="GrayHat" lastName="Mwenda" />;

}
```

Using Spread operator:

```
function App() {
   const props = {firstName: 'GrayHat', lastName: 'Mwenda'};

   return <Hello {...props} />;
}
```

Note that from the official Facebook react docs,

– You can spread the props attributes to pass it in JSX using the Spread operator which passes the whole props object. So when your intent is to pass the whole props object, that’s fine to use!
– Spread attributes can be useful but they also make it easy to pass unnecessary props to components that don’t care about them or to pass invalid HTML attributes to the DOM. We recommend using this syntax sparingly.

– **Using the Spread Operator with setState()** **for Setting the Nested State**.

Let’s suppose, you need to use the setState() method to update your component state. Since the setState() may work asynchronously in manu cases for performance reasons. You would need to write the following code:

```
this.setState((state) => ({
  state1: state.state1 + 1,
}))
```

The state1 variable which is part of our component state will be incremented by one.

Now, let's suppose we have a state with a nested object in our component:

```
this.state = {
  stateObj: {
    attr1: '',
    attr2: '',
  },
}
```

– **The Rest Operator in React Example**

The dots syntax can be also used to call the Rest operator in ES6. It’s mostly useful when you want to avoid using the arguments object to access the passed arguments to a function.

This is an example pattern that uses the Rest operator to create authenticated routes in React using React Router:

```
const AuthenticatedRoute = ({ ...rest }) => {
  const id = this.state;
  if (!id) {
    return <Redirect to= />;
  }
  return <Route {...rest} />;
};
// In Use
```

```
<AuthenticatedRoute
  path=′/dashboard′
  data={this.state.data}
  render={() => (
    <SomeComponent someProps={this.someProps} />
  )}
/>
```

Note that we have various ways of using spread operators in react:

– Passing props to child components,
– Using the Spread operator with the setState() method for setting the nested data
– Using the Rest operator in React.

Best Wishes.
Lux Tech Academy.