



Tutorial Link <https://course.testpad.chitkara.edu.in/tutorials/React: Importing and Exporting Files/6045a7c270ff4c89cfddb36>

TUTORIAL

React: Importing and Exporting Files

In React.js, every JavaScript file in the application is invisible to every other JavaScript file by default. Sometimes it is needed to import a file that contains components or other functions.

If we want to use a variable/class/function that's declared in a different file, then it has to be imported. To import a variable, use an import statement:

```
import <object_name> from 'path/to/file';
```

While using an import statement, if the string at the end begins with either a dot or a slash, then import will treat that string as a file path. import will follow that filepath, and import the file that it finds.

While importing a variable from a file that is not the current file, then an import statement isn't quite enough. An export statement is also needed, written in the other file, exporting the variable that is to be used in the first file.

export comes from ES6's module system, just like import does. export and import are meant to be used together, but it will be rare to see one without the other.

There are a few different ways to use export. We'll be using a style called "named exports." Here's how named exports works:

In one file, place the keyword export immediately before something that is to be exported. That something can be any top-level var, let, const, function, or class:

```
export class Example2 extends React.Component{  
  render(){
```

```

    return (
      <h1> Component in another file </h1>
    );
  }
}

```

This style of importing and exporting in JavaScript is known as “named exports.” While using named exports, always wrap the imported variables in curly braces, such as:

```
import { Example2 } from './Example2.js';
```

So let's see how to render a component from other file:

```

// Example2.js:

import React from 'react';

export class Example2 extends React.Component{
  render(){
    return (
      <h1> Component in another file </h1>
    );
  }
}

// index.js:
import React from 'react';
import ReactDOM from 'react-dom';
import { Example2 } from './Example2.js';

class Example extends React.Component {
  render() {
    return <Example2 />
  }
}

ReactDOM.render(
  <Example />,
  document.getElementById('root')
);

```

In index.js, Example2 is imported from “Example2.js”. Then in class “Example” a component of “Example2” is created and returned in the class’s render() method. In ReactDOM.render() method “Example” is rendered. Hence the output:



CodeQuotient

codequotient.com

Tutorial by codequotient.com | All rights

reserved, CodeQuotient 2024