



Consumir servicios usando React

Jefferson A. Peña Torres
jefferson.amado.pena@correounivalle.edu.co

- Consumiendo APIs (*Continuación*)
 - Fetch API
 - Axios

Consumir servicios usando React

Repasemos, operaciones asíncronas y webs APIs

1. Consumiendo API

2. Consumiendo API: Fetch API

3. Consumiendo API: Axios



Realizar tareas u operaciones mientras se carga o se obtiene información desde otra fuente de manera paralela o asíncrona es una de las ventajas que provee Javascript en sus versiones ES6 y ES7. React utiliza métodos y operaciones heredadas que permiten realizar de manera asíncronas.

Aunque lo asíncrono suena bien, existen errores que están en manos del programador como la actualización o carga de datos en un componente que ha sido desmontado.

Con Suspense y Lazy el código de una aplicación puede ser dividido en partes que se ejecuten bajo demanda o cuándo el usuario los solicite. Esta división de código se logra con algunas herramientas de empaquetado como Webpack.

Los WEB APIs son servicios que ofrecen recursos a través de internet. De estas existen varias arquitecturas que se soportan en el intercambio de información en formato HTML, XML o JSON.



Consumir servicios usando React

Consumiendo API: Superagente

1

A veces, un componente necesita obtener algunos datos desde una ubicación remota. Esta puede ser una Web API, que siga la arquitectura REST, SOAP u otras. Un servicio (aplicación en el servidor) ofrece los recursos de información que podrán ser accedidos utilizando una URI

2

3

• • •

```
componentDidMount () {  
  superagent  
    .get('/search')  
    .query({ query: 'Manny' })  
    .query({ range: '1..5' })  
    .query({ order: 'desc' })  
    .set('API-Key', 'foobar')  
    .set('Accept', 'application/json')  
    .end((err, resp) => {  
      if (!err) {  
        this.setState(  
          {someData: resp.text})  
      }  
    })  
}
```

• • •



El futuro digital
es de todos

MinTIC



Consumir servicios usando React

Consumiendo API: Fetch

1

A veces, un componente necesita obtener algunos datos desde una ubicación remota. Esta puede ser una Web API, que siga la arquitectura REST, SOAP u otras. Un servicio (aplicación en el servidor) ofrece los recursos de información que podrán ser accedidos utilizando una URI

2

3

• • •

```
componentDidMount () {  
  fetch('/api/users')  
    .then(response => response.json())  
    .then(json => this.setState(  
      { users: json.data }  
    ));  
}
```

• • •



Consumir servicios usando React

Consumiendo API: Axios

1

A veces, un componente necesita obtener algunos datos desde una ubicación remota. Esta puede ser una Web API, que siga la arquitectura REST, SOAP u otras. Un servicio (aplicación en el servidor) ofrece los recursos de información que podrán ser accedidos utilizando una URI

2

3

• • •

```
componentDidMount () {  
  axios.get('/api/users')  
    then(json => this.setState({ users: json.data }));  
    .then(json => this.setState(  
      { users: json.data }  
    ));  
}
```

• • •



Consumir servicios usando React

Tenga en cuenta

```
1  const options = {  
2    url: 'http://localhost:6000/users',  
3    method: 'POST',  
4    headers: {  
5      'Accept': 'application/json',  
6      'Content-Type': 'application/json;',  
7    },  
8    data: {  
9      first: "Jefferson",  
10     last: "Peña",  
11   }  
12 };  
13  
14 axios(options)  
15   .then(response => {  
16     console.log(response.status);  
17   });
```

Se puede utilizar *options*.
Configuración - datos

Consumir servicios usando React

Tenga en cuenta

```
const url = 'http://localhost/test.htm';
const options = {
  method: 'POST',
  headers: {
    'Accept': 'application/json',
    'Content-Type': 'application/json;'
  },
  body: JSON.stringify({
    first: "Jefferson",
    last: "Peña",
  })
};

fetch(url, options)
  .then(response => {
    console.log(response.status);
  });
```

Se puede utilizar *options*.
Configuración - datos



El futuro digital
es de todos

MinTIC



Consumir servicios usando React

Tenga en cuenta

1

```
axios.get('https://localhost:6000/users')  
  .then(response => {  
    console.log(response.data);  
  }, error => {  
    console.log(error);  
  });
```

2

3

X

```
fetch('https://localhost:6000/users')  
  .then(response => response.json())  
  .then(data => {  
    console.log(data)  
  })  
  .catch(error => console.error(error));
```




Consumir servicios usando React

Tenga en cuenta

1

```
axios.all([
  axios.get('https://localhost:6000/teachers'),
  axios.get('https://localhost:6000/tutors')
])
.then(axios.spread((obj1, obj2) => {
  console.log(obj1[0].name);
  console.log(obj1[2].name');
})));
```

2

3

X

```
Promise.all([
  fetch('https://localhost:6000/teachers'),
  fetch('https://localhost:6000/tutors')
])
.then(async([res1, res2]) => {
  const obj1 = await res1.json();
  const obj2 = await res2.json();
  console.log(obj1[0].name);
  console.log(obj1[2].name');
})
.catch(error => {
  console.log(error);
});
```



Consumir servicios usando React

Resumen

1. Consumiendo API

2. Consumiendo API: Fetch API

3. Consumiendo API: Axios

Existen varios módulos para conectarse a una API desde un proyecto o aplicación SPA frontend. Es común utilizar uno para el proyecto y realizar con este las solicitudes HTTP a los endpoints del Backend.



www.upb.edu.co/es/mision-tic
#MisiónTICSomosTodos