

Cliente y Servidor RESTful



María Monserrat
Karen Farias

Framework e Interoperabilidad 2018
FAI - UNCOMA

Temas

- REST y RESTful
- Herramientas
- ¿Qué es React?
- Componentes de React
- Ejemplo Cliente y Servidor RESTful con ReactJS
- Código
- Conclusión

¿Qué es REST?

- "REpresentational State Transfer".
- Estilo de arquitectura que proporciona estándares entre los sistemas en la WEB.
- Facilita la comunicación entre los sistemas.

¿Qué es RESTful?

- Sistemas basados en REST
- Se implementa el cliente y servidor, sin que cada uno sepa del otro.
- Cada lado si debe saber el formato de respuesta que debe enviar al otro. (Ej: JSON)

HERRAMIENTAS

Servidor

PHP

MySQL

Postman

Cliente

React

Bootstrap 4

React DevTools

React

Es una librería de JavaScript para crear interfaces de usuario dependiendo de su estado.

Brinda muchas ventajas en performance, modularidad y promueve un flujo de datos y eventos muy claro, lo que facilita la planeación y el desarrollo de apps complejas.

Versión actual **16.4**

Versión usada en el ejemplo **15.1**

React - Componentes

React permite crear componentes, estos son como funciones de JavaScript. Son elementos independientes y reutilizables.

Todos los componentes de React tienen funciones, las cuales especifican cómo deben comportarse y visualizarse.

React - Componentes

Ciclo de vida:

Cuando se invoca un componente, se espera un método de renderizado, el cual desencadena un ciclo de vida.

- **componentWillMount()** Se lanza antes de que se renderice el componente
- **componentDidMount()** Se lanza después del renderizado del componente
- **shouldComponentUpdate()** Devuelve con un valor si el componente debería actualizarse
- **componentWillUnmount()** Se lanza antes de que el componente se elimine.

Ejemplo:

Cliente y Servidor RESTful con React JS

Dominio:

Una editorial podrá acceder a través de una aplicación web a su registro personal de libros. La aplicación permite ver todos los libros registrados, así como ver, crear, editar o eliminar un libro.

La aplicación consume un servicio web de libros, el cual responde a las solicitudes del cliente.

Web Service REST - Libros (Servidor)

- ▲ wsEditorial
 - ▲ config
 - 🐘 database.php
 - ▲ libro
 - 🐘 create.php
 - 🐘 delete.php
 - 🐘 read.php
 - 🐘 unLibro.php
 - 🐘 update.php
 - ▲ objects
 - 🐘 libro.php

Funciones

```
header("Access-Control-Allow-Methods:...");  
//especifica el método aceptado cuando se accede al recurso en  
respuesta
```

```
json_decode(file_get_contents("php://input"));  
//Decodifica un string de JSON  
//Transmite un fichero completo a una cadena
```

```
json_encode(...)  
//Devuelve un string con la representación JSON
```

Postman

<https://www.getpostman.com/>

Es una herramienta para generar peticiones a APIs y crear colecciones de peticiones que nos permitan probarlas de una manera rápida y sencilla.

Se puede definir el tipo de petición (GET, POST, etc.), tokens de autenticación, cabeceras asociadas a la petición, etc.

Las peticiones se pueden exportar a múltiples, y ser usadas en nuestros archivos.

Postman - Ejemplo

The screenshot displays the Postman web interface. At the top, the address bar shows `http://localhost/editor`. The main request bar is configured with the method `GET` and the URL `http://localhost/editorialTPuno/wsEditorial/libro/unLibro.php?id=3`. The response status is `200 OK`, with a response time of `26 ms` and a size of `968 B`. The response body is shown in the `JSON` tab, displaying a JSON object with the following details:

```
1 {
2   "id": "3",
3   "nombre": "Crónica de una muerte anunciada",
4   "descripcion": "En un pequeño y aislado pueblo en la costa del Caribe, se casan Bayardo San Román, un hombre rico y recién
                    llegado, y Ángela Vicario. Al celebrar su boda, los recién casados se van a su nueva casa, y allí Bayardo descubre que su
                    esposa no es virgen. Inmediatamente, Bayardo devuelve a Ángela Vicario a la casa de sus padres donde es golpeada por su
                    madre. Ángela culpará a Santiago Nasar, un vecino del pueblo.",
5   "isbn": "32442343",
6   "autor": "Gabriel García Márquez",
7   "fecha": "2018-04-10"
8 }
```

API RESTful - Editorial (Cliente)

```
└─ reactJS-App
  └─ app
    └─ assets
      └─ css
        └─ # style.css
      └─ JS navbar.component.js
    └─ libro
      └─ JS create_libro.component.js
      └─ JS delete_libro.component.js
      └─ JS libro_table_row.component.js
      └─ JS libro_table.component.js
      └─ JS read_libro.component.js
      └─ JS top_actions.component.js
      └─ JS unLibro.component.js
      └─ JS update_libro.component.js
    └─ JS main.component.js
  └─ <> index.html
```

Funciones

```
componentDidMount: function() {  
    $('.page-header h1').text('Ver libro');  
},  
//Se lanza cuando se renderiza el componente.
```

```
componentWillUnmount: function() {  
    this.serverRequest.abort();  
},  
//Se lanza antes que el componente se elimine.
```

```
getInitialState: function() {}  
//Es parte del constructor del componente. Se indica el estado inicial  
de variables, atributos, estados.
```

React DevTools

<https://chrome.google.com/webstore/detail/react-developer-tools/fmkadmapgofadopljbjfkapdkoienihi>

Es una extensión de desarrollo para navegadores, que permite inspeccionar las jerarquías de los componentes y subcomponentes de la aplicación desarrollada con React. Podemos ver y modificar las propiedades, estados, datos, etc.

Es muy útil para entender cómo funciona React.

Conclusión

A lo largo del desarrollo de este trabajo, comprendimos que es realmente REST y RESTful, así como sus diferencias. Términos que comúnmente escuchamos o leímos, pero sin darles la importancia que se merecen. Entendimos cómo funcionan las APIs REST locales y externas, y cómo consumirlas con cualquier tipo de cliente.

Aprendimos a usar herramientas y tecnologías muy útiles como Postman, React y React Dev Tools.

Además, mientras buscábamos documentación y ejemplos, vimos muchas librerías y frameworks que despiertan la curiosidad y que deseamos probar como Express y Redux.