

Pontificia Universidad Católica Madre y Maestra



Programación web ISC-415-T

Presentado por:

Cristian Bueno

2015-1256

Iván rodríguez

2016-0773

Presentado al profesor:

Carlos Camacho

Tema:

Reporte de practica 3

Fecha de entrega:

Jueves, 25 de noviembre del 2019

Introducción

La transferencia de estado representacional (REST) es un estilo arquitectónico que define un conjunto de restricciones que se utilizarán para crear servicios web. REST API es una forma de acceder a los servicios web de una manera simple y flexible sin tener ningún procesamiento.

La tecnología REST generalmente se prefiere a la tecnología más sólida del Protocolo simple de acceso a objetos (SOAP) porque REST usa menos ancho de banda, es simple y flexible, lo que la hace más adecuada para el uso de Internet. Se utiliza para buscar o proporcionar información de un servicio web. Toda la comunicación realizada a través de la API REST utilizó solo la solicitud HTTP.

Se utilizó la practica 2 como estructura para esta práctica.

Link: <https://github.com/cristianbg11/practica2.git>

Desarrollo

Del proyecto **sparkjava-rest**, crear una cliente en el lenguaje de su preferencia que consulte las siguientes operaciones:

- Listar Todos los Estudiantes.

```
get( path: "/listar", (request, response) -> {  
    URL url = new URL( spec: "http://localhost:4567/rest/estudiantes/");  
    BufferedReader br = new BufferedReader(new InputStreamReader(url.openStream()));  
  
    JSONArray reccs = new JSONArray(br.readLine());  
  
    for (int i = 0; i < reccs.length(); ++i) {  
        Estudiante estudiante = new Estudiante();  
        JSONObject rec = reccs.getJSONObject(i);  
        estudiante.matricula = rec.getInt( key: "matricula");  
        estudiante.nombre = rec.getString( key: "nombre");  
        if(rec.isNull( key: "correo") || rec.isNull( key: "carrera")){  
            estudiante.correo = "null";  
            estudiante.carrera = "null";  
        } else {  
            estudiante.correo = rec.getString( key: "correo");  
            estudiante.carrera = rec.getString( key: "carrera");  
        }  
        estudiantes.add(estudiante);  
    }  
    response.redirect( location: "/lista");  
    return "Listado estudiantes";  
}
```

Inicio

Crear

Listar

Sign out

Matricula	Nombre	Apellido	Telefono	
14	Estudiante 14	null	null	<div>Ver</div> <div>Editar</div> <div>Eliminar</div>
15	Estudiante 15	null	null	<div>Ver</div> <div>Editar</div> <div>Eliminar</div>
16	Estudiante 16	null	null	<div>Ver</div> <div>Editar</div> <div>Eliminar</div>
17	Estudiante 17	null	null	<div>Ver</div> <div>Editar</div> <div>Eliminar</div>
18	Estudiante 18	null	null	<div>Ver</div> <div>Editar</div> <div>Eliminar</div>
19	Estudiante 19	null	null	<div>Ver</div> <div>Editar</div> <div>Eliminar</div>

- Consultar Estudiantes.

Close

Matricula

14

Nombre

Estudiante 14

Apellido

null

Telefono

null

- Crear un nuevo Estudiante.

```
function crear() {  
    var estudiante = {  
        matricula: $("#matricula").val(),  
        nombre: $("#nombre").val(),  
        correo: $("#correo").val(),  
        carrera: $("#carrera").val()  
    }  
  
    $('#target').html('enviando..');  
  
    $.ajax({  
        url: 'http://localhost:4567/rest/estudiantes/',  
        type: 'post',  
        dataType: 'json',  
        contentType: 'application/json',  
        success: function (data) {  
            $('#target').html(data.msg);  
        },  
        data: JSON.stringify(estudiante)  
    });  
}
```

Conclusión

La principal ventaja del uso de una API REST reside en la independencia que proporciona frente a cualquier consumidor, sin importar el lenguaje o plataforma con el que se acceda a ella. Esto permite que una misma API REST sea consumida por infinidad de clientes sea cual sea la naturaleza de estos y que el cambio a cualquier otro tipo de consumidor no provoque impacto alguno en ella. Esta característica proporciona fiabilidad, escalabilidad y una fácil portabilidad a cualquier otra plataforma, ya que aísla por completo al cliente del servidor.

Sólo se requiere que el intercambio de información de las respuestas se haga en un formato soportado, por lo general JSON o XML. Dicha separación entre el cliente y el servidor hace que se pueda migrar a otros servidores o bases de datos de manera transparente, siempre y cuando los datos se sigan enviando de manera correcta. Esto convierte a las APIs REST en una de las arquitecturas web más utilizadas actualmente por la flexibilidad que aportan a cualquier entorno de trabajo sea cual sea su naturaleza.

Bibliografía

Sparkjava.com. (2019). *Documentation - Spark Framework: An expressive web framework for Kotlin and Java*. [online] Available at: <http://sparkjava.com/documentation> [Accessed 25 Oct. 2019].

GitHub. (2019). aaberg/sql2o. [online] Available at: <https://github.com/aaberg/sql2o/wiki/Updates-and-inserts> [Accessed 25 Oct. 2019].

Stalla, A. (2019). *A Guide to the sql2o JDBC Wrapper / Baeldung*. [online] Baeldung. Available at: <https://www.baeldung.com/java-sql2o> [Accessed 25 Oct. 2019].

Tribalyte Technologies. (2019). ¿Qué son las APIs REST? | Tribalyte Technologies. [online] Available at: <https://tech.tribalyte.eu/blog-que-es-una-api-rest> [Accessed 4 Dec. 2019].