




Blog sobre Java EE

Estás aquí: [Inicio/Spring/Spring-REST/PostMan App con Spring REST](#)

PostMan App con Spring REST

4 diciembre, 2017 por [Cecilio Álvarez Caules](#) — [Deja un comentario](#)



Prueba gratuita

Google Cloud
opciones de
acuerdo a s

El uso de PostMan App es muy habitual en el trabajo con Servicios REST. La mayor parte de nuestras aplicaciones web publican información **en forma de recurso** para que otras aplicaciones puedan acceder a ella. Eso nos obliga a estar continuamente probando el funcionamiento de estos servicios y gestionar las llamadas **GET , POST, PUT ,DELETE etc.** Postman App es una aplicación multiplataforma que nos permite probar nuestros servicios **REST de una forma sencilla.** Vamos a crear un proyecto con Spring Boot y Spring MVC para ver las ventajas que tiene. El primer paso es definir un `@RestController` que nos devuelva una colección de Libros.

```
1 package com.arquitecturajava.web.springweb;
2
3 import java.util.ArrayList;
4 import java.util.List;
5
6 import org.springframework.http.HttpStatus;
7 import org.springframework.http.ResponseEntity;
8 import org.springframework.web.bind.annotation.GetMapping;
9 import org.springframework.web.bind.annotation.PostMapping;
10 import org.springframework.web.bind.annotation.RestController;
11
12 @RestController
13 public class ControladorLibro {
14
15     static List<Libro> lista=new ArrayList<Libro>();
16     static {
17
18         lista.add(new Libro("titulo1","autor1",200));
19         lista.add(new Libro("titulo2","autor2",200));
20
21     }
22
23     @GetMapping("/libros")
24     public List<Libro> getLibros() {
25
26
```

Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continúa navegando está dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra [política de cookies](#), pinche el enlace para mayor información.

[plugin cookies](#)[ACEPTAR](#)

```
37 }
38 }

1 package com.arquitecturajava.web.springweb;
2
3 public class Libro {
4
5     private String titulo;
6     private String autor;
7     private int paginas;
8
9     public Libro(String titulo, String autor, int paginas) {
10         super();
11         this.titulo = titulo;
12         this.autor = autor;
13         this.paginas = paginas;
14     }
15     public String getTitulo() {
16         return titulo;
17     }
18     public void setTitulo(String titulo) {
19         this.titulo = titulo;
20     }
21     public String getAutor() {
22         return autor;
23     }
24     public void setAutor(String autor) {
25         this.autor = autor;
26     }
27     public int getPaginas() {
28         return paginas;
29     }
30     public void setPaginas(int paginas) {
31         this.paginas = paginas;
32     }
33     public Libro() {
34         super();
35     }
36
37 }
```

Si ejecutamos la aplicación podemos invocar el Servicio REST y acceder a la lista en formato JSON.

Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continúa navegando está dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra [política de cookies](#), pinche el enlace para mayor información.

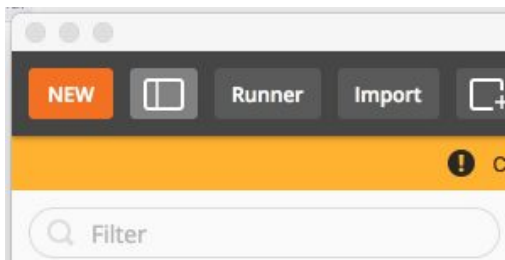
[plugin cookies](#)[ACEPTAR](#)

Utilizando PostMan App

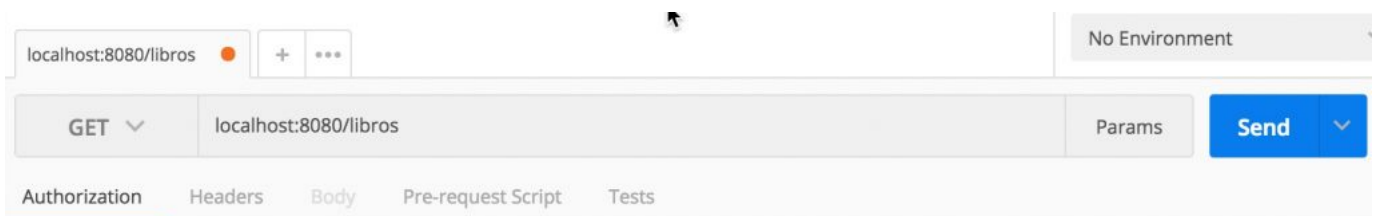
En este caso ha sido muy sencillo ejecutar el proyecto y acceder a la lista de Libros. Lamentablemente existen muchas situaciones en las que las operaciones **pueden ser más complejas**. Vamos a ver como podemos realizar la misma operación utilizando **PostMan**. Para ello el primer paso es descargar la aplicación instalarla y registrarnos



Una vez registrada la aplicación podremos acceder de forma sencilla a sus capacidades a la hora de realizar peticiones HTTP. Para ello pulsamos en el botón de new y creamos una nueva petición.



Nada más pulsar nos aparece una pestaña en donde podemos rellenar la información de la petición y ejecutarla a través del método send.

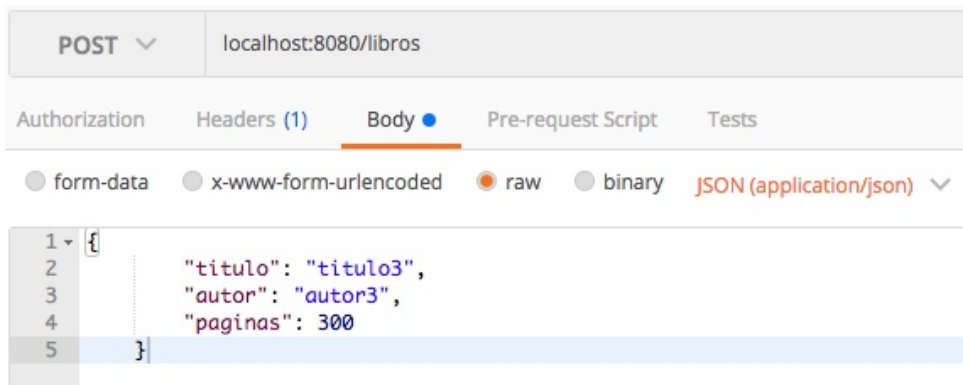


Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continúa navegando está dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra [política de cookies](#), pinche el enlace para mayor información.

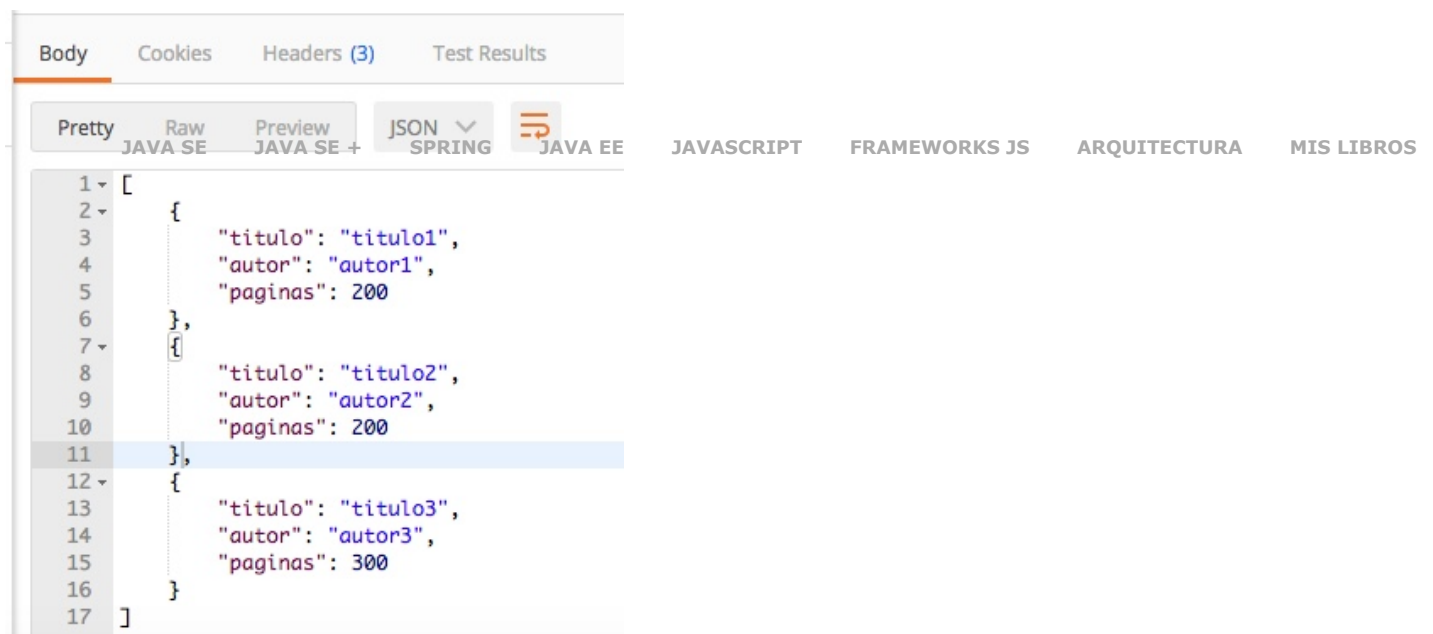
[plugin cookies](#)[ACEPTAR](#)

```
6    },
7    {
8      "titulo": "titulo2",
9      "autor": "autor2",
10     "paginas": 200
11   }
12 ]
```

Esta primera operación no aporta mucho respecto **a una petición clásica** . Sin embargo si realizamos un POST las ventajas son claras. Vamos a crear una nueva pestaña y configurar la petición en donde adjuntamos **los datos en JSON de forma muy sencilla**.



Realizamos una petición POST que recibirá nuestra aplicación de Spring MVC con muy poco trabajo. Acto seguido volvemos a realizar la primera petición GET que nos devolverá un libro adicional.



[/ spring REST SERVICE CON J2EE/REST/CONTOUR/](#)

Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continúa navegando está dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra [política de cookies](#), pinche el enlace para mayor información.

[plugin cookies](#)

ACEPTAR



in

0
COMPART**Prueba gratuita**Almacenamie
seguro, ilimita
escalable parArchivada en: [Spring-REST](#)Etiquetada con: [Arquitectura](#)

Leave a Reply

Be the First to Comment!

Notify of



Start the discussion

BUSCAR

Mis Cursos de Java Gratuitos[Java Herencia](#)[Java JDBC](#)

Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continúa navegando está dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra [política de cookies](#), pinche el enlace para mayor información.

[plugin cookies](#)[ACEPTAR](#)

POPULAR

[Maven Parent POM y uso de librerías](#)[Mis Cursos de Java para desarrolladores](#)[Java Generic Repository y JPA](#)[Spring GetMapping ,PostMapping etc](#)[Java Stream map y estadísticas](#)[El concepto de JavaScript Proxy](#)[Spring Boot WAR sin Microservicios](#)[Spring Boot Properties utilizando @Value](#)

Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continúa navegando está dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra [política de cookies](#), pinche el enlace para mayor información.

[plugin cookies](#)[ACEPTAR](#)

LO MAS LEIDO

[¿Qué es Spring Boot?](#)[PostMan App con Spring REST](#)[Usando Java Session en aplicaciones web](#)[Java Constructores this\(\) y super\(\)](#)[Java Iterator vs ForEach](#)[Spring Boot Properties utilizando @Value](#)[Introducción a Servicios REST](#)[¿Cuales son las certificaciones Java?](#)[Ejemplo de JPA , Introducción \(I\)](#)[¿Qué es Gradle?](#)[Ejemplo de Java Singleton \(Patrones y ClassLoaders\)](#)[Usando el patron factory](#)[REST JSON y Java](#)[Uso de Java Generics \(I\)](#)[Mis Libros](#)[Java Generic Repository y JPA](#)[Comparando java == vs equals](#)[Java Override y encapsulación](#)[¿Qué es un Microservicio?](#)[Spring MVC Configuración \(I\)](#)
