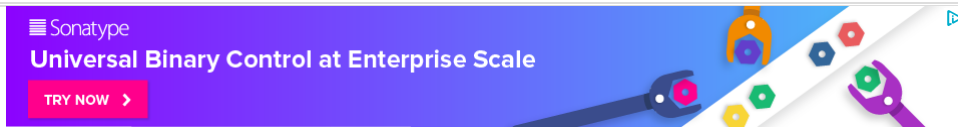


Blog sobre Java EE



Estás aquí: [Inicio](#) / [Spring](#) / [Spring-REST](#) / Spring REST CORS y su configuración

Spring REST CORS y su configuración

29 noviembre, 2018 por [Cecilio Álvarez Caules](#) — [Deja un comentario](#)

Usar **Spring REST CORS** es muy habitual hoy en día ya que la mayoría de peticiones a servicios REST que se realizan es utilizando **algún tipo de tecnología Javascript y por lo tanto utilizando AJAX**. Crear un servicio REST con Spring Framework es muy sencillo hoy en día ya que es suficiente con crear una clase anotada con `@RestController`. Vamos a verlo utilizando Spring Boot, para ello necesitaremos instalar las dependencias de Web.

```
1 <dependencies>
2   <dependency>
3     <groupId>org.springframework.boot</groupId>
4     <artifactId>spring-boot-starter-web</artifactId>
5   </dependency>
6
7   <dependency>
8     <groupId>org.springframework.boot</groupId>
9     <artifactId>spring-boot-starter-test</artifactId>
10    <scope>test</scope>
11  </dependency>
12 </dependencies>
```

Una vez instaladas estas dependencias el siguiente paso es construimos un Controlador de tipo REST que Spring nos provee por defecto con la típica url de `/mensaje`.

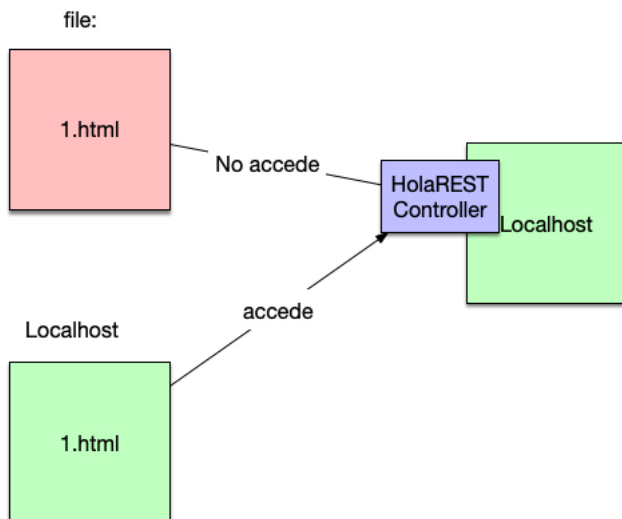
```
1 package com.arquitecturajava.rest;
2
3 import org.springframework.web.bind.annotation.GetMapping;
4 import org.springframework.web.bind.annotation.RequestMapping;
5 import org.springframework.web.bind.annotation.RestController;
6
7 @RestController
8 public class HolaRestController {
9
10    @GetMapping("/mensaje")
11    public String mensaje() {
12
13        return "hola desde spring rest";
14    }
15 }
```

Creado el controlador nos será suficiente con arrancar la aplicación de Spring Boot y solicitar la url.

El servicio REST nos responde sin ningún problema . Ahora bien esto se debe a que realizamos una invocación directa y no vía JavaScript . Si intentamos solicitar esta url con JavaScript utilizando jQuery nos encontraremos con un problema de Cross Origin Resource Sharing (CORS) que nos impide el acceso.

```
1  </pre>
2  <html>
3  <head>
4  <script type="text/javascript" src="jquery-3.3.1.min.js">
5  </script>
6  <script type="text/javascript">
7  $(document).ready(function() {
8
9  $.get("http://localhost:8080/mensaje",function(datos) {
10
11  console.log(datos);
12  })
13
14  })
15  </script>
16  </head>
17  <body>
18
19  </body>
20  </html>
21  </pre>
```

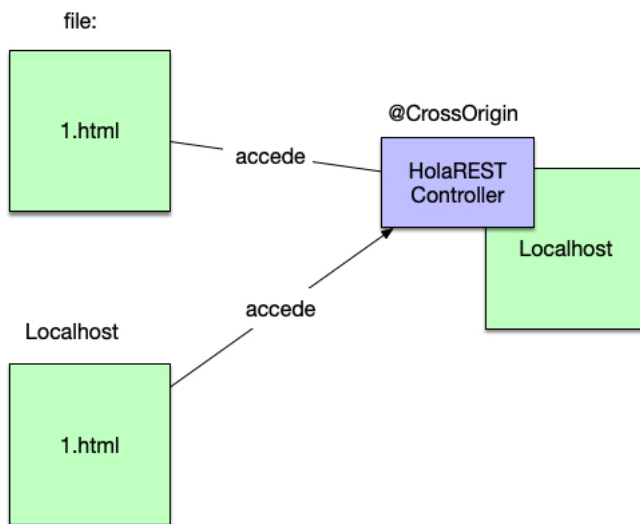
Si cargamos esta página veremos rapidamente un error en la consola que nos restringe el acceso debido a que estamos realizando una petición ajax desde JavaScript y estas peticiones por defecto **están limitadas a ficheros JavaScript que nos descarguemos desde el mismo servidor.**



Si nosotros este fichero le cargamos desde un navegador directamente con file:// nos aparecerá el siguiente error:

Spring REST CORS

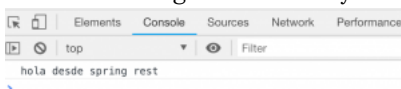
El recurso esta por defecto bloqueado para peticiones que no hagan desde localhost , caso de nuestra petición que se realiza desde un fichero directamente . Para solventar este problema es suficiente con modificar el servicio de Spring y añadir una cabecera **@CrossOrigin** para que nos permite el acceso desde otras ubicaciones.



En este caso vamos a ser generalistas y permitir acceder al recurso desde cualquier lugar.

```
1 package com.arquitecturajava.rest;
2
3 import org.springframework.web.bind.annotation.CrossOrigin;
4 import org.springframework.web.bind.annotation.GetMapping;
5 import org.springframework.web.bind.annotation.RequestMethod;
6 import org.springframework.web.bind.annotation.RestController;
7
8 @RestController
9 @CrossOrigin(origins = "*", methods= {RequestMethod.GET,RequestMethod.POST})
10 public class HolaRESTController {
11
12     @GetMapping("/mensaje")
13     public String mensaje() {
14
15         return "hola desde spring rest";
16     }
17 }
```

Volvemos a cargar el servidor y ahora si podremos acceder al mensaje almacenado en la url:



[JAVA SE](#)[SPRING](#)[JAVA EE](#)[JAVASCRIPT](#)[FRAMEWORKS JS](#)[ARQUITECTURA](#)[MIS LIBROS](#)[MIS CURSOS](#)

Otros artículos relacionados :

1. [¿Que es CORS ?](#)
2. [Spring REST Client con RestTemplates](#)
3. [Swagger documentando nuestro API REST](#)
4. [Spring REST](#)



Archivada en: [Spring-REST](#)

Deja un comentario

Tu dirección de correo electrónico no será publicada. Los campos obligatorios están marcados con *

Comentario

Nombre *

Correo electrónico *

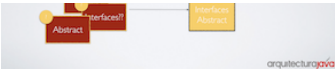
Web

PUBLICAR COMENTARIO

Este sitio usa Akismet para reducir el spam. [Aprende cómo se procesan los datos de tus comentarios.](#)

BUSCAR

Buscar en este sitio ...



Cupon 50%-TYPESCRIPT2018

Mis Cursos de Java Gratuitos

Java Herencia

Java JDBC



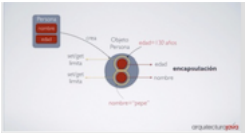
Servlets

Intro JPA

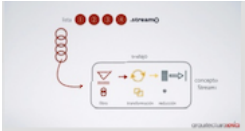


Mis Cursos de Java

Programación Orientada a Objeto en Java



Java APIS Core



Java Web



Pack Java Core



Arquitectura Java Solida con Spring



POPULAR

El patrón de inyección de dependencia y su utilidad

Framework vs Libreria dos conceptos importantes

Los Frameworks y su lado oscuro

[JAVA SE](#) [SPRING](#) [JAVA EE](#) [JAVASCRIPT](#) [FRAMEWORKS JS](#) [ARQUITECTURA](#) [MIS LIBROS](#) [MIS CURSOS](#)

[Static Method vs instance method y su uso correcto](#)

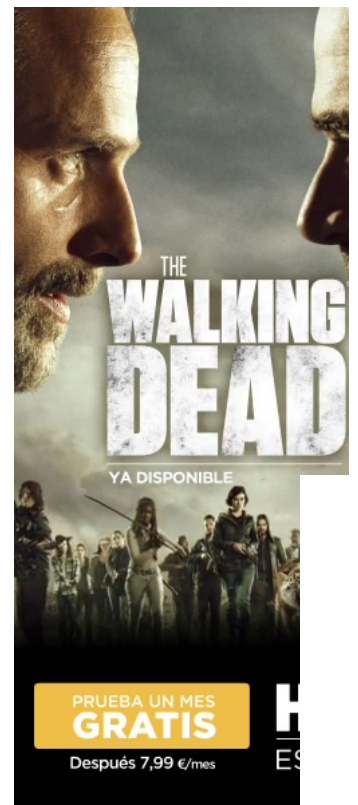
[Utilizando un JSON Generator en Java](#)

[Java new String y la creación de objetos](#)

[JPA @Basic , optimizando los fetchings](#)

CONTACTO

contacto@arquitecturajava.com



LO MAS LEIDO

[¿Qué es Spring Boot?](#)

[Spring REST CORS y su configuración](#)

[Java Constructores this\(\) y super\(\)](#)

[Usando Java Session en aplicaciones web](#)

[Ejemplo de Java Singleton \(Patrones y ClassLoaders\)](#)

[Java Iterator vs ForEach](#)

[Introducción a Servicios REST](#)

[Java Override y encapsulación](#)

[Comparando java == vs equals](#)

[Usando el patron factory](#)

[REST JSON y Java](#)

[JAVA SE](#) [SPRING](#) [JAVA EE](#) [JAVASCRIPT](#) [FRAMEWORKS JS](#) [ARQUITECTURA](#) [MIS LIBROS](#) [MIS CURSOS](#)

[Simple Web Application y REST](#)

[¿Qué es Gradle?](#)

[Black Friday en mis cursos de Java](#)

[¿Qué es un Microservicio?](#)

[Mis Libros](#)

[Spring Boot JPA y su configuración](#)

[Spring MVC Configuración \(I\)](#)

[El patrón de inyección de dependencia y su utilidad](#)

[¿ Que es REST ?](#)

[Mi Nuevo Curso de Typescript](#)

[Los Frameworks y su lado oscuro](#)

Copyright © 2018 · [eleven40 Pro Theme](#) en [Genesis Framework](#) · [WordPress](#) · [Iniciar sesión](#)