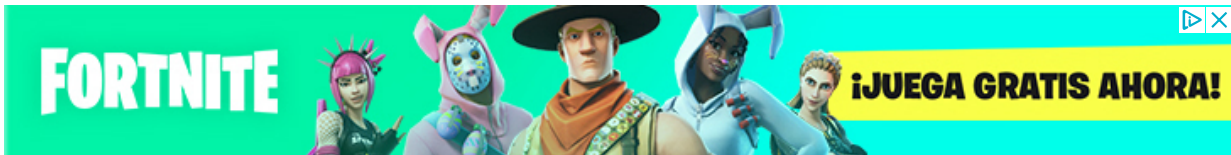


## Blog sobre Java EE

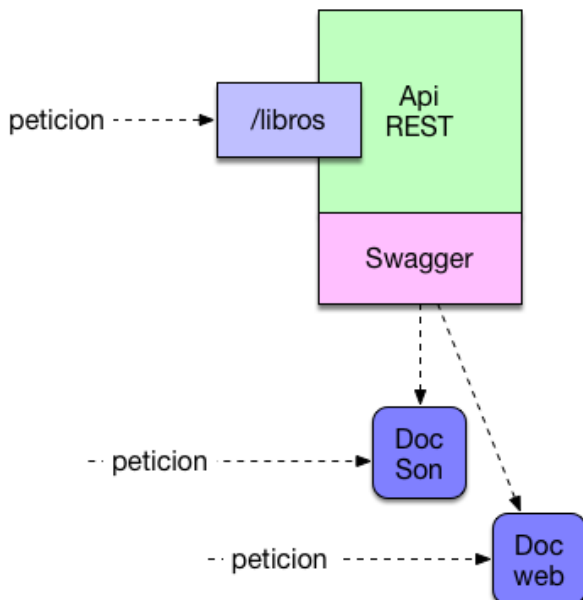
Estás aquí: [Inicio](#)/[Java EE](#)/[REST](#)/Swagger documentando nuestro API REST

# Swagger documentando nuestro API REST

1 junio, 2018 por [Cecilio Álvarez Caules](#) — 2 comentarios



**Swagger** es una herramienta que nos permite documentar de una forma sencilla **nuestras APIs REST** hoy por hoy se ha convertido en uno de los estándares de facto a la hora de trabajar. Cada día publicamos un mayor número de APIs REST y lo lógico es que publiquemos **una documentación clara de como trabajar con ellas**. Swagger nos permite hacerlo de una forma muy rápida y directa.



Vamos a ver un ejemplo, para ello partiremos de un proyecto de Spring boot que tiene un servicio que devuelve una lista de personas. Las dependencias de maven son:

JAVA SE

SPRING

JAVA EE

JAVASCRIPT

FRAMEWORKS JS

ARQUITECTURA

MIS LIBROS

MIS CURSOS

```

11     </dependency>
12
13     <dependency>
14         <groupId>io.springfox</groupId>
15         <artifactId>springfox-swagger2</artifactId>
16         <version>2.6.1</version>
17     </dependency>
18
19     <!-- https://mvnrepository.com/artifact/io.springfox/springfox-swag
20     <dependency>
21         <groupId>io.springfox</groupId>
22         <artifactId>springfox-swagger-ui</artifactId>
23         <version>2.6.1</version>
24     </dependency>
25 </dependencies>

```

## Spring @RestController

En ellas podemos ver como hemos incluido las nuevas librerías. El siguiente paso es definir un servicio REST con la anotación @RestController.

```

1  package com.arquitecturajava.rest;
2
3  import java.util.ArrayList;
4  import java.util.List;
5
6  import org.springframework.web.bind.annotation.GetMapping;
7  import org.springframework.web.bind.annotation.RestController;
8
9  @RestController
10 public class ControladorPersona {
11
12     @GetMapping("/personas")
13     public List<Persona> findAll() {
14
15         List<Persona> lista= new ArrayList<Persona>();
16         lista.add(new Persona("pepe", "perez", 25));
17         lista.add(new Persona("juan", "sanchez", 35));
18         lista.add(new Persona("ana", "gomez", 25));
19         return lista;
20     }
21
22 }

```

## Configuración de Spring

Es momento de configurar la herramienta para que nos genere una documentación del API REST. Para ello añadimos un nuevo fichero de configuración de Spring.

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.

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

JAVA SE

SPRING

JAVA EE

JAVASCRIPT

FRAMEWORKS JS

ARQUITECTURA

MIS LIBROS

MIS CURSOS

```
8 import springfox.documentation.swagger2.annotations.EnableSwagger2;
9 import springfox.documentation.builders.PathSelectors.regex;
10
11 import static springfox.documentation.builders.PathSelectors.regex;
12 @Configuration
13 @EnableSwagger2
14 public class SwaggerConfiguration {
15
16     @Bean
17     public Docket productApi() {
18         return new Docket(DocumentationType.SWAGGER_2)
19             .select()
20             .apis(RequestHandlerSelectors.basePackage("com.arquitectura
21             .paths(regex("/personas.*"))
22             .build();
23     }
24 }
25
26 }
```

Aquí configuramos swagger para que publique documentación para la url que publicamos en este caso nuestra url de /personas. Nos queda por ver el fichero de configuración principal de Spring framework que importará el de Swagger.

```
1 package com.arquitecturajava.rest;
2
3 import org.springframework.boot.SpringApplication;
4 import org.springframework.boot.autoconfigure.SpringBootApplication;
5 import org.springframework.context.annotation.Import;
6 import org.springframework.web.servlet.config.annotation.ResourceHandlerReg
7 import org.springframework.web.servlet.config.annotation.WebMvcConfigurer;
8
9 @SpringBootApplication
10 @Import(SwaggerConfiguration.class)
11 public class RestApplication implements WebMvcConfigurer{
12
13     public static void main(String[] args) {
14         SpringApplication.run(RestApplication.class, args);
15     }
16
17     @Override
18     public void addResourceHandlers(ResourceHandlerRegistry registry) {
19
20         registry.addResourceHandler("swagger-ui.html")
21             .addResourceLocations("classpath:/META-INF/resources/")
22
23     }
24 }
```

## Swagger y documentación

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](#)



JAVA SE

SPRING

JAVA EE

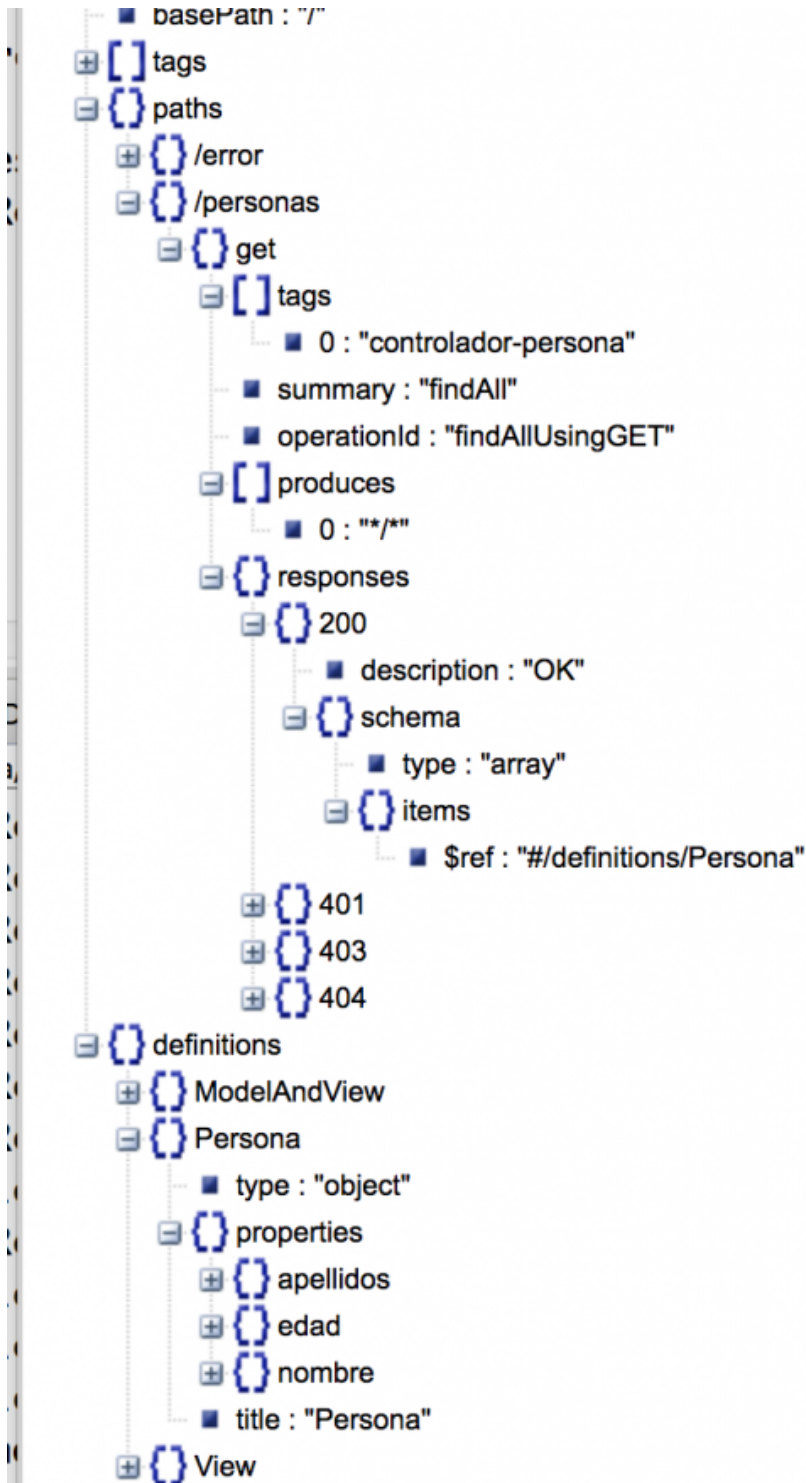
JAVASCRIPT

FRAMEWORKS JS

ARQUITECTURA

MIS LIBROS

MIS CURSOS



Podemos ver claramente **la definición de la operación y el tipo de objeto que devuelve**, así como información adicional. Otra de las opciones que tenemos es acceder a una vista web de la documentación mucho más amable **utilizando la url `/swagger-ui.html`**.

controlador-persona : Controlador Persona

[ BASE URL: / , API VERSION: 1.0 ]

Entramos a controlador persona y vemos los métodos que tiene soportados:

controlador-persona : Controlador Persona

Show/Hide | List Operations | Expand Operations

GET

/personas

findAll

Response Class (Status 200)

OK

Model

Example Value

```
[
  {
    "apellidos": "string",
    "edad": 0,
    "nombre": "string"
  }
]
```

Acabamos de **documentar nuestro api REST con swagger**.

Otros artículos relacionados

- 1. [REST HTTP return codes y sus curiosidades](#)
- 2. [Spring REST Service con @RestController](#)
- 3. [Arquitecturas REST y sus niveles](#)
- 4. [Spring Boot](#)



Archivado en: [REST](#)

[JAVA SE](#)[SPRING](#)[JAVA EE](#)[JAVASCRIPT](#)[FRAMEWORKS JS](#)[ARQUITECTURA](#)[MIS LIBROS](#)[MIS CURSOS](#)[1 junio, 2018 en 10:10](#)

Gracias por la info

[Responder](#)



Cecilio Álvarez Caules dice

[1 junio, 2018 en 22:15](#)

de nada 😊

[Responder](#)

## Deja un comentario

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

Comentario

Nombre \*

Correo electrónico \*

Web

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.

[ACEPTAR](#)

[plugin cookies](#)

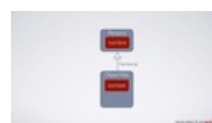


[JAVA SE](#)[SPRING](#)[JAVA EE](#)[JAVASCRIPT](#)[FRAMEWORKS JS](#)[ARQUITECTURA](#)[MIS LIBROS](#)[MIS CURSOS](#)[PUBLICAR COMENTARIO](#)

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

**BUSCAR**

**Cupón  
Descuento  
50%  
ARQUITECTURA  
JAVASPRING**

**Mis Cursos de Java Gratuitos****Java Herencia****Java JDBC****Servlets****Introduccion JPA**

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.

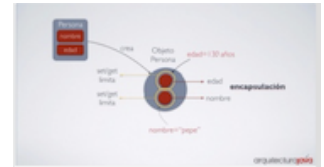
[plugin cookies](#)**ACEPTAR**



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

## Mis Cursos de Java

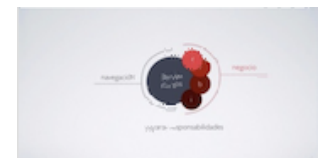
### Programación Orientada a Objeto en Java



### Java APIS Core



### Java Web



### Pack Java Core



### Arquitectura Java Solida con Spring



## POPULAR

[Arquitecturas REST y sus niveles](#)[Angular 5 Hello World y su funcionamiento](#)[Nuevo Curso:Arquitectura Java Sólida con Spring 4.3 y Anotaciones](#)

[JAVA SE](#)[SPRING](#)[JAVA EE](#)[JAVASCRIPT](#)[FRAMEWORKS JS](#)[ARQUITECTURA](#)[MIS LIBROS](#)[MIS CURSOS](#)[Spring REST - Test utilizando Rest Assured](#)[Angular ngFor la directiva y sus opciones](#)[Java Composite Pattern y recursividad](#)[Spring Web Initializer ,eliminando el web.xml](#)

## CONTACTO

[contacto@arquitecturajava.com](mailto:contacto@arquitecturajava.com)

## LO MAS LEIDO

[¿Qué es Spring Boot?](#)[Java Parallel Stream y rendimiento](#)[Java Constructores this\(\) y super\(\)](#)[Usando Java Session en aplicaciones web](#)[Java Iterator vs ForEach](#)[Swagger documentando nuestro API REST](#)[Introducción a Servicios REST](#)[Java Override y encapsulación](#)[JBCNCONF, el congreso de programadores Java más grande de España](#)[¿Cuales son las certificaciones Java?](#)[Ejemplo de Java Singleton \(Patrones y ClassLoaders\)](#)[Usando el patron factory](#)[¿Qué es Gradle?](#)[REST JSON y Java](#)[Ejemplo de JPA , Introducción \(I\)](#)[Uso de Java Generics \(I\)](#)[Comparando java == vs equals](#)

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.

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

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

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

---

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.

[ACEPTAR](#)

[plugin cookies](#)