

Blog sobre Java EE

Estás aquí: [Inicio](#)/[Arquitectura](#)/REST HTTP return codes y sus curiosidades

REST HTTP return codes y sus curiosidades

4 mayo, 2018 por [Cecilio Álvarez Caules](#) — 1 comentario

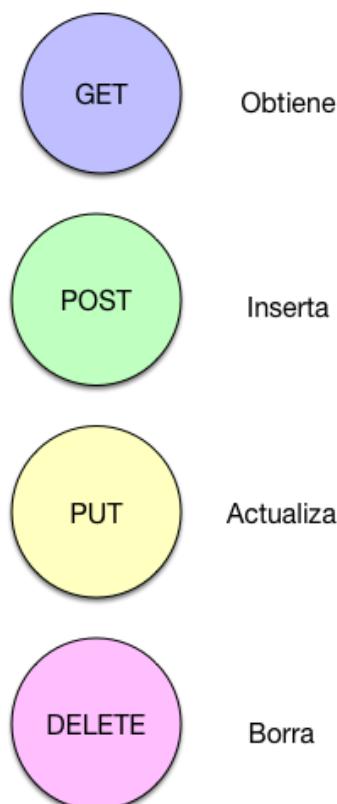
1 Test Case Management Tool Web-Based Test Management Tool to Track & Organize Test Cases. Get Free 30-Day Trial Now! [gurock.com](#)

2 Hosting para aplicaciones JAVA - Garantía de reembolso 30 días. Incluye servidor de aplicaciones dedicado: Tomcat, Glassfish, Payara o WildFly. [anw.es](#)

3 eBook: Test Automation [Free eBook] Discover 10 essential strategies for a successful test automation project! [ranorex.com](#)

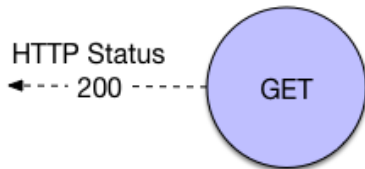
¿Cuales son los REST HTTP return codes más habituales o correctos? . Todos utilizamos hoy en día tecnologías REST y realizamos las peticiones clásicas de GET,POST, PUT y DELETE .

Practicamente todo el mundo tiene muy claro que de esta forma se define un recurso REST



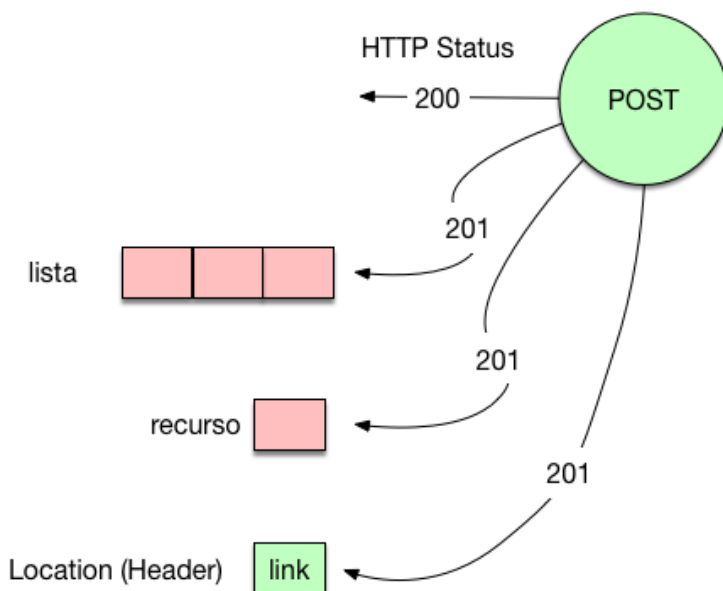
REST HTTP Return codes y GET

El primer caso es muy sencillo ante una petición GET , simplemente devolvemos un status code de 200 que es OK y el recurso o lista de recursos que necesitamos. **No hay nada más que hacer.**



REST HTTP Return codes y POST

Este tipo de peticiones ya **son mucho mas abiertas** . Hay gente que opta por devolver simplemente **un status code de 200** . Pero no es una gran solución, normalmente es preferible devolver un status **code de 201 (recurso creado)**. Una vez esto está claro la pregunta que nos viene a la mente es **si debemos de devolver algún tipo de contenido**. Hay enfoques que devuelven la nueva lista de recursos , o devuelven el nuevo recurso creado . Son opciones posibles , pero probablemente la más flexible **sea devolver una cabecera de location con la url del nuevo recurso**.



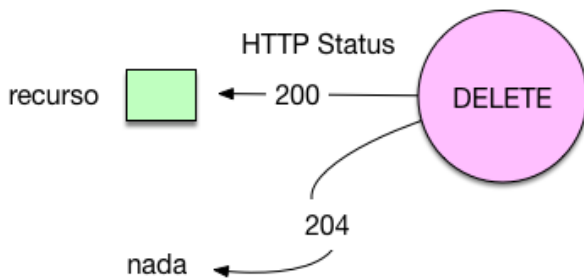
REST HTTP Return codes y DELETE

La casuística de los borrados es muy diferente a la de las inserciones pero básicamente existen dos opciones principales. Devolver un status code de 200 en el caso de que hayamos borrado y

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](#)

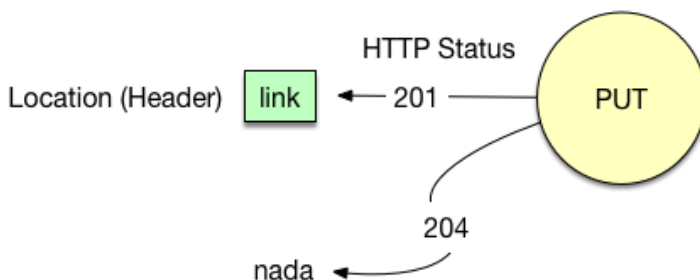
[ACEPTAR](#)



REST HTTP Return codes y PUT

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

Este caso es un poco más peculiar porque una orden de put puede hacer dos cosas o modificar un recurso existente o si este recurso no existe puede generarlo. En el caso de que simplemente actualice el registro enviaremos un status code de 204 . Si lo que sucede es que se genera un nuevo recurso enviaremos un **status code de 201 y un location header**.



El diseño de arquitecturas REST siempre ha sido algo complejo.

Otros artículos relacionados

1. [Arquitecturas REST y sus niveles](#)
2. [Spring REST Client con RestTemplates](#)
3. [PostMan App con Spring REST](#)
4. [¿ Qué es REST ?](#)

Links Externos

Introducción a REST

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](#)



JVM DEDICADA
HOSTING JAVA APPS
Tomcat, Glassfish/Payara y WildFly desde sólo 79€/Año

[MÁS INFO](#)www.anw.es

Archivada en: [Arquitectura](#)

Leave a Reply

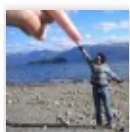
1 Comment on "REST HTTP return codes y sus curiosidades"



Join the discussion

[Subscribe](#) ▼

▲ [newest](#) ▲ [oldest](#) ▲ [most voted](#)



Guest

Sebastián



Agregaría que en el PUT a veces se envían los datos a ser cambiados (no todos) y en otros casos se envían todos ya que se supone que lo que no es enviado significa qué hay que quitarlos. Esa semántica no está de más aclararla, no?

+ 0 - [Reply](#)

🕒 1 hour ago

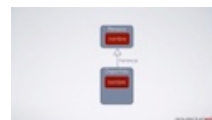


BUSCAR

PACKJAVACORE

Mis Cursos de Java Gratuitos

Java Herencia



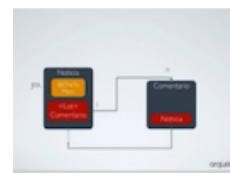
Java JDBC



Servlets



Introduccion JPA



Mis Cursos de Java

Programación Orientada a Objeto en Java



Java APIS Core





Arquitectura Java Solida con Spring



POPULAR

[Spring REST Client con RestTemplates](#)

[Nuevo Curso:Arquitectura Java Sólida con Spring 4.3 y Anotaciones](#)

[Arquitecturas REST y sus niveles](#)

[Angular 5 Hello World y su funcionamiento](#)

[Java 9 Modules y el concepto de modularidad](#)

[Java 8 Lambda Syntax ,simplificando nuestro código](#)

[El concepto de Java Annotations y su funcionamiento](#)

[Spring REST Test utilizando Rest Assured](#)

[Java 9 Collections y sus novedades](#)

[Spring @Qualifier utilizando @Autowired](#)

CONTACTO

contacto@arquitecturajava.com

LO MAS LEIDO

[¿Qué es Spring Boot?](#)

[Introducción a Servicios REST](#)

[¿Cuales son las certificaciones Java?](#)

[Java Override y encapsulación](#)

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

[¿Qué es Gradle?](#)

[REST JSON y Java](#)

[Usando el patron factory](#)

[Ejemplo de JPA , Introducción \(I\)](#)

[Uso de Java Generics \(I\)](#)

[¿Qué es un Microservicio?](#)

[Comparando java == vs equals](#)

[Mis Libros](#)

[Arquitecturas REST y sus niveles](#)

[Spring MVC Configuración \(I\)](#)

[Angular router y su configuración](#)

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