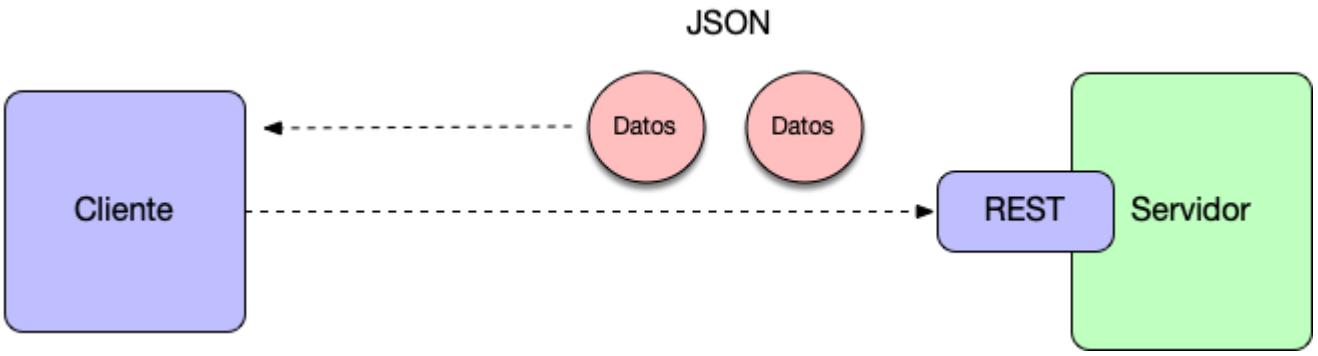
Tabla de Contenidos

- Servicios REST y Comunicación
- Stateless
- Neutralidad Tecnológica
- REST Recursos y Uniformidad
- Servicios REST y verbos HTTP
- Otros artículos relacionados

Cada día necesitamos más usar servicios REST . ¿Porque? .Porque cada día nos encontramos con una mayor necesidad de comunicar aplicaciones diferentes y compartir datos entre ellas. Nos guste o no nuestras aplicaciones dependen de otras que publican otra información que nosotros necesitamos.

Servicios REST y Comunicación

Cuando trabajamos con servicios REST estamos trabajando con una arquitectura Cliente Servidor en la cual el servidor publica la información en formato de dato puro . Normalmente en formato JSON estos datos pueden ser accedidos por una aplicación cliente que se encargará de procesarlos y presentar el resultado en un interface de usuario a nuestro Cliente.

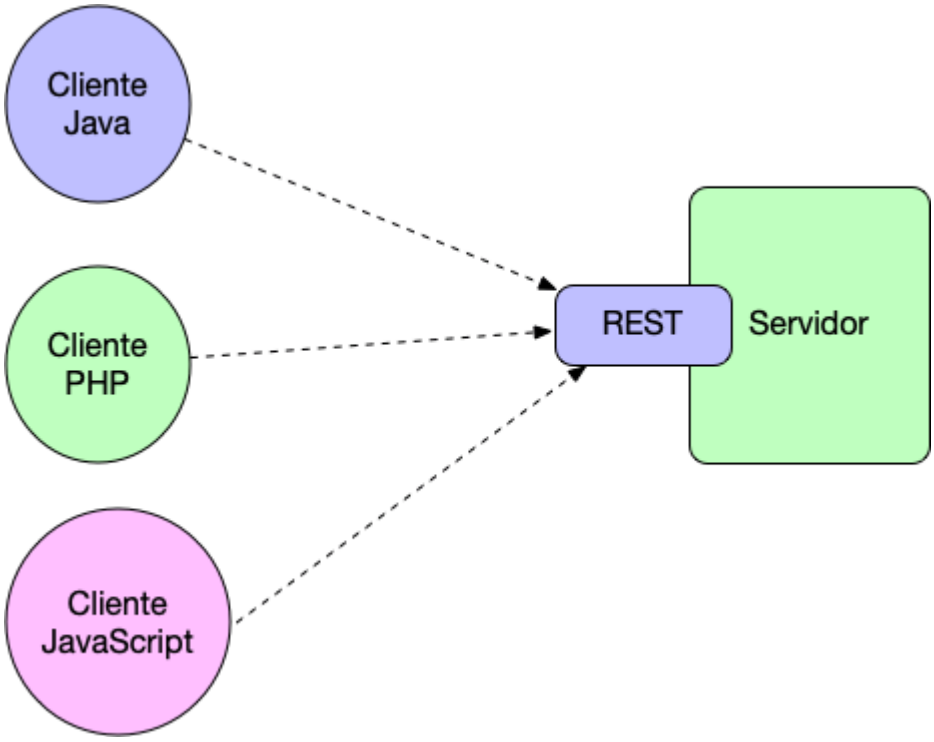


Stateless

Estos servicios no se encargan de mantener ningún tipo de estado entre peticiones y cada una de las peticiones es totalmente independiente de la siguiente. Al no mantener estado nos encontramos que se incrementa la escalabilidad de estos.

Neutralidad Tecnológica

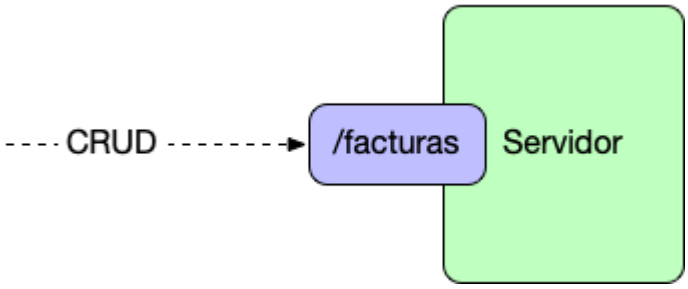
Otra de las grandes ventajas de los Servicios REST es su neutralidad tecnológica ya que permite a prácticamente cualquier tipo de cliente y de lenguaje conectarse a ellos.



El contenido de los servicios web REST ha se puede cachear de tal forma que una vez realizada la primera petición al servicio el resto puedan apoyarse en la cache si fuera necesario.

REST Recursos y Uniformidad

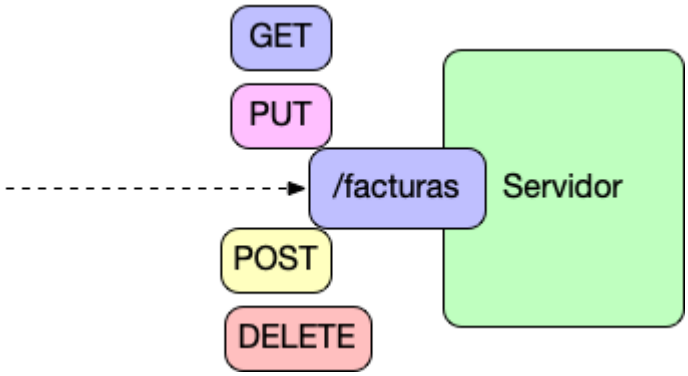
Otro de los conceptos claves de los servicios REST es el uso URLs orientadas a recursos en donde cada una de ellas gestiona todas las operaciones que un recurso concreto soporta . Ejemplos de recursos pueden ser /facturas /clientes o /libros . Cada una de estas URLs gestiona las operaciones CRUD de cada uno de los recursos . Búsquedas ,inserciones , actualizaciones , borrados etc



Servicios REST y verbos HTTP

Para realizar estas operaciones se hace uso de los verbos clásicos del protocolo HTTP

- GET: Verbo que solicita información al servicio
- POST : Verbo que inserta información del servicio
- PUT : Verbo que actualiza la información del servicio
- DELETE : Verbo que borra la información del servicio



A continuación se muestra un video del Curso gratuito de Introducción a Spring REST que nos muestra como configurar a nivel de Spring Framework un servicio REST con su método GET el resto de los verbos está cubierto en esta introducción:

Otros artículos relacionados

- REST URL formatos y buenas prácticas.
- Visual Studio Code REST Client
- Curso Spring REST
- Spring Boot REST JPA y JSON