

Unidad 8.

Acceso y creación de Servicios Web

¿Qué es un servicio web?

Los servicios web son un tipo particular de aplicación web: una aplicación pensada no solo para ser usada por usuarios humanos, sino por otras aplicaciones.

Un servicio web es una aplicación web capaz de comunicarse e intercambiar información con otra aplicación (que denominaremos cliente) independientemente de la plataforma en la que cada una se ejecute.

Es decir, el servicio web puede estar programado en PHP y correr bajo un sistema operativo GNU/Linux y el cliente puede estar programado con C# y correr bajo un Windows, y deberían ser capaces de comunicarse y trabajar juntas. Pero es importante que quede claro que, en este caso, la aplicación web (servidor) y la aplicación cliente son dos aplicaciones diferentes.

Los mensajes que las aplicaciones se intercambian generalmente tienen formato XML o JSON.

Existen dos estándares principales en la industria para implementar servicios web, denominados SOAP y REST. En este tema vamos a ver el estándar REST.

Diferencias entre servicios web y aplicaciones web

Una aplicación web está diseñada para que un ser humano interactúe con ella a través de un interfaz. Un servicio web, en cambio, está pensado para que lo use otra aplicación informática (el cliente), no necesariamente un ser humano.

Por ese motivo, los servicios web suelen carecer de interfaz de usuario y no producen salidas HTML legibles. Es decir, un servicio web puro no suele tener vistas.

Por lo demás, un servicio web puede tener una arquitectura aproximadamente MVC, y digo aproximadamente porque el servicio web, como acabo de contarte, carece de vistas. Pero puede seguir conservando sus controladores y sus modelos. Los controladores se encargarán de convertir los datos de los modelos a JSON o XML y devolverlos al cliente.

REST

REST (Representational State Transfer) es un mecanismo de intercambio de información entre clientes y servidores de una red.

Está orientado a los datos, esto es, proporciona siempre los mismos tipos de acceso a los recursos, sin posibilidad de definir nuevas operaciones.

Actualmente, gran parte de las APIs, ya sean públicas o privadas, se programan según el diseño REST para que los programadores que las usen sepan qué esperar de la API sin necesidad de consultar farragosas páginas de documentación o de pelearse con el servidor mediante el infalible método de ensayo y error.

Los verbos http: GET, POST, PUT, PATCH y DELETE

El protocolo http define dos tipos de petición al servidor, GET y POST. El estándar REST aumenta estos tipos en otros tres: PUT, PATCH y DELETE.

Veamos qué significa exactamente cada uno:

- GET se utiliza para solicitar datos al servidor. Por ejemplo: “Dame toda la información de un producto”.
- POST se utiliza para enviar datos al servidor. Por ejemplo: “Aquí tienes toda la información de un producto; anda, almacénalo en tu base de datos”.
- PUT/PATCH se utiliza para solicitar al servidor la modificación de datos que ya existen. Por ejemplo: “Este es el nuevo precio de un producto que ya estaba en tu base de datos. Tómallo y actualízalo”.
Si se van a actualizar todos los datos de un producto, se utiliza PUT. Si se va a actualizar solo una parte de los datos de un producto, se utiliza PATCH.
- DELETE se usa para solicitar la eliminación de datos en el servidor. Por ejemplo: “Elimina este producto”.

Los verbos http: GET, POST, PUT, PATCH y DELETE

Operación	Significado	Verbo	URL típica
index	Mostrar todos los producto	GET	https://servidor/producto/
show	Mostrar un producto	GET	https://servidor/producto/id
create	Mostrar formulario de creación de un producto	GET	https://servidor/producto/create
store	Crear un producto con los datos procedentes de un formulario	POST	https://servidor/producto/store
edit	Mostrar el formulario de edición de un producto	GET	https://servidor/producto/edit/id
update	Actualizar el producto con los datos procedentes del formulario	PUT o PATCH	https://servidor/producto/update/id
destroy	Eliminar un producto	DELETE	https://servidor/producto/destroy/id

Autenticación

