

Servicios web y REST

Agentes de extracción de información en Internet

Departamento de Sistemas Informáticos

Escuela Técnica superior de Ingeniería de Sistemas Informáticos - UPM

16 de febrero de 2024

License CC BY-NC-SA 4.0

Un servicio web es un conjunto de protocolos y estándares que permiten la comunicación entre aplicaciones

- Se basan en estándares abiertos que funcionan sobre el protocolo HTTP
- Su origen proviene de estándares de ejecución remota de código de los 90
- Un conjunto de servicios web con propósito común se denomina *API (Application Programming Interface)*
- Pueden estar documentados también para máquinas usando *WSDL (Web Services Description Language)*

En la actualidad, estos estándares están en desuso por el auge de la web

- La transmisión de datos es a través del puerto 80 o 443 (adios *firewalls*)
- Protocolos que funcionan sobre un estándar ya probado (SOAP)
- O simplemente conjuntos de pautas para hacer la comunicación más ligera (REST)

Acerca de REST

Representational State Transfer (REST) es un estilo de arquitectura de software

- No es un protocolo, es un conjunto de principios para la creación de servicios web

Sus principios arquitectónicos son:

- **Interfaz uniforme:** La información se envía en formato estándar (XML, JSON, etc.)
- **Sin estado:** El servidor completa solicitudes independientes (es inherente a HTTP)
- **Distribuido:** Permite que varios servidores trabajen juntos
- **Almacenamiento en caché:** Ayuda a mejorar el rendimiento
- **Código bajo demanda:** El servidor puede enviar código al cliente para ejecutarlo

Las API que siguen la arquitectura de REST se llaman **API RESTful**

API RESTful

Son aquellos **servicios web** que implementan una arquitectura **REST**

Requieren que las peticiones contengan los siguientes **elementos principales**:

- **URL**: La ruta hacia el recurso o servicio
- **Método**: La acción que se quiere realizar sobre el servicio, que suelen ser:
 - **GET**: Para obtener listas (e.g. `GET /games`) o elementos (e.g. `GET /game/1`)
 - **POST**: Para enviar información o crear elementos (e.g. `POST /game`)
 - **PUT** o **PATCH**: Para actualizar elementos (e.g. `PUT /game/1`)
 - **DELETE**: Para eliminar elementos (e.g. `DELETE /game/1`)
- **Cuerpo**: La información enviada al servidor (para peticiones **POST**, **PUT** o **PATCH**)
- **Cabeceras**: Información adicional de la petición, (e.g. datos de autenticación)

Es un estándar muy usado en la actualidad, pero no es obligatorio

Gracias