



Definiciones de **Servicios Web**

Universidad de Málaga: Grado en Ingeniería Informática

— **Antonio J. Galán Herrera**

Índice

Introducción	1
JSON JavaScript Object Notation	1
REST Representational State Transfer	2
SOAP Simple Object Access Protocol	3
SOA Service Oriented Architecture	3
XML Extensible Markup Language	4
Conclusión	4
Bibliografía	5

Introducción

El diseño del software tiende a ser cada vez más modular, por lo que las aplicaciones se componen de una serie de componentes (servicios) reutilizables que pueden encontrarse distribuidos por máquinas conectadas a la red.

Un **Servicio Web** es una colección de procedimientos a los que poder invocar desde cualquier parte de Internet (o una Intranet), siendo esta invocación totalmente independiente a la plataforma que se utilice y del lenguaje de programación implementado internamente en el servicio.

En este documento se tratarán 5 **Servicios Web** distintos: JSON, REST, SOAP, SOA y XML.

JSON *JavaScript Object Notation*

Descripción

Se trata de un **formato ligero de intercambio de datos**, cuya función principal es ser **simple de leer y de escribir** para las **personas**, y **simple de generar y de interpretar** para las **máquinas**.

JSON emplea convenciones en su lenguaje que son familiares con conceptos de los lenguajes C#, Java, JavaScript, Perl, Python... lo que hacen de este formato un lenguaje ideal para el intercambio de datos.

Se basa en **2** estructuras: una **colección de pares** nombre-valor (conocido como *objeto*, *registro*, *struct*, *diccionario*...); y una **lista ordenada** de valores (*array*, *vector*...).

Ejemplos

MyJson: <http://myjson.com/api>

PrexView: <https://prexview.com/>

REST *Representational State Transfer*

Descripción

Básicamente, se define como un servicio que no posee estado. Ampliando más la definición, quiere decir que **no ofrece «recordar» datos, sino que tras cada llamada o invocación del servicio, los datos mencionados deben volver a ser introducidos.**

Su mayor **desventaja** es evidente: resulta como mínimo, **tedioso**, pues con cada llamada debe añadirse también el estado (los datos); sin embargo, su **ventaja** es aún mayor: proporciona **escalabilidad**, para un servidor equivale a no tener que memorizar ni guardar datos proveniente de sus clientes, lo que supone no llegar a llenar la memoria de datos de clientes.

Ejemplos

Gmail API: <https://developers.google.com/gmail/api/>

Req Res: <https://reqres.in/>

SOAP *Simple Object Access Protocol*

Descripción

Se trata de un **protocolo de simplificación de datos y sin estado** con el que poder establecer una **comunicación** entre aplicaciones sobre **diferentes sistemas operativos, tecnologías o lenguajes** de programación.

Este protocolo está **basado en XML** y se compone de **3** partes:

1. **Sobre:** Define qué hay en el mensaje y cómo procesarlo.
2. **Reglas de codificación:** Expresar instancias de tipos de datos.
3. **Convención:** Representar llamadas a procedimientos y respuestas.

Ejemplo

SOAP UI: <https://www.soapui.org/>

SOA *Service Oriented Architecture*

Descripción

SOA es un **tipo de arquitectura TI orientada a servicios**, que consiste en un **marco de trabajo que establece una estructura de diseño para la integración de aplicaciones**.

Esto permite la **reducción de costos** de implementación, **innovación de servicios** a clientes, **agilización** ante cambios... debido a que se **combinan las nuevas tecnologías con aplicaciones independientes de una manera sencilla**.

Ejemplos

Al usar nuestra tarjeta de crédito en algún establecimiento para para un bien o un servicio.

Al usar el teléfono, el consumo de tiempo utilizado es facturado automáticamente en el plan en tiempo real.

XML *Extensible Markup Language*

Descripción

Se trata de un **meta-lenguaje** que permite definir lenguajes de marcas desarrollado por el *World Wide Web Consortium (W3C)*, **utilizado para almacenar datos de forma legible**.

Define la gramática de lenguajes específicos para estructurar documentos grandes, proporcionando **soporte** a bases de datos y permitiendo que varias aplicaciones puedan comunicarse entre sí o integrar información.

Ejemplo

PAN-OS XML API: <https://docs.paloaltonetworks.com/pan-os/8-1/pan-os-panorama-api/about-the-pan-os-xml-api.html>

PrexView: <https://prexview.com/>

Conclusión

Personalmente, considero útil esta tarea porque hace al alumno investigar por su cuenta las definiciones, lo que implica leer más, dotándolo de perspectiva a la hora de definir mentalmente cada uno de esos términos.

Por otra parte, referente a los términos mencionados, los encuentro interesantes al tratarse protocolos y arquitecturas, precisamente por la importancia -y presencia- que estos tienen en el mundo de la informática.

Por último, también ha resuelto una curiosidad que sentía hacia JSON, ya que había oído hablar de él y que estaba ganando fama debido a su simplicidad y a su gran utilidad, pero no sabía muy bien de qué se traba. De igual forma, conocía XML pero no sabía realmente qué era y ni para qué se utilizaba.

Este trabajo me ha ayudado a profundizar en términos de la informática que estoy seguro que alguna vez voy a usar.

Bibliografía

Introducción

<http://www.jtech.ua.es/j2ee/publico/servc-web-2012-13/sesion01-apuntes.html>

JSON

<https://www.json.org/json-es.html>

REST

<https://desarrolloweb.com/articulos/que-es-rest-caracteristicas-sistemas.html>

SOAP

https://www.w3schools.com/xml/xml_soap.asp

https://es.wikipedia.org/wiki/Simple_Object_Access_Protocol

SOA

<http://www.i2btech.com/blog-i2b/tech-deployment/que-se-entiende-por-soa-y-cuales-son-sus-beneficios/>

<https://www.megapractical.com/blog-de-arquitectura-soa-y-desarrollo-de-software/soa-en-nuestra-vida-diaria>

XML

https://es.wikipedia.org/wiki/Extensible_Markup_Language#cite_note-1