**¿SOAP o REST?**

Cuando hablamos de **tipos de PAC** nos encontramos con 2 tipos, aquellos que ofrecen servicios **REST** y los que lo hacen por **SOAP**.

**¿Que es REST o RESTful como WebService?**

**REST** se define como Transferencia de estado representacional en inglés (**RE**presentational **S**tate **T**ransfer).

REST no define tantos estándares como SOAP. REST es para exponer las API públicas (es decir, la API de Facebook, la API de Google Maps) a través de Internet para manejar las operaciones de CRUD en los datos. REST se enfoca en acceder a recursos nombrados a través de una única interfaz consistente.

**RESTful**hace referencia a un servicio web que implementa la arquitectura REST.

**¿Que es SOAP como WebService?**

**SOAP** de las siglas en inglés (**S**imple **O**bject **A**ccess **P**rotocol)

SOAP expone operaciones. SOAP se enfoca en acceder a operaciones con nombre, cada operación implementa alguna lógica de negocios. Aunque a SOAP se le conoce comúnmente como servicios web, esto es incorrecto. SOAP tiene muy poco o nada que ver con la web. REST proporciona verdaderos servicios web basados en URI y HTTP.

**Diferencias**

**SOAP**

* WSDL define el esquema entre el cliente y el servicio y es estático por su naturaleza.
* SOAP construye un protocolo basado en XML sobre HTTP o, a veces, TCP / IP.
* SOAP es un sucesor de XML-RPC y es muy similar, pero describe una forma estándar de comunicación.
* Varios lenguajes de programación tienen soporte nativo para SOAP, normalmente necesita de una URL de servicio web y puede llamar a sus funciones de web service sin la necesidad de un código específico.

**REST**

* REST permite muchos formatos de datos diferentes, mientras que SOAP solo permite XML. Si bien esto puede parecer que agrega complejidad a REST porque necesita manejar múltiples formatos, en mi experiencia, ha sido bastante beneficioso. JSON por lo general es un mejor ajuste para los datos y analiza mucho más rápido. REST permite un mejor soporte para los clientes del navegador debido a su soporte para JSON.
* REST tiene mejor rendimiento y escalabilidad. Las lecturas REST se pueden almacenar en caché; Las lecturas basadas en SOAP no se pueden almacenar en caché.
* No se requiere de ninguna herramienta costosa para interactuar con el servicio web. Incluso en lenguajes de programación antiguos se puede hacer la comunicación a través de un HTTP.
* Eficiente (SOAP usa XML para todos los mensajes, REST puede usar formatos de mensajes más pequeños como son texto plano, HTML, XML, JSON etc.).
* Rápido (no requiere procesamiento extenso).

**Ejemplo de Request-Response**

Tomando como ejemplo un consumo de WebService para la obtención del token, tenemos los siguientes casos.

**SOAP**

**Request**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20 | POST http:*//*[*pruebascfdi.smartweb.com.mx/Autenticacion/wsAutenticacion.asmx*](http://pruebascfdi.smartweb.com.mx/Autenticacion/wsAutenticacion.asmx) *HTTP/1.1*  Accept-Encoding: gzip,deflate  Content-Type: text/xml;charset=UTF-8  SOAPAction: "<http://sufacturacion.com/AutenticarBasico>"  Content-Length: 402  Host: pruebascfdi.smartweb.com.mx  Connection: Keep-Alive  User-Agent: Apache-HttpClient/4.1.1 (java 1.5)    <soapenv:Envelope xmlns:soapenv="<http://schemas.xmlsoap.org/soap/envelope/>" xmlns:suf="<http://sufacturacion.com/>">     <soapenv:Header/>     <soapenv:Body>        <suf:AutenticarBasico>           <!--Optional:-->           <suf:usuario>demo</suf:usuario>           <!--Optional:-->           <suf:password>123456789</suf:password>        </suf:AutenticarBasico>     </soapenv:Body>  </soapenv:Envelope> |

**Response**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | HTTP/1.1 200 OK  Cache-Control: private, max-age=0  Content-Type: text/xml; charset=utf-8  Server: Microsoft-IIS/7.5  X-AspNet-Version: 4.0.30319  X-Powered-By: ASP.NET  Date: Fri, 23 Nov 2018 16:28:40 GMT  Content-Length: 1313    <?xml version="1.0" encoding="utf-8"?>  <soap:Envelope xmlns:soap="<http://schemas.xmlsoap.org/soap/envelope/>" xmlns:xsd="<http://www.w3.org/2001/XMLSchema>" xmlns:xsi="<http://www.w3.org/2001/XMLSchema-instance>">  <soap:Body>  <AutenticarBasicoResponse xmlns="<http://sufacturacion.com/>">  <AutenticarBasicoResult>T2lYQ0t4L0RHVkR4dHZ5Nkk1VHNEakZ3Y0J4Nk9GODZuRyt4cE1wVm5tbXB3YVZxTHdOdHAwVXY2NTdJb1hkREtXTzE3dk9pMmdMdkFDR2xFWFVPUTQyWFhnTUxGYjdKdG8xQTZWVjFrUDNiOTVrRkhiOGk3RHladHdMaEM0cS8rcklzaUhJOGozWjN0K2h6R3gwQzF0c0g5aGNBYUt6N2srR3VoMUw3amtvPQ..ux9Wx968vjlsY7HXR0j-JtZjvDRwm8na9Yn5A0AumI4</AutenticarBasicoResult>  </AutenticarBasicoResponse>  </soap:Body>  </soap:Envelope> |

**REST**

**Request**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | POST http:*//*[*services.test.sw.com.mx/security/authenticate*](http://services.test.sw.com.mx/security/authenticate) *HTTP/1.1*  Content-Type: application/json  user: demo  password: 123456789  Content-Length: 0  cache-control: no-cache  User-Agent: PostmanRuntime/7.4.0  Accept: \*/\*  Host: services.test.sw.com.mx  accept-encoding: gzip, deflate  Connection: keep-alive |

**Response**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 | HTTP/1.1 200 OK  Content-Type: application/json; charset=utf-8  Vary: Accept-Encoding  Server: Kestrel  X-Powered-By: ASP.NET  Date: Fri, 23 Nov 2018 16:30:08 GMT  Content-Length: 1018    {    "data": {      "token": "T2lYQ0t4L0RHVkR4dHZ5Nkk1VHNEakZ3Y0J4Nk9GODZuRyt4cE1wVm5tbXB3YVZxTHdOdHAwVXY2NTdJb1hkREtXTzE3dk9pMmdMdkFDR2xFWFVPUTQyWFhnTUxGYjdKdG8xQTZWVjFrUDNiOTVrRkhiOGk3RHladHdMaEM0cS8rcklzaUhJOGozWjN0K2h6R3gwQzF0c0g5aGNBYUt6N2srR3VoMUw3amtvPQ..42C9cyxIrlkMMPvrZAwRMRyp84t8ag-kC2N2EJ\_Ft\_w",      "expires\_in": 1542997809,      "tokeny\_type": "Bearer"    },    "status": "success"  } |

**Diferencias**

En SOAP tenemos 402 bytes como request y 1313 como response, lo que nos dejaría con un total de 1715 bytes transmitidos en un tiempo de 193 ms.

En REST tenemos 0 bytes como request y 1018 bytes como response, lo que nos deja con un total de 1018 bytes, transmitidos en un tiempo de 87 ms.

Señalar, que en este ejemplo el SOAP solo recibe el token y a través del servicio REST se envían datos adicionales, como un timestamp, y el tipo de token.

**Conclusiones**

Hablando en términos de un timbrado de facturas electrónicas tendríamos que considerar lo siguiente.

**SOAP**

Cuando enviemos datos estos serán procesados además del tiempo de envío y recepción, lo que nos dejará con un tiempo aproximado de 2-15 segundos para la emisión de una factura. En nuestro lenguaje de programación cuando obtengamos la respuesta del Servicio Web, necesitamos hacer el parseo de la respuesta a través de un DOM Document para obtener los datos importantes, lo cual suma un poco más de tiempo.

Cuando el Servicio Web se vuelva más popular, es probable que el servicio se sature cada vez más y esté se vuelva más lento.

**REST**

Cuando enviamos datos, solo enviaremos aquello que es importante, en cuestión de timbrado sería el XML, el cual se procesa de manera rápida (Hacer verificación de memoria caché para evitar un timbre duplicado), lo cual nos deja con tiempo de 100ms – 1 segundo para la emisión de una factura. En nuestro lenguaje de programación cuando obtengamos la respuesta hacemos el parseo del JSON (el cual es más rápido), lo cual nos suma unos pocos mili-segundos.

Cuando el Servicio Web se vuelva más popular, simplemente se añaden más instancias en la nube, así el trabajo se distribuye usando un balanceador de carga.