

ELEVATE 2018 RECAP SEMINAR

PowerBuilder User Regional Seminar – Barcelona, Spain

Hotel NH Sants Barcelona

REST/JSON vs SOAP/XML



Marco MEONI
27-28 November 2018

©2018 Appeon Limited and its subsidiaries. All rights reserved.

APPEON

DISCLAIMER

This presentation was authored by volunteer(s) in the Appeon community. This is not a work for hire by Appeon. The views and opinions expressed in this presentation are those of the author(s).

Its contents are protected by US copyright law and may not be reproduced, distributed, transmitted, displayed, published or broadcast without the prior written permission of Appeon. All rights belong to their respective owners.

Any reference to third-party materials, including but not limited to Websites, content, services, or software, has not been reviewed or endorsed by Appeon. YOUR USE OF THIRD-PARTY MATERIALS SHALL BE AT YOUR OWN RISK.

Appeon makes no warranty of any kind, either express or implied, including but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement. Appeon assumes no responsibility for errors or omissions.

Presentation Agenda

- SOAP vs REST
 - Basic Concepts
 - Pros and Cons
 - Examples
 - Use Cases
 - PowerBuilder Support
 - Debate

Author Profile



Marco MEONI

-  [linkedin.com/in/meonimarco](https://www.linkedin.com/in/meonimarco)
-  twitter.com/marcomeoni
-  pb_support@softpi.com

Key Skills

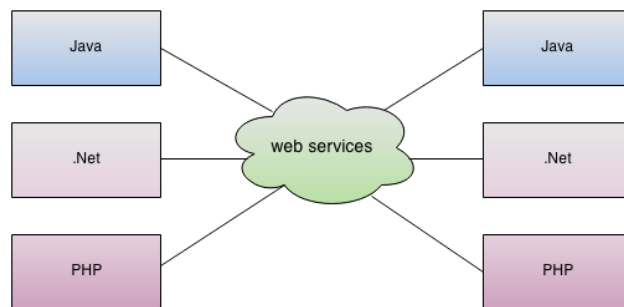
- Appeon MVP
- PowerServer
- Big Data
- PowerBuilder
- Machine Learning

Recent Projects

- **2011 - 2018** : hands on ~100 PowerServer projects
- **2014 - 2018** : PhD on Machine Learning applied to Big Data

Web Service

- Set of operations published and accessed over the network
- Platform and language independent
- Stateless design



SOAP to wash REST to relax



Origin

REST

- Representational State Transfer
- Created in 2000 by Roy Fielding in UC, Irvine
- Developed in academic environment
- Embraces the philosophy of the open Web

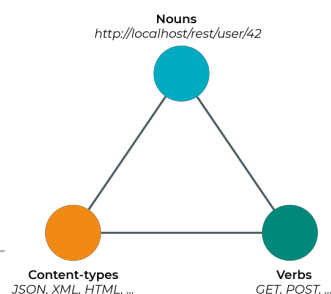
SOAP

- Simple Object Access Protocol
- Created in 1998 by Dave Winer et al in collaboration with MS
- Developed in large software company
- Addresses the needs of enterprise market

Basic Concepts

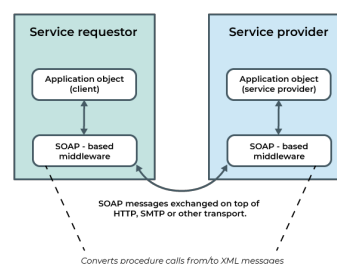
REST

- Makes data available as resources (nouns)
- E.g. “user” or “invoice”



SOAP

- Makes data available as services (verb + noun)
- E.g. “getUser” or “PayInvoice”



Pros



REST

- Clearly separates client and server implementations
- Easier to scale because reads can be cached and prevent multiple calls
- Lightweight: no extra-layers over HTTP
- Javascript friendly: executing HTTP requests is natively supported
- Can return data in multiple formats (JSON, XML etc)



SOAP

- Works on any communication protocol, even asynchronously
- It is a W3C standard
- Information about objects is communicated to clients
- Security and authorization are part of the protocol (XML encrypt, retry on failure, WS-Security, WS-Addressing, WS-Coordination, etc...)
- Can be fully described using WSDL

Cons



REST

- Only works on top of the HTTP protocol
- Hard to enforce authorization and security on top of it



SOAP

- Spends a lot of bandwidth communicating metadata
- Complex XML makes it slower than middleware such as CORBA and ACID-compliant transactions
- Only uses XML as encoding scheme for transport of request/response

SOAP Request

```
POST /InStock HTTP/1.1
Host: www.example.org
Content-Type: application/soap+xml; charset=utf-8
Content-Length: nnn

<?xml version="1.0"?>

<soap:Envelope
xmlns:soap="http://www.w3.org/2003/05/soap-envelope/"
soap:encodingStyle="http://www.w3.org/2003/05/soap-encoding">

  <soap:Body xmlns:m="http://www.example.org/stock">
    <m:GetStockPrice>
      <m:StockName>IBM</m:StockName>
    </m:GetStockPrice>
  </soap:Body>

</soap:Envelope>
```



SOAP Response

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: nnn

<?xml version="1.0"?>

<soap:Envelope
xmlns:soap="http://www.w3.org/2003/05/soap-envelope/"
soap:encodingStyle="http://www.w3.org/2003/05/soap-encoding">

  <soap:Body xmlns:m="http://www.example.org/stock">
    <m:GetStockPriceResponse>
      <m:Price>34.5</m:Price>
    </m:GetStockPriceResponse>
  </soap:Body>

</soap:Envelope>
```



WSDL

```
<?xml version="1.0"?>
<definitions>
  <message name="GetStockPriceRequest">
    <part name="StockName" type="xs:string"/>
  </message>
  <message name="GetStockPriceResponse">
    <part name="Price" type="xs:string"/>
  </message>
  <portType name="StockPrices">
    <operation name="GetStockPrice">
      <input message="GetStockPriceRequest"/>
      <output message="GetStockPriceResponse"/>
    </operation>
  </portType>
</definitions>
```

APPEON®

REST Request

```
GET /articles?include=author HTTP/1.1
```

APPEON®

REST Response

```
HTTP/1.1 200 OK
Content-Type: application/json
{
  "data": [{
    "type": "articles",
    "id": "1",
    "attributes": { "title": "JSON API paints my bikeshed!", "body": "The shortest article. Ever.", },
    "relationships": { "author": { "data": { "id": "42", "type": "people" } } }
  ]},
  "included": [
    {
      "type": "people",
      "id": "42",
      "attributes": { "name": "John", "age": 80, "gender": "male" }
    }
  ]
}
```

APPEON®

When to use



- When clients and servers operate on Web/Mobile envs
- When object info doesn't need to be communicated to clients
- Projects with strong emphasis on simplicity and performance
- High Throughput



- When clients need access to objects on servers
- When enforce a formal contract between client and server
- Long-hawl projects with strong emphasis on security and reliability

APPEON®
elevate.appeon.com

page
16

When not to use



- When you need to enforce a strict contract between client and server
- When performing transactions that involve multiple calls

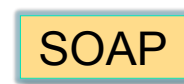


- When you want the majority of developers to easily use your API
- When your bandwidth is very limited

Common Use Cases



- Public Facing API
- Social Media services
- Social Networks
- Web Chat services
- Mobile services



- Financial services
- Payment gateways
- Telecommunication services

Popular API

REST

- Twitter
- LinkedIn
- Google

SOAP

- Salesforce
- Paypal
- Clickatell

PowerBuilder Support

REST

- HTTPClient
- RESTClient
- JSONParser, JSONGenerator
- JSONPackage
- C# WEB API
- C# DataStore

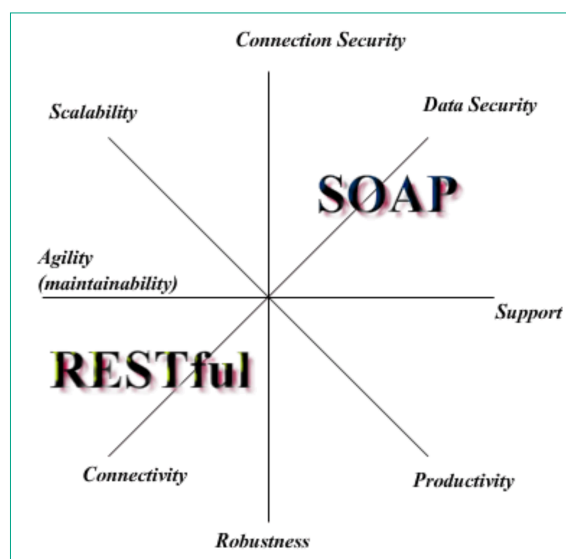
SOAP

- Web Service Target
- Web Service Proxy
- Web Service DataWindow
- INet
- HTTPClient
- OLEObject (MS XML Core Services, MSXML)

SOAP vs REST Debate

- Saying that REST or SOAP is better is fundamentally wrong
- Highly emotional debate
- Both have specific usages, strengths and weaknesses
- SOAP is strongly function oriented, REST is mostly data driven
- SOAP is designed as an RPC, REST is designed to get/put data
- REST has taken the fold over SOAP in the last 10 years
- Stronger need for simplicity+performance vs reliability+security?

Conclusions



Q&A Time



Connect with the Appeon Community



community.appeon.com

Discussions, tech articles and videos, free online training, and more.



facebook.com/AppeonPB

Encourage us with a "like", see cool pics, and get notified of upcoming events.



twitter.com/AppeonPB

Follow Appeon and community members to get the latest tech news.



linkedin.com

Build up your career profile, and stay in contact with other professionals.



youtube.com/c/AppeonHQ

Share important Appeon videos with others; no account registration required.



google.appeon.com

Follow Appeon and community members to get the latest tech news.

「 Thank You 」