

Webservices

O que é um web service?

- Solução utilizada na integração de sistemas e na comunicação entre aplicações diferentes
- Permite que um código rodando em uma máquina (cliente) interaja com um código rodando em outra (servidor)
- Permite que sistemas desenvolvidos em plataformas diferentes sejam compatíveis

Características

- O transporte de mensagens é através do protocolo HTTP
- Cada aplicação pode ter a sua própria "linguagem", que é traduzida para uma linguagem universal, um formato intermediário como *XML*, *Json*, *CSV*, *etc*
- Web Service faz com que os recursos da aplicação do software estejam disponíveis sobre a rede de uma forma normalizada
- Outras tecnologias fazem a mesma coisa, como por exemplo, os browsers da Internet acessam às páginas Web disponíveis usando por norma as tecnologias da Internet, HTTP e HTML. No entanto, estas tecnologias não são bem sucedidas na comunicação e integração de aplicações.

Por que web services?

- Frequentemente, é mais fácil integrar do que re-construir
- Permite combinar o melhor de cada ferramenta
- Segurança: publica serviços de forma simples e totalmente isolados da base de dados

Muitos consideram que os Web services corrigem um grande problema da informática: a integração de sistemas. Os Web services permitem que a integração de sistemas seja realizada de maneira compreensível, reutilizável e padronizada. É uma tentativa de organizar um cenário cercado por uma grande variedade de diferentes aplicativos, fornecedores e plataformas.

Muitas empresas temiam, no passado, prover funcionalidades na Internet devido ao medo de expor seus dados. Mas com advento dos Web services elas podem publicar serviços de forma simples e que são totalmente isolados da base de dados

Tipos de web services

- estilo-RPC: Arquitetura orientada ao serviço
 - Foco na ação sendo realizada
- estilo-REST: Arquitetura orientada ao recurso
 - Foco no objeto que está sendo manipulado

XML-RPC

- Passa as mensagens em um formato simples baseado em XML.
- Exemplo de requisição:

```
<?xml version="1.0"?>
<methodCall>
  <methodName>examples.getStateName</methodName>
  <params>
    <param>
      <value><i4>40</i4></value>
    </param>
  </params>
</methodCall>
```

- Exemplo de resposta:

```
<?xml version="1.0"?>
<methodResponse>
  <params>
    <param>
      <value><string>South Dakota</string></value>
    </param>
  </params>
</methodResponse>
```

- Biblioteca Python: xmlrpclib

XML-RPC (Exemplo 1)

Vamos fazer uma query no PyPI

```
>>> import xmlrpclib
>>> from pprint import pprint
>>> client = xmlrpclib.ServerProxy('http://pypi.python.org/pypi')
>>> client.release_urls('Plone', '4.0.1')
>>> pprint(client.release_urls('Plone', '4.0.1'))
[{'comment_text': '',
  'downloads': 177,
  'filename': 'Plone-4.0.1.zip',
  'has_sig': False,
  'md5_digest': 'be72596d49295b7207f0a861ee3530ed',
```

```
'packagetype': 'sdist',
'python_version': 'source',
'size': 1507065,
'upload_time': <DateTime '20101004T02:30:01' at 10071a248>,
'url': 'http://pypi.python.org/packages/source/P/Plone/Plone-4.0.1.zip']}]
```

XML-RPC (Exemplo 2)

- O pacote wsapi4plone.core é um exemplo de uso do XML-RPC.
- Esse pacote provê uma interface XML-RPC para se interagir com um site Plone.
- Vamos falar mais sobre ele adiante...

SOAP

- "web services grandes": descritos por vários padrões WS-* W3C
- Assim como XML-RPC, transmite mensagens em XML porém, mais complicadas (podem descrever tipos complexos)
- WSDL (web service description language) - descreve a interface em XML (para leitura por máquinas)
- Bibliotecas Python:
 - soaplib
 - suds

SOAP (Exemplo de requisição)

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:ns0="http://cicero.azavea.com/"
  xmlns:ns1="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <ns1:Body>
    <ns0:GetOfficialsByAddress>
      <ns0:authToken>FOO</ns0:authToken>
      <ns0:address>1402 3rd Ave</ns0:address>
      <ns0:city>Seattle</ns0:city>
      <ns0:state>WA</ns0:state>
      <ns0:postalCode>98101</ns0:postalCode>
      <ns0:country>US</ns0:country>
      <ns0:districtType>NATIONAL_UPPER</ns0:districtType>
      <ns0:includeAtLarge>false</ns0:includeAtLarge>
    </ns0:GetOfficialsByAddress>
  </ns1:Body>
</SOAP-ENV:Envelope>
```

SOAP (Exemplo de resposta)

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <GetOfficialsByAddressResponse xmlns="http://cicero.azavea.com/">
      <GetOfficialsByAddressResult>
        <ElectedOfficialInfo>
          <ElectedOfficialID>326f9123-4196-49ff-a9ab-cca8194a12a8</ElectedOfficialID>
          <AssemblyName />
          (snip)
          <FirstName>Maria</FirstName>
          <MiddleInitial>E.</MiddleInitial>
          <LastName>Cantwell</LastName>
          (snip)
          <LastUpdateDate>2009-03-26T00:00:00</LastUpdateDate>
        </ElectedOfficialInfo>
      </GetOfficialsByAddressResult>
    </GetOfficialsByAddressResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP (Exemplo usando a biblioteca suds)

Vamos usar o Azavea, que provê um serviço para informações geográficas sobre candidatos eleitos nos EUA.

Instalar suds:

```
sudo pip install suds
```

Exemplo:

```
>>> from suds.client import Client
>>> url_service = 'http://www.webservices.net/globalweather.asmx?WSDL'
>>> client = Client(url_service)
>>> cidades = client.service.GetCitiesByCountry(CountryName="France")
>>> tempo_em_belo_horizonte = client.service.GetWeather(CountryName="Brazil",
CityName="Belo Horizonte")
```

REST

- Orientado a recursos
- Encoraja o uso direto de funcionalidades do HTTP (métodos de requisição, passagem de parâmetros, caching, etc)

- A representação das respostas pode varia, XML e JSON são comuns.
- Bibliotecas em Python:
 - urllib/urllib2 para transferência
 - ElementTree, lxml, etc para processar XML
 - json para processar JSON

REST (Exemplo)

O OpenWeatherMap provê acesso à informações do clima de mais de 200.000 cidades do mundo.

Podemos fazer uma requisição nesse serviço usando urllib:

```
>>> from urllib import urlopen
>>> url = 'http://api.openweathermap.org/data/2.5/weather?q=Belo Horizonte'
>>> res = urlopen(url).read()
>>> res
>>> res_dic = eval(res)
>>> res_dic
```