



Проектирование и эксплуатация информационных систем в медиаиндустрии

Выломова Екатерина Алексеевна
e-mail: evylotova@gmail.com

0. Лекция 5

- Открытые системы
- Веб-сервисы
- SOAP, WSDL, UDDI
- Примеры

0. POGE

Principle of good enough

“Лучшее – враг хорошего”



Но не забывать:

Всё следует упрощать до тех пор, пока это возможно, но не более того.

I. Лекция 6. Открытые системы. Сервис-Ориентированная Архитектура

- RPC
- SOAP
- REST
- Форматы: XML, JSON
- Примеры

I. RPC

Удалённый вызов процедур— класс технологий, позволяющих компьютерным программам вызывать функции или процедуры в другом адресном пространстве (как правило, на удалённых компьютерах).

Включает в себя:

- Сетевой протокол для обмена в режиме клиент-сервер
- Язык сериализации объектов

Реализации:

SOAP, .Net Remoting, XML-RPC, JSON-RPC, Java RMI, DCOM

I. XML-RPC

XML-RPC - стандарт/протокол вызова удалённых процедур, основанный на XML, является прародителем SOAP, отличается исключительной простотой применения

Запрос:

```
<?xml version="1.0"?>
<methodCall>
<methodName>examples.getStateName</methodName>
<params>
<param>
<value><i4>41</i4></value>
</param>
</params>
</methodCall>
```

Ответ:

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

I. SOAP

SOAP — протокол обмена структурированными сообщениями в распределённой вычислительной среде.

SOAP-конверт

SOAP-заголовок

Элемент заголовка 1

Элемент заголовка 2

...

Элемент заголовка N

Тело SOAP

Элемент тела N

...

Элемент тела 2

Элемент тела 1

Используют: сервисы Google

Сервис-ориентированная архитектура

I. REST

REST — Передача состояния представления (Representational State Transfer) — это стиль архитектуры программного обеспечения для распределенных гипермедиа систем, подобных Всемирной паутине.

Веб-сервисы, созданные с использованием протокола HTTP и принципов REST, называются RESTful web service

Плюсы:

- Читабельный
(<http://search.yahooapis.com/ContentAnalysisService/V1/termExtraction>)
- Легковесный
- Не требует дополнительных утилит

Используют: все сервисы Yahoo, Twitter

Сервис-ориентированная архитектура

I. Форматы данных

- XML
- JSON:
 - Набор пар ключ/значение. В различных языках это реализовано как объект, запись, структура, словарь, хэш-таблица, список с ключом или ассоциативный массив.
 - Пронумерованный набор значений. Во многих языках это реализовано как массив, вектор, список или последовательность.

I. JSON. Пример

```
{  
    "firstName": "Иван",  
    "lastName": "Иванов",  
    "address": {  
        "streetAddress": "Московское ш., 101, кв.101",  
        "city": "Ленинград",  
        "postalCode": 101101  
    },  
    "phoneNumbers": [  
        "812 123-1234",  
        "916 123-4567"  
    ]  
}
```

Сервис-ориентированная архитектура

I. REST.Yahoo

```
import urllib2,urllib
url = 'http://developer.yahoo.com/'
try:
    data =
urllib2.urlopen(url).read()
except urllib2.HTTPError, e:
    print "HTTP error: %d" %
e.code
except urllib2.URLError, e:
    print "Network error: %s" %
e.reason.args[1]
url =
'http://search.yahooapis.com/ContentA
nalysisService/V1/termExtraction'
appid = 'YahooDemo'
```

```
context = ""
Italian sculptors and painters of the
renaissance favored
the Virgin Mary for inspiration
""
```

```
query = 'madonna'
```

```
params = urllib.urlencode({
    'appid': appid,
    'context': context,
    'query': query
})
```

```
data = urllib.urlopen(url, params).read()
```

I. REST.Twitter

Твиты, которые содержат @twitterapi и @anywhere:

```
url='http://search.twitter.com/search.json?q=%40twitterapi%20%40anywhere'  
from urllib2 import urlopen  
from simplejson import loads  
content = loads(urlopen(url).read())
```

Подробнее: <https://dev.twitter.com/docs/using-search>

I. Web-сервисы

Используем модуль suds

```
from suds.client import Client  
myclient = Client('http://example.com/?wsdl')  
result = myclient.service.dosmth()  
print result
```

Список веб-сервисов по категориям:

<http://www.webservicelist.com/>

I. GeoIP

Определяет страну по IP адресу

<http://www.webservicex.net/geoipservice.asmx>

WSDL:

<http://www.webservicex.net/geoipservice.asmx?WSDL>

Используем сервис:

```
from suds.client import Client
myclient = Client('http://www.webservicex.net/geoipservice.asmx?WSDL')
print myclient
result = myclient.service.GetGeoIP("173.194.32.48")
print result
```

I. EBI Web Services

Доступ к внутренним БД

<http://www.ebi.ac.uk/Tools/webservices/wsdl/list>

Используем:

<http://www.ebi.ac.uk/ontology-lookup/OntologyQuery.wsdl>

```
myclient = Client('http://www.ebi.ac.uk/ontology-lookup/OntologyQuery.wsdl')  
myclient.service.getOntologyNames()
```

I. Facebook graph API

<http://developers.facebook.com/docs/reference/api/>

```
from urllib2 import urlopen  
from simplejson import loads  
content = loads(urlopen('http://graph.facebook.com/2439131959').read())
```

Формат: JSON

Примеры