

Web Services (SOAP & REST)

Thomas Strub (strut1)

Marc Touw (touwm1)

Content

- Theoretical Introduction
 - Web Services
 - SOAP
 - REST
- Practical Application
 - Live Demo
 - Coding Session (optional)

Theoretical Introduction Web Services

- What most of you probably know already:
 - Web Applications present data to human users
 - Web Services deliver data to non-human clients
- What some might not now:
 - Web Services should not transform but merely serialise the data they deliver
 - Web Services can deliver the data in any format they like, possibly various

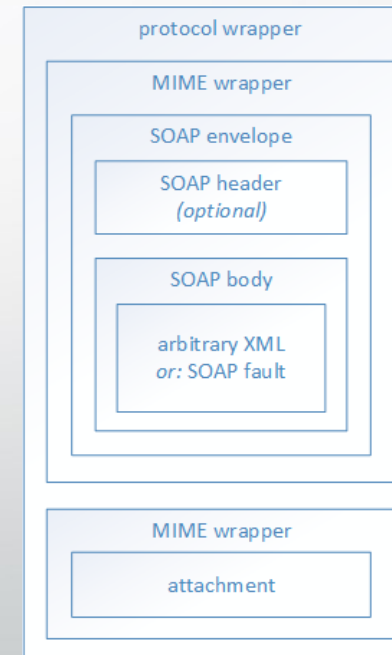
Theoretical Introduction Web Services

- Web Services
 - are a great way for distributed applications to communicate
 - are designed to bypass firewalls
 - should be platform and language independent
 - can be implemented in various ways
 - can be described in WSDL / WADL

Theoretical Introduction

SOAP

- Full-fledged protocol
 - Specification: <http://www.w3.org/TR/soap/>
- Usually served over HTTP, but can use other protocols as well
- Mandatorily XML-based
 - Envelope with header and body
- Optionally wrapped as MIME message
 - Used to attach non-XML data



Theoretical Introduction

SOAP – GetStatistics

Request Header

- POST http://localhost:5678/ HTTP/1.1
- Content-Type: text/xml; charset=UTF-8
- SOAPAction:
"http://tempuri.org/TerminalService/GetStatistics"
- Content-Length: 218
- Host: localhost:5678

Request Body

```
<soapenv:Envelope  
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"  
  xmlns:tem="http://tempuri.org/"  
<soapenv:Header/>  
<soapenv:Body>  
  <tem:GetStatistics/>  
</soapenv:Body>  
</soapenv:Envelope>
```

Theoretical Introduction

SOAP – GetStatistics

Response Header

- HTTP/1.1 200 OK
- Content-Length: 440
- Content-Type: text/xml; charset=utf-8
- Date: Thu, 21 May 2015 07:45:28 GMT

Response Body

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <GetStatisticsResponse xmlns="http://tempuri.org/">
      <GetStatisticsResult
        xmlns:a="http://schemas.datacontract.org/2004/07/BFH.NetD
        xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <a:numberOfTimeStamps>13</a:numberOfTimeStamps>
        <a:numberOfUniqueUsers>2</a:numberOfUniqueUsers>
      </GetStatisticsResult>
    </GetStatisticsResponse>
  </s:Body>
</s:Envelope>
```

Theoretical Introduction

SOAP – SetStatistics

Request Header

- POST http://localhost:5678/ HTTP/1.1
- Content-Type: text/xml; charset=UTF-8
- SOAPAction:
"http://tempuri.org/TerminalService/SetNews"
- Content-Length: 293
- Host: localhost:5678

Request Body

```
<?xml version='1.0' encoding='utf-8'>
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:tem="http://tempuri.org/"
  <soapenv:Header/>
  <soapenv:Body>
    <tem:SetNews>
      <tem:news>{\rtf1\ansi\deff0\def tab720{\fonttbl{\f0\fn
    </tem:SetNews>
  </soapenv:Body>
</soapenv:Envelope>
```


Theoretical Introduction

SOAP – SetStatistics

Response Header

- HTTP/1.1 200 OK
- Content-Length: 140
- Content-Type: text/xml; charset=utf-8
- Date: Thu, 21 May 2015 08:04:40 GMT

Response Body

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">  
  <s:Body>  
    <SetNewsResponse xmlns="http://tempuri.org/">  
  </s:Body>  
</s:Envelope>
```

Theoretical Introduction REST

- Only a pattern / style
- Can be used with any protocol
- Can be used with any data format
- Defines six constraints
 - Client-Server
 - Stateless
 - Cacheable
- Constraints (continued)
 - Layered System
 - Code on Demand (optional)
 - Uniform Interface
 - Identification of resources
 - Manipulation of resources through these representations
 - Self-descriptive messages
 - Hypermedia as the engine of application state

Theoretical Introduction

REST – GET

Request

- GET http://localhost:6789/employees HTTP/1.1
- Host: localhost:6789

Response

- HTTP/1.1 200 OK
- Content-Length: 47
- Content-Type: application/json; charset=utf-8
- Date: Thu, 21 May 2015 07:39:41 GMT
- [{"login":"thomas","name":"Thomas Reto Strub"}]

Theoretical Introduction

REST – POST

Request

- POST http://localhost:6789/employees HTTP/1.1
- Content-Type: application/json
- Content-Length: 46
- Host: localhost:6789
- {"login": "marc", "name": "Marc Touw"}

Response

- HTTP/1.1 200 OK
- Content-Length: 35
- Content-Type: application/json; charset=utf-8
- Date: Thu, 21 May 2015 07:41:28 GMT
- {"login":"marc","name":"Marc Touw"}

Theoretical Introduction

REST – PUT

Request

- PUT http://localhost:6789/employees/thomas HTTP/1.1
- Content-Type: application/json
- Content-Length: 61
- Host: localhost:6789
- {"timeStamps":["2015-04-30T00:01:57.9883655+02:00"]}

Response

- HTTP/1.1 200 OK
- Content-Length: 351
- Content-Type: application/json; charset=utf-8
- Date: Mon, 25 May 2015 16:32:07 GMT
- {"login":"thomas","timeSpan":...,"timeStamps":[...]}

Theoretical Introduction

SOAP vs. REST

- Not fair to compare apples and oranges
- SOAP is a well-defined industry standard protocol
- REST is an open architectural style with established standard patterns
- Use SOAP for big, complex applications and REST for small, dynamic ones

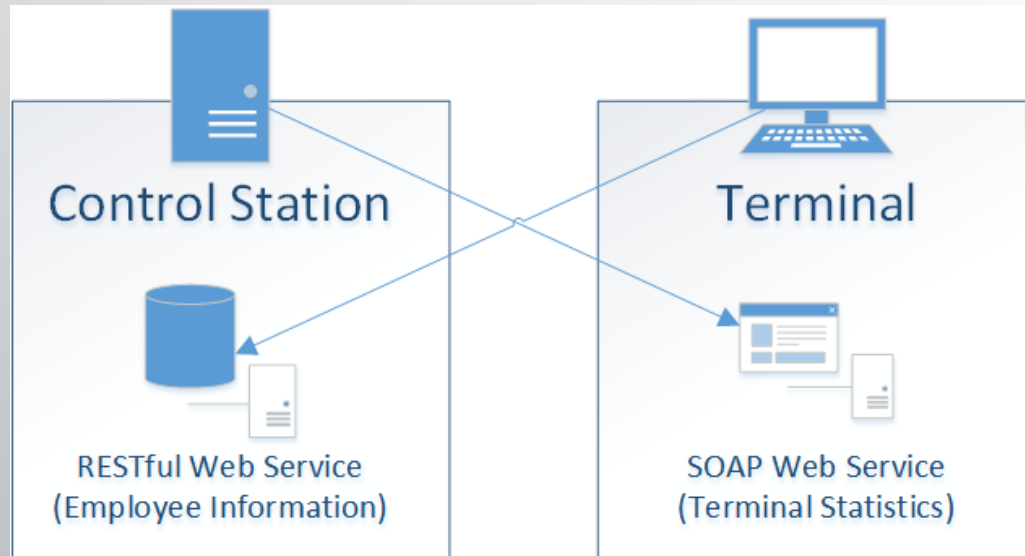
Practical Application SOAP vs. REST

- Benchmark results:
 - 1st run: JavaScriptSerializer
 - 2nd & 3rd run: Newtonsoft.Json
- There is virtually no difference
 - REST payload is slightly smaller
- <https://github.com/JamesNK/Newtonsoft.Json>

```
[AdmThS] C:\...\system32> C:\Users\ThS\Documents\fachhochschule\NetDS\11_project\Benchm
SOAP: 10'000 requests in 7456.0005ms
REST: 10'000 requests in 11867.468ms
[AdmThS] C:\...\system32> C:\Users\ThS\Documents\fachhochschule\NetDS\11_project\Benchm
SOAP: 10'000 requests in 8127.3013ms
REST: 10'000 requests in 7549.2874ms
[AdmThS] C:\...\system32> C:\Users\ThS\Documents\fachhochschule\NetDS\11_project\Benchm
SOAP: 10'000 requests in 7328.9555ms
REST: 10'000 requests in 7073.7599ms
[AdmThS] C:\...\system32>
```

Practical Application

Live Demo



<https://github.com/StrubT/NetDSWebServices>

Practical Application Coding Session

- Only you are interested and there is still some time left
- <https://github.com/StrubT/NetDSWebServices> (afterwards)

Questions & Answers

- Do you have any further questions?
- Do you have either positive or negative feedback?
- Thank you all for your undivided attention!
- Good luck with the final exam on the **29th of June at 0900.**