

1

History of XML-RPC

- RPC – Remote Procedure Call
 - ◆ Originally provided by Sun
 - ◆ Along with eXternal Data Representation
- Extended by Microsoft to XML-RPC
 - ◆ Dave Winer and Don Box
- Extended again to SOAP
 - ◆ Simple Object Access Protocol
 - ◆ Not really that simple

2

XML-RPC Data Types

- int/i4 – 32 bit signed integer
- boolean – 0 or 1 only
- string – text; default type
- double – 64 bit double
- dateTime.iso8601 – 20070125T14:12:30 CST
- base64 – encoded binary
- array
- struct

3

XML-RPC Compound Types

- Arrays
 - ◆ Variable length
 - ◆ Can contain any type
 - ◆ Types can be mixed
- Structs
 - ◆ Variable number of members
 - ◆ Members have name and value
 - ◆ Value may be any type
 - ◆ Can nest with arrays and other structs

4

Sample XML-RPC array

```
<array>
  <data>
    <value><i4>12</i4></value>
    <value><string>Hi</string></value>
    <value><boolean>0</boolean></value>
    <value><double>9.6</double></value>
  </data>
</array>
```

5

Sample XML-RPC struct

```
<struct>
  <member>
    <name>real</name>
    <value><double>9.1</double></value>
  </member>
  <member>
    <name>imaginary</name>
    <value><double>-2</double></value>
  </member>
</struct>
```

6

XML-RPC methodCall

- XML Document
- Contains 1 methodName
- Contains 0 or 1 params
- Each param contains 0 or more param
- A param has a value
- A value contains a data type

7

Sample XML-RPC methodCall

```
POST /RPC2 HTTP/1.0
Host: mydomain.com
Content-Type: text/xml
Content-length: ...

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

8

XML-RPC methodResponse

- XML Document
- Contains 1 params or 1 fault
- The params contains 1 param
- The param contains 1 value
- The value matches and valid type

9

Sample XML-RPC methodResponse

```
HTTP/1.1 200 OK
Connection: close
Content-Length: ...
Content-Type: text/xml

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

10

Sample XML-RPC fault

```

HTTP/1.1 200 OK
-
<?xml version="1.0"?>
<methodResponse>
  <fault>
    <value>
      <struct>
        <member>
          <name>faultCode</name>
          <value><int>3</int></value>
        </member>
        <member>
          <name>faultString</name>
          <value><string>Can't handle negatives</string></value>
        </member>
      </struct>
    </value>
  </fault>
</methodResponse>
    
```

11

From: <http://www.w3.org/TR/soap12-part0/#Example2>

Sample SOAP-RPC Call

```

<?xml version="1.0" ?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope" >
  <env:Header>
    <t:transaction
      xmlns:t="http://thirdparty.example.org/transaction"
      env:encodingStyle="http://example.com/encoding"
      env:mustUnderstand="true" ></t:transaction>
  </env:Header>
  <env:Body>
    <m:chargeReservation
      env:encodingStyle="http://www.w3.org/2003/05/soap-encoding"
      xmlns:m="http://travelcompany.example.org/">
      <m:reservation xmlns:m="http://travelcompany.example.org/reservation">
        <m:code>FT352BQ</m:code>
      </m:reservation>
      <o:creditCard xmlns:o="http://mycompany.example.com/financial">
        <n:name xmlns:n="http://mycompany.example.com/employees">
          Åke Jögvän Öyvind
        </n:name>
        <o:number>12345678909999</o:number>
        <o:expiration>2005-02</o:expiration>
      </o:creditCard>
    </m:chargeReservation>
  </env:Body>
</env:Envelope>
    
```

12

From: <http://www.w3.org/TR/soap12-part0/#Example51>

Sample SOAP-RPC Response

```

<?xml version="1.0" ?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope" >
  <env:Header>
    <t:transaction
      xmlns:t="http://thirdparty.example.org/transaction"
      env:encodingStyle="http://example.com/encoding"
      env:mustUnderstand="true"></t:transaction>
  </env:Header>
  <env:Body>
    <m:chargeReservationResponse
      env:encodingStyle="http://www.w3.org/2003/05/soap-encoding"
      xmlns:rpcs="http://www.w3.org/2003/05/soap-rpc"
      xmlns:m="http://travelcompany.example.org/">
      <rpc:result>
        <status>confirmed</m:status>
        <m:code>FT352BQ</m:code>
        <m:viewAt>
          http://travelcompany.example.org/reservations?code=FT352BQ
        </m:viewAt>
      </m:chargeReservationResponse>
  </env:Body>
</env:Envelope>
    
```
