XML RPC

Didier DONSEZ

Université Joseph Fourier

IMA –IMAG/LSR/ADELE

`Didier.Donsez@imag.fr`,

`Didier.Donsez@ieee.org`

Motivation

- Remote Procedure Call (RPC)
  - Sun RPC, CORBA IIOP, Java RMI, MS DCOM ORPC
- Requête-réponse
  - Méthode POST de HTTP pour le transport
  - XML pour l'encodage
  - HTTP/SSL pour la sécurité

- Prémisse des Web Services
- Mais devenu obsolète avec SOAP
  - qui s'inspire largement de XML RPC
  - voir cours http://www-adele.imag.fr/~donsez/cours/soap.pdf

La DTD pour les types de données

- Types primitifs

```
<!ELEMENT i4 (#PCDATA)>
<!ELEMENT int (#PCDATA)>
<!ELEMENT boolean (#PCDATA)>
<!ELEMENT string (#PCDATA)>
<!ELEMENT double (#PCDATA)>
<!ELEMENT dateTime.iso8601 (#PCDATA)>
<!ELEMENT base64 (#PCDATA)>
<!ELEMENT nil (EMPTY)>
```

- Tableau

```
<!ELEMENT array (data)>
<!ELEMENT data (value*)>
```

- Structure

```
<!ELEMENT struct (member*)>
<!ELEMENT member (name, value)>
<!ELEMENT name (#PCDATA)>
```

- Récursion

```
<!ELEMENT value ( i4 | int | boolean | string | dateTime.iso8601| double | base64 |
    struct | array )>
```

Types de données (et correspondance en Java)

Type XML-RPC	Type primitif	Type objet
i4	int	java.lang.Integer
int	int	java.lang.Integer
boolean	boolean	java.lang.Boolean
string	java.lang.String	java.lang.String
double	double	java.lang.Double
dateTime.iso8601	java.util.Date	java.util.Date
struct	java.util.Hashtable	java.util.Hashtable
array	java.util.Vector	java.util.Vector
base64	byte[]	byte[]
nil (extension)	null	null

Exemple de données primitives

```
<value><int>42</int></value>
<value><i4>-32764</i4></value>
<value><int>0</int></value>
<value><int>+42</int></value>
<value><double>2.0</double></value>
<value><double>-0.32653</double></value>
<value><double>67234.45</double></value>
<value><boolean>1</boolean></value>
<value><boolean>0</boolean></value>
<value>Hello, World!</value>
<value><string>Hello, World!</string></value>
<value>Tom &amp; Jerry</value>
<value><string>Once upon a time</string></value>
<value><string>"Twelve apples please," she said.</string></value>
<value><string>3 &lt; 5</string></value>
<value><dateTime.iso8601>20031231T13:30:25</dateTime.iso8601></value>
<value><base64>SGVsbG8sIFdvcmxkIQ==</base64></value>
<value><nil/></value>
```

Exemple de données complexes

```
<value>
  <struct>
    <member>
       <name>age</name><value><int>38</int></value>
    </member>
    <member>
      <name>name</name><value><string>Alice</string></value>
    </member>
    <member>
      <name>children</name>
      <value>
        <array>
          <data>
            <value><string>Bob</string></value>
            <value><string>Cary</string></value>
          </data>
        </array>
      </value>
    </member>
  </struct>
</value>
```

La DTD pour les requêtes et réponses

- Requête

```
<!ELEMENT methodCall (methodName, params)>
<!ELEMENT methodName (#PCDATA)>
<!ELEMENT params (param*)>
<!ELEMENT param (value)>
```

- Réponse

```
<!ELEMENT methodResponse (params|fault)>
<!ELEMENT fault (value)>
<!-- note that the content model for fault is underspecified here -->
```

Exemple de requête

```
POST /rpcxml HTTP/1.0
User-Agent: AcmeXMLRPC/1.0
Content-Type: text/xml
Content-Length: 165

<?xml version="1.0"?>
<methodCall>
  <methodName>sayHello</methodName>
  <params>
    <param><value><nil/></value></param>
    <param><value><string>ca</string></value></param>
  </params>
</methodCall>
```

Exemple de réponse

```
HTTP/1.1 200 OK
Date: Sun, 29 Apr 2003 12:08:58 GMT
Server: Apache/1.3.12 (Unix)
Connection: close
Content-Type: text/xml
Content-length: 133

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

Exemple de réponse en erreur

```
<?xml version="1.0"?>
<methodResponse>
  <fault>
    <value>
      <struct>
        <member>
          <name>faultCode</name>
          <value><int>3</int></value>
        </member>
        <member>
          <name>faultString</name>
          <value><string>No such method.</string></value>
        </member>
      </struct>
    </value>
  </fault>
</methodResponse>
```

Les spécifications fixent la significations d'un certain nombre de faultCode

Exemple de réponse en erreur

```
<?xml version="1.0"?>
<methodResponse>
  <fault>
    <value>
      <struct>
        <member>
          <name>faultCode</name>
          <value><int>401</int></value>
        </member>
        <member>
          <name>faultString</name>
          <value><string>Unknown language, ’ca’.</string></value>
        </member>
      </struct>
    </value>
  </fault>
</methodResponse>
```

Implémentations

Under Construction
En Construction

- Propose la partie cliente et/ou la partie serveur
- *Java*
  - Apache XML RPC (ex Helma)
    - *Fournit la partie cliente, une servlet et un serveur HTTP standalone pour les tests*
- *PHP*
- *PERL*
- *Python*
- *ASP (VBScript, JScript)*

Exemple de serveur (Apache XML RPC)

```
try {
    // Start the server, using built-in version
    System.out.println("Attempting to start XML-RPC Server...");
    WebServer server = new WebServer(8899);
    System.out.println("Started successfully.");
    // Register our handler class as area
    server.addHandler("hello", new HelloHandler("en"));
    System.out.println("Registered HelloHandler class to hello.");
    System.out.println("Now accepting requests. (Halt program to stop.)");
 } catch (IOException e) {
    System.out.println("Could not start server: " +e.getMessage(  ));
 }
```

Exemple de handler statique
(Apache XML RPC)

```
public class HelloHandler {
   private String currentLanguage;
   public HelloHandler(String language){setCurrentLanguage(language); }
   public String getCurrentLanguage(String language) { return this.currentLanguage; }
   public void setCurrentLanguage(String language) { this.currentLanguage=language; }
   public String sayHello(String name, String language) throws XmlRpcException {
    if(language==null) language=currentLanguage;
    if(language.equals("en")){ if(name==null) name="World";
        return "Hello "+name; }
    if(language.equals("es")){ if(name==null) name="Mundo";
        return "Hola "+name; }
    if(language.equals("fr")){ if(name==null) name="tout le monde";
        return "Bonjour "+name; }
    throw new XmlRpcException(401,"Unknown language, '"+language+"'");
   } }
```

Exemple de handler dynamique
(Apache XML RPC)

```
public class HelloDynHandler implements XmlRpcHandler {
    private HelloHandler hh;
    public HelloDynHandler(String language){
      hh=new HelloHandler(language);
    }
    public Object execute(String methodName, Vector parameters)
      throws java.lang.Exception {
        if (methodName=="sayHello") {
            String name=(String) parameters.elementAt(0);
            String lang=(String) parameters.elementAt(1);
            return hh.sayHello(name,lang);
        } else {
            throw new XmlRpcException(3,"No such method");
        }
    }
}
```

Exemple de handler dynamique avec authentification (Apache XML RPC)

Under Construction
En Construction

Exemple de client (Apache XML RPC)

```
try {
        // Create the client, identifying the server
        XmlRpcClient client = new XmlRpcClient("http://localhost:8899/");
        // Create the request parameters
        Vector params = new Vector( );
        params.addElement(null);
        params.addElement("ca");
        // Issue a request sayHello
        String result (String) = client.execute("hello.sayHello", params);
        // Report the results
        System.out.println("The server says: " + result.toString( ));
    } catch (IOException e) {
        System.out.println("IO Exception: " + e.getMessage( ));
    } catch (XmlRpcException e) {
        System.out.println("Exception within XML-RPC: " + e.getMessage( ));
    }
```

Webographie et Bibliographie

- Sites
  - http://www.xml-rpc.com
  - http://jakarta.apache.org
- Livre
  - Simon St.Laurent, Joe Johnston, Edd Dumbill , Programming Web Services with XML-RPC, Ed O'Reilly, June 2001, ISBN 0-596-00119-3