

# FahadAizaz

## Embed string TAG in struct element using Apache XML RPC Library

As per the xml rpc specifications the struct element containing a value with specifying any TAG is considered to be string.

Some servers are not configured to handle such XML data structures. Apache XML RPC Library provides a flexibility for the client code to handle such improvisations.

### Example

Standard specification considers the below mentioned XML data structure to be valid

```
<?xml version="1.0" encoding="utf-8" ?>
<methodCall><methodName>Request.myMethod</methodName>
<params>
<param>
<value>
<struct>
<member><name>Fieldname1</name><value>ANY_STRING</value></member>
<member><name>Fieldname2</name><value>ANY_STRING</value></member>
<member><name>Fieldname3</name><value><int>INTEGER</int></value></member>
<member><name>Fieldname4</name><value><dateTime.iso8601>DATE</dateTime.iso8601></value></member>
<member><name>Fieldname5</name><value><int>INTEGER</int></value></member>
<member><name>Fieldname6</name><value>ANY_STRING</value></member>
</struct>
</value>
</param>
</params>
</methodCall>
```

and equal to

```
<?xml version="1.0" encoding="utf-8" ?>
<methodCall><methodName>Request.myMethod</methodName>
<params>
<param>
<value>
<struct>
<member><name>Fieldname1</name><value><string>ANY_STRING</string></value></member>
<member><name>Fieldname2</name><value><string>ANY_STRING</string></value></member>
<member><name>Fieldname3</name><value><int>INTEGER</int></value></member>
<member><name>Fieldname4</name><value><dateTime.iso8601>DATE</dateTime.iso8601></value></member>
<member><name>Fieldname5</name><value><int>INTEGER</int></value></member>
<member><name>Fieldname6</name><value><string>ANY_STRING</string></value></member>
</struct>
</value>
</param>
</params>
</methodCall>
```

But as mentioned before some servers do not interpret both codes the same and therefore changes on the client's end can save the day.

### Code

In order to do this custom class has to be created by extending Serializer

```
public class CustomStringSerializer extends StringSerializer
{
    public void write(ContentHandler pHandler, Object pObject) throws SAXException
    {
        write((ContentHandler)pHandler, STRING_TAG, pObject.toString());
    }
}
```

STRING\_TAG does the job for you. Next custom class [TypeFactory](#) has to be created in order to realize this change. This custom [TypeFactory](#) utilizes the above mentioned Custom Serializer while XML data structure is generated.

```
public class CustomTypeFactory extends TypeFactoryImpl
{
    public MyTypeFactory(XmlRpcController pController)
    {
        super(pController);
    }
    public TypeSerializer getSerializer(XmlRpcStreamConfig pConfig, Object pObject) throws SAXException
    {
        if (pObject instanceof String)
        {
            return new CustomStringSerializer();
        }
        else
        {
            return super.getSerializer(pConfig, pObject);
        }
    }
}
```

Last step creation of client

```
HashMap data = new HashMap();
data.put("Fieldname1", new String("ANY_STRING") );
data.put("Fieldname2", new String("ANY_STRING") );
data.put("Fieldname3", new Integer(INTEGER));
data.put("Fieldname4", new Date() );
data.put("Fieldname5", new String("ANY_STRING") );
data.put("Fieldname6", new Integer(INTEGER) );
Vector params = new Vector();
params.addElement(data);

XmlRpcClientConfigImpl config = new XmlRpcClientConfigImpl();
config.setServerURL(new URL("http://localhost:8080/RPC2"));

XmlRpcClient client = new XmlRpcClient();
client.setTypeFactory(new CustomTypeFactory(client));
client.setConfig(config);
client.execute("Request.myMethod", params);
```

## References

I am new to Apache's XML RPC Library, this article could not have been written without the help from below mentioned references

[ws.apache.org/xmlrpc/advanced.html](http://ws.apache.org/xmlrpc/advanced.html)

Marcelo Grassi [Your query at nabble forum helped me write the above code]