University of Innsbruck
Bernhard FRITZ 1316136
2 November 2015

Web Services PS 703128

XML-RPC

Required software:

- Java JDK: http://www.oracle.com/technetwork/java/javase/downloads/
- Eclipse: http://www.eclipse.org/downloads/
- Apache XML-RPC library: http://ws.apache.org/xmlrpc/
- Apache XML-RPC library v2.0.1: http://sti-innsbruck.at/sites/default/files/courses/xmlrpc-2.0.1.jar

Exercise 1 – Write XML-RPC Server – Multiple Procedures

Implement an XML-RPC server for temperature conversion between three temperature units (Celsius, Fahrenheit, Kelvin) by using the library provided by Apache.

Requirements:

1. Your implementation provides all possible conversions (CelsiusToFahrenheit, FahrenheitToKelvin, ...)
2. Each procedure accepts only one parameter (input value)
3. Return one output parameter (converted value)

Exercise 2 – Write XML-RPC Server – Multiple Parameters

Re-create Exercise-1, with the following requirements:

1. Your implementation provides only one procedure (e.g. TemperatureConversion)
2. The procedure accepts at least three parameters (unit source, unit target, input value)
3. Return three output parameters (unit source, unit target, converted value)

Exercise 3 – Write XML-RPC Client

Select four different temperature conversion scenarios. Write XML-RPC client to call the procedures based on the scenarios you have selected by calling the first implementation (Exercise 1) twice and the second implementation (Exercise 2) twice.

Exercise 4 – XML-RPC Request/Response

Use the TCP/IP Monitor[1] integrated in Eclipse to capture request and response between client and server from the scenarios you have implemented in Exercise-3.

Remote procedure #1:

Request:

```
<?xml version="1.0"  standalone="no"?>
<methodCall>
<methodName>TemperatureConversion.celsiustofahrenheit</methodNam
e>
<params>
<param>
<value>
<double>20.0</double>
</value>
</param>
</params>
</methodCall>
```

Response:

```
<?xml version="1.0"  standalone="no"?>
<methodResponse>
<params>
<param>
<value>
<double>68.0</double>
</value>
</param>
</params>
</methodResponse>
```

[1]Info: http://www.avajava.com/tutorials/lessons/how-do-i-monitor-http-communication-in-eclipse.html

Remote procedure #2:

Request:

```
<?xml version="1.0"  standalone="no"?>
<methodCall>
<methodName>TemperatureConversion.fahrenheittokelvin</methodName
>
<params>
<param>
<value>
<double>68.0</double>
</value>
</param>
</params>
</methodCall>
```

Response:

```
<?xml version="1.0"  standalone="no"?>
<methodResponse>
<params>
<param>
<value>
<double>293.15</double>
</value>
</param>
</params>
</methodResponse>
```

Remote procedure #3:

Request:

```
<?xml version="1.0"  standalone="no"?>
<methodCall>
<methodName>TemperatureConversion.temperatureconversion</methodN
ame>
<params>
<param>
<value>celsius</value>
</param>
<param>
<value>fahrenheit</value>
</param>
<param>
<value>20.0</value>
</param>
</params>
</methodCall>
```

Response:

```
<?xml version="1.0"  standalone="no"?>
<methodResponse>
<params>
<param>
<value>
<array>
<data>
<value>celsius</value>
<value>fahrenheit</value>
<value>68.0</value>
</data>
</array>
</value>
</param>
</params>
</methodResponse>
```

Remote procedure #4:

Request:

```
<?xml version="1.0"  standalone="no"?>
<methodCall>
<methodName>TemperatureConversion.temperatureconversion</methodN
ame>
<params>
<param>
<value>fahrenheit</value>
</param>
<param>
<value>kelvin</value>
</param>
<param>
<value>68.0</value>
</param>
</params>
</methodCall>
```

Response:

```
<?xml version="1.0"  standalone="no"?>
<methodResponse>
<params>
<param>
<value>
<array>
<data>
<value>fahrenheit</value>
<value>kelvin</value>
<value>293.15</value>
</data>
</array>
</value>
</param>
</params>
</methodResponse>
```