Wireless Embedded Systems and Networking

Lab Day 2:
Part 2: Custom IP/USN application via Web Services

Lab Assistant: Jaein Jeong
University of California, Berkeley

Compare REST and SOAP

• REST (Representational State Transfer) API
  – Clients access Gateway through HTTP GET or POST.
  – Response is in XML and can be parsed by clients.
  – Easy to start with: just needs web browser.

• SOAP (Simple Object Access Protocol) API
  – Clients access Gateway using host programming language
    (e.g. java, php, C/C++) through SOAP library.
  – Response is parsed into host PL format by SOAP library.
  – Takes some time to start, but easier to integrate with host
    programming language.
Some Examples of REST

Assuming the Gateway Server is 192.168.0.2

• Get the Last Temp. Readings from the Sensors
  – http://192.168.0.2/gw/rest/V1/?method=events.readLast&name=TemperatureReadEvent

• Change the Sensor Sample Period
  – http://192.168.0.2/gw/rest/V1/?method=attributes.get&name=SamplePeriod
  – http://192.168.0.2/gw/rest/V1/?method=attributes.set&name=SamplePeriod&value=300000

• Ping a Node
  – http://192.168.0.2/gw/rest/V1/?method=mgmt.pingNode.addr=ffffffffffffffff

List of REST API

• http://192.168.0.2/developer/gw/rest/
Using SOAP

- Host Programming Language to Use
  - Java, php, C/C++, C#, perl, python
- Step 1: Install Development Kit
  - JDK, Php development kit, C/C++ compiler
- Step 2: Install SOAP library and client stubs
- Step 3: Accessing SOAP in host programming language.

Using SOAP with Java

- Step 1: Install JDK
- Step 2a: Install SOAP library
  - Apache Axis, commons-httpclient-3.0-rc3.jar, commons-codec-1.3.jar
  - Add library to CLASSPATH.
- Step 2b: Create SOAP client stubs
  - Java org.apache.axis.wsd1.WSDL2Java
- Step 3a: Acquiring gateway port
  - GWServiceLocator locator = new GWServiceLocator();
  - GWPort proxy = locator.getGWPort(portURL);
- Step 3b: Sending a request to gateway
  - GW__PingNode_Result[] results;
  - results = proxy.mgmtPingNode(addr, 1, TIMEOUT, "**");
  - System.out.println(results[i].getLongAddr());
Using SOAP with PHP

- Step 1, 2: Install PHP development kit
  - Select SOAP as an external component.
  - Do not select other external components (bug?).
- Step 3a: Acquiring gateway port
  - $url = http://gateway/gw/rest/V1/?method=gw.getWSDL;
  - $proxy = new SoapClient($url);
- Step 3b: Sending a request to gateway
  - $ret = $proxy->mgmtPingNode("ffffffffffffffff", 2, "*");

SOAP API

- http://192.168.0.2/developer/gw/soap/
Example: Get the last temperature readings from sensors (Java)

- Corresponding REST call
  - http://192.168.0.2/gw/rest/V1/?method=events.readLast&name=TemperatureReadEvent

- Start from a sample application (e.g. Get.java)
- Modify the method invocation.
  
  ```java
  results = proxy.attributesGet(name, addr, TIMEOUT, "*");
  ```

Example: Get the last temperature readings from sensors (Java)

- Find the name of method.
  - http://192.168.0.2/developer/gw/

- Find the signature for the method.

  ```bash
  $ grep -n "eventsReadLast\(\)" java
  GWPort.java:27:   public gw.GW__eventsReadLast_Result eventsReadLast(java.lang.String name, java.lang.String addr)
  ```

- Declare any data structure required.
Example: Get the last temperature readings from sensors (Java)

```java
public void get(String name, String addr) {
    GW__eventsReadLast_Result result;
    GW__Event_Result[] results;
    GW__Event value;
    try {
        result = proxy.eventsReadLast(name, addr);
        results = result.getResults();
        for (int i = 0; i < results.length; i++) {
            value = results[i].getValue();
            System.out.println("Mote " + results[i].getAddr() + " Time " + results[i].getTimestamp() + " TemperatureReadEvent " + name + " " + value.getTemperatureReadEvent());
        }
    } catch (Exception e) {
        e.printStackTrace(); System.exit(2);
    }
}
```

1. Intermediate data type declaration
2. Method Invocation
3. Reading Results

Example: Get the last temperature readings from sensors (Java)

- Results

```
$ java GetLastTemperature 192.168.0.2
- Unable to find required classes (javax.activation.DataHandler and javax.mail.internet.MimeMultipart). Attachment support is disabled.
Mote 00173b00f0000100 Time 1.18355699518779E9 TemperatureReadEvent 7246
Mote 00173b0010241593 Time 1.18255699178323E9 TemperatureReadEvent 7297
```
Example: Get the last temperature readings from sensors (PHP)

- REST call
  - http://192.168.0.2/gw/rest/V1/?method=events.readLast&name=TemperatureReadEvent

- Find method as in Java.
  - Same as Java
    » Method name, member variable name
  - Different from Java
    » No explicit type declaration.
    » Member variables are accessed directly, not through methods as in Java.

```php
function get($gateway, $attrName, $mote)
{
    $url = "http://" . $gateway . "/gw/rest/V1/?method=gw.getWSDL";
    $proxy = new SoapClient($url);
    $ret = $proxy->eventsReadLast($attrName, $mote);
    $ret_arr = $ret->results;
    foreach ($ret_arr as $result) {
        echo "Mote " . $result->addr .
               " Time " . $result->timestamp .
               " Name " . $attrName .
               " Value " . $result->value->TemperatureReadEvent . "\n";
    }
}
```

$ php get_last_temperature.php 192.168.0.2
Mote 00173b000fecb28f Time 1182556899.4362 TemperatureReadEvent 7252
Mote 00173b0010241593 Time 1182556899.1605 TemperatureReadEvent 7304
Exercises

- Write at least two applications either in Java or PHP.
- Download skeleton code from the built server.
  - Java: Ex2_1.java, Ex2_2.java, …, Ex2_6.java
  - PHP: ex2_1.php, ex2_2.php, …, ex2_6.php
- Complete the code by filling in the TODO in the skeleton code.

Exercise-1

- Get the Names of Nodes
  - REST Call:
    http://192.168.0.2/gw/rest/V1/?method=metadata.get&name=name

```$ java Ex2_1 192.168.0.2 (or php ex2_1.php 192.168.0.2)
- Unable to find required classes (javax.activation.DataHandler and
  javax.mail.internet.MimeMultipart). Attachment support is disabled.
Mote 00173b000c827501 Name 1fourteen
Mote 00173b000c834627 Name 1twelve
Mote 00173b000c847d22 Name 1thirteen
Mote 00173b000fecb28f Name Node 00173b000fecb28f
Mote 00173b0010203e21 Name 2two
Mote 00173b00102412ef Name 6six
Mote 00173b0010241593 Name Node 00173b0010241593
Mote 00173b00102426d2 Name 1one
Mote 00173b00102430f7 Name 3three
Mote ffffffffffffffffe Name SnIPSnap```
Exercise-2

• Get the Locations of the Nodes
  - http://192.168.0.2/gw/rest/V1/?method=metadata.get&name=map-x
  - http://192.168.0.2/gw/rest/V1/?method=metadata.get&name=map-y

$ java Ex2_2 192.168.0.2 (or php ex2_2.php 192.168.0.2)
- Unable to find required classes (javax.activation.DataHandler and java.mail.internet.MimeMultipart).
- Attachment support is disabled.
- Mote 00173b000c827501 map-x 170
- Mote 00173b000c834627 map-x 20
- Mote 00173b000c847d22 map-x 170
- Mote 00173b000fecz28f map-x 180
- Mote 00173b0010203e21 map-x 90
- Mote 00173b00102412ef map-x 170
- Mote 00173b0010241593 map-x 185
- Mote 00173b00102426d2 map-x 90
- Mote 00173b00102430f7 map-x 330
- Mote fffffffffffffffe map-x 20
- Mote 00173b000c827501 map-y 270
- Mote 00173b000c834627 map-y 420
- Mote 00173b000c847d22 map-y 355
- Mote 00173b000fecz28f map-y 170
- Mote 00173b0010203e21 map-y 270
- Mote 00173b00102412ef map-y 180
- Mote 00173b0010241593 map-y 345
- Mote 00173b00102426d2 map-y 180
- Mote 00173b00102430f7 map-y 100
- Mote fffffffffffffffe map-y 380

Exercise-3

• Get All Data in a Recent Time Window (1 day)
  - http://192.168.0.2/gw/rest/V1/?method=events.readRelative&t1Offset=-86400&t2Offset=0

$ java Ex2_3 192.168.0.2 (or php ex2_3.php 192.168.0.2)
- Unable to find required classes (javax.activation.DataHandler and java.mail.internet.MimeMultipart).
- Attachment support is disabled.
- Mote 00173b0010241593 timestamp 1.18247200152407E9 TemperatureReadEvent 7281
- Mote 00173b000fecz28f timestamp 1.18247200164372E9 TemperatureReadEvent 7227
- Mote 00173b0010241593 timestamp 1.18247200151439E9 TemperatureReadEvent 7281
- Mote 00173b000fecz28f timestamp 1.18247200151439E9 TemperatureReadEvent 7225
- TemperatureReadEvent 7225
- ...
Exercise-4

• Request a Sample from All Enabled Sensors
  – http://192.168.0.2/gw/rest/V1/?method=rpc.execute&name=sampleRequest

```
$ php ex2_4.php 192.168.0.2 (or java Ex2_4 192.168.0.2)
Mote 00173b000fceb28f Time 1182558375.7276
Mote 00173b000c834627 Time 1182558375.7276
Mote 00173b0010203e21 Time 1182558375.7276
Mote 00173b00102426d2 Time 1182558375.7276
Mote 00173b000c827501 Time 1182558375.7276
Mote 00173b000c847d22 Time 1182558375.7276
Mote 00173b00102430f7 Time 1182558375.7276
Mote 00173b0010241593 Time 1182558375.7276
Mote 00173b00102412ef Time 1182558375.7276
```

Exercise-5

• Get the Battery Voltage of the Nodes
  – http://192.168.0.2/gw/rest/V1/?method=events.readLast&name=NodeStatsEvent

```
$ php ex2_5.php 192.168.0.2 (or java Ex2_5 192.168.0.2)
Mote 00173b00102426d2 Time 1182558331.1052 Voltage 2999
Mote 00173b000c847d22 Time 1182558222.7587 Voltage 2965
Mote 00173b0010203e21 Time 1182558420.274 Voltage 2999
Mote 00173b00102430f7 Time 1182558324.6725 Voltage 2999
Mote 00173b000c827501 Time 1182558419.6195 Voltage 2999
Mote 00173b000fceb28f Time 1182558256.1617 Voltage 2999
Mote 00173b000c834627 Time 1182558267.2399 Voltage 2999
Mote 00173b0010241593 Time 1182558359.115 Voltage 2852
Mote 00173b00102412ef Time 1182558402.1318 Voltage 2999
```
Exercise-6

• Get Statistics on Node Traffic and Network Reliability
  – http://192.168.0.2/gw/rest/V1/?method=events.readRelative&name=NodeStatsEvent&t1Offset=-86400&t2Offset=0

$ php ex2_6.php 192.168.0.2 (or java Ex2_6 192.168.0.2)
Mote 00173b000c827501 timestamp 1182472228.9383 sent 1512 delivered 1428
Mote 00173b00102426d2 timestamp 1182472229.3708 sent 1477 delivered 1425
Mote 00173b0010241593 timestamp 1182472231.9069 sent 1783 delivered 1745
Mote 00173b000fecd28f timestamp 1182472263.1175 sent 2222 delivered 2098
Mote 00173b000c834627 timestamp 1182472278.3247 sent 1471 delivered 1422
Mote 00173b00102412ef timestamp 1182472349.9155 sent 1503 delivered 1433
Mote 00173b00102430f7 timestamp 1182472352.6 sent 1476 delivered 1415

TO-DO list for PHP Web Service

• Install PHP
  – Tested with php-5.2.3-win32-installer.

• Install Apache Web Server
  – Tested with apache_2.0.59-win32-x86-no_ssl.
  – Configure httpd.conf file to active PHP in the web server.
    # For PHP 5 do something like this:
    LoadModule php5_module "c:/Program Files/PHP/php5apache2.dll"
    AddType application/x-httpd-php .php
    # configure the path to php.ini
    PHPIniDir "C:/Program Files/PHP"
    – Move PHP source code to the web server root directory.
      » C:\Program Files\Apache Group\Apache2\htdocs
**PHP Web Service Example**

- Connects to the gateway server.

**PHP Web Service Tutorial**

Get last temperature reading

- Lists sensor reading for a node

**CMD PHP vs. web-based PHP**

<table>
<thead>
<tr>
<th>Command Line PHP</th>
<th>Web-based PHP</th>
</tr>
</thead>
</table>
| **Command Line Passing** | **Named parameters.**  
E.g. $gateway = $argv[1];  
E.g. Caller: <a href="node.php?server=$server&ip=$ip">$ip</a>  
E.g. Callee: $server = $_GET['server'];  
**Etc**  
Nested variables allowed.  
E.g: echo $result->value->NodeStatsEvent->voltage;  
**Etc**  
Nested variables should be declared in cascade for access.  
E.g. $value = $result->value;  
$event = $value->NodeStatsEvent;  
echo $event->voltage; |  
| **Debugging** | Both program result and error on console.  
Program result on web browser.  
Error on web server error log. |
PHP Web Service Example

- Exercise 2-7: Modify `get_last_temperature_web.php` so that it displays temperature in Celsius rather than Fahrenheit.

Get Last Temperature Reading

<table>
<thead>
<tr>
<th>Mote</th>
<th>Time</th>
<th>Temperature</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>00173b000f0e0f0f</td>
<td>Tue 26 Jun 20 24:17 -0700</td>
<td>77.25 °F</td>
<td>F</td>
</tr>
<tr>
<td>00173b0010241f50</td>
<td>Tue 26 Jun 20 22:44 -0700</td>
<td>71.46 °F</td>
<td>F</td>
</tr>
</tbody>
</table>

Get Last Temperature Reading

<table>
<thead>
<tr>
<th>Mote</th>
<th>Time</th>
<th>Temperature</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>00173b000f0e0f0f</td>
<td>Tue 26 Jun 20 44:17 -0700</td>
<td>21.05 °C</td>
<td>C</td>
</tr>
<tr>
<td>00173b0010241f50</td>
<td>Tue 26 Jun 20 42:44 -0700</td>
<td>21.05 °C</td>
<td>C</td>
</tr>
</tbody>
</table>