**Addendum to Detailed Design Document**

**Project**:

**Scoop**: UCCX (HA) XML interaction with RESTful Web Service

**Issue**: Create URL Document - Post Feature unable to interact with Web Service adequately

**Date**: 10/8/11

**Concept Testing In Lab**

In lab, a similar RESTful web service was created using myeclipseide as below.

<http://192.168.1.30:8080/restdemo/services/customers>

|  |  |
| --- | --- |
| Customer.java | CustomersResource.java |
| package com.myeclipseide.ws;import javax.xml.bind.annotation.XmlRootElement;@XmlRootElementpublic class Customer { private int id; private String name; private String address;  public int getId(){ return id; }  public void setId(int id) { this.id = id; }  public String getName() { return name; }  public void setName(String name) { this.name = name; } public String getAdress(){ return address; } public void setAddress(String address) { this.address = address; }} | package com.myeclipseide.ws;import java.util.ArrayList;import java.util.List;import java.util.TreeMap;import javax.ws.rs.Consumes;import javax.ws.rs.GET;import javax.ws.rs.POST;import javax.ws.rs.Path;import javax.ws.rs.PathParam;import javax.ws.rs.Produces;import com.sun.jersey.spi.resource.Singleton;@Produces("application/xml")@Path("customers")@Singletonpublic class CustomersResource { private TreeMap<Integer, Customer> customerMap = new TreeMap<Integer, Customer>();  public CustomersResource() { // hard code one customer to demo Customer customer = new Customer(); customer.setName("Harold Abernathy"); customer.setAddress("Sheffield, UK"); addCustomer(customer); } @GET public List<Customer> getCustomers() { List<Customer> customers = new ArrayList<Customer>(); customers.addAll(customerMap.values()); return customers;  } @GET @Path("{id}") public Customer getCustomer(@PathParam("id") int cId) { return customerMap.get(cId); } @POST @Path("add") @Produces("text/html") @Consumes("application/xml") public String addCustomer(Customer customer) { int id = customerMap.size(); customer.setId(id); customerMap.put(id, customer); return "Customer " + customer.getName() + " added with Id " +id; }} |

We can browse to the lab WebService to see the hardcoded PUT. See below.



We can also POST another entry using the Firefox POSTER tool. Getting the following document returned.

|  |  |
| --- | --- |
|  |  |

Again we can confirm the Add was completed.



Similar to the POSTER tool, UCCX has the step, ***Create URL Document***. However we are limited in that there is no free form content entry point, as in the POSTER tool. Instead we are limited to parameters.



We could also use the ***Get HTTP Contact Info*** step in UCCX. However here we are limited to a query string.



Attempts to use the POST method with parameters are rejected by the lab RESTful engine.

|  |  |
| --- | --- |
|  |  |