**WebService Test Plan**

You will create five users that send requests to One page. Also, you will tell the users to run their tests twice. So, the total number of requests is (5 users) x (1 requests) x (repeat 2 times) = 10 HTTP requests. To construct the Test Plan, you will use the following elements: [Thread Group](http://jmeter.apache.org/usermanual/test_plan.html#thread_group), [HTTP Request](http://jmeter.apache.org/usermanual/component_reference.html#HTTP_Request), and [Aggregate Graph](http://jmeter.apache.org/usermanual/component_reference.html#Aggregate_Graph).

If the sampler appears to be getting an error from the webservice, double check the SOAP message and make sure the format is correct. In particular, make sure the xmlns attributes are exactly the same as the WSDL. If the xml namespace is different, the webservice will likely return an error.

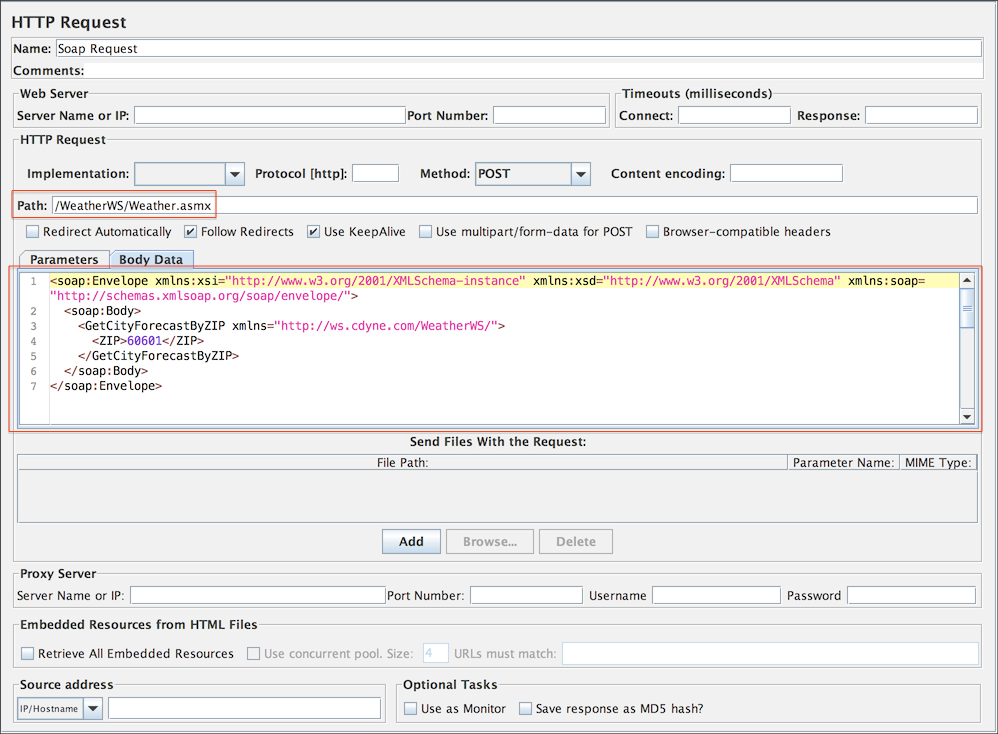
**Creating WebService Test Plan**

In our Test Plan, we will use a .NET webservice. We won't go into the details of writing a webservice. If you don't know how to write a webservice, google for webservice and familiarize yourself with writing webservices for Java and .NET. It should be noted there is a significant difference between how .NET and Java implement webservices. The topic is too broad to cover in the user manual. Please refer to other sources to get a better idea of the differences.

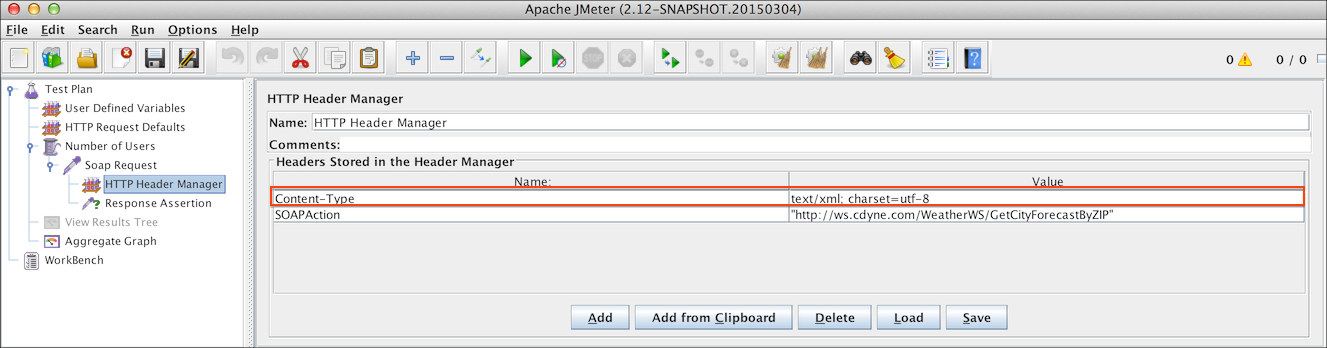
JMeter sends requests in the order that they appear in the tree.

Start by using menu File > "Templates..." and select template "Building a SOAP Webservice Test Plan". Then, click "Create" button.

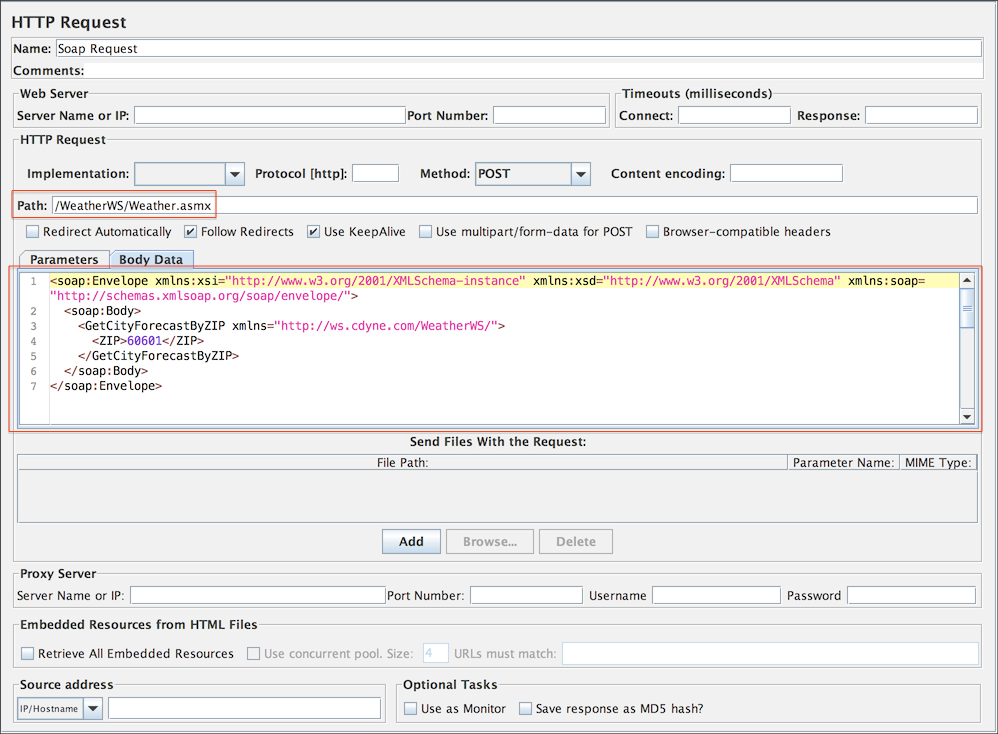
[
Figure 10.1.0. Webservice Template](http://jmeter.apache.org/images/screenshots/ws_template.png)

1. In "HTTP Request Defaults" change "Server Name of IP"
2. In "Soap Request", change "Path:"[](http://jmeter.apache.org/images/screenshots/ws_http_request.png)

Next, select "HTTP Header Manager" and update "SOAPAction" header to match your webservice. Some webservices may not use SOAPAction in this case remove it.  
Currently, only .NET uses SOAPAction, so it is normal to have a blank SOAPAction for all other webservices. The list includes JWSDP, Weblogic, Axis, The Mind Electric Glue, and gSoap.

[](http://jmeter.apache.org/images/screenshots/ws_header.png)

The last step is to paste the SOAP message in the "Body Data" text area.

[](http://jmeter.apache.org/images/screenshots/ws_http_request.png)

**Adding Users**

The [Thread Group](http://jmeter.apache.org/usermanual/test_plan.html#thread_group) tells JMeter the number of users you want to simulate, how often the users should send requests, and the how many requests they should send.

Select the Thread Group element in the tree, if you have not already selected it. You should now see the Thread Group Control Panel in the right section of the JMeter window

[
Figure 10.2. Thread Group with Default Values](http://jmeter.apache.org/images/screenshots/webtest/threadgroup.png)

Start by providing a more descriptive name for our Thread Group. In the name field, enter JMeter Users.

Next, increase the number of users (called threads) to 10.

In the next field, the Ramp-Up Period, leave the the default value of 0 seconds. This property tells JMeter how long to delay between starting each user. For example, if you enter a Ramp-Up Period of 5 seconds, JMeter will finish starting all of your users by the end of the 5 seconds. So, if we have 5 users and a 5 second Ramp-Up Period, then the delay between starting users would be 1 second (5 users / 5 seconds = 1 user per second). If you set the value to 0, then JMeter will immediately start all of your users.

Finally, clear the checkbox labeled "Forever", and enter a value of 2 in the Loop Count field. This property tells JMeter how many times to repeat your test. If you enter a loop count value of 0, then JMeter will run your test only once. To have JMeter repeatedly run your Test Plan, select the Forever checkbox.

In most applications, you have to manually accept changes you make in a Control Panel. However, in JMeter, the Control Panel automatically accepts your changes as you make them. If you change the name of an element, the tree will be updated with the new text after you leave the Control Panel (for example, when selecting another tree element).

[
Figure 10.3. JMeter Users Thread Group](http://jmeter.apache.org/images/screenshots/webtest/threadgroup2.png)

**Adding a Listener to View Store the Test Results**

The final element you need to add to your Test Plan is a [Listener](http://jmeter.apache.org/usermanual/component_reference.html#listeners). This element is responsible for storing all of the results of your HTTP requests in a file and presenting a visual model of the data.

Select the JMeter Users element and add a [Aggregate Graph](http://jmeter.apache.org/usermanual/component_reference.html#Aggregate_Graph) listener (Add --> Listener --> Aggregate Graph). Next, you need to specify a directory and filename of the output file. You can either type it into the filename field, or select the Browse button and browse to a directory and then enter a filename.

[
Figure 10.4. Graph Results Listener](http://jmeter.apache.org/images/screenshots/ws_listener.png)

**Rest Webservice**

Testing a REST Webservice is very similar as you only need to modify in HTTP Request

* Method: to select the one you want to test
* Body Data: which can be JSON, XML or any custom text

You may also need to modify "HTTP Header Manager" to select the correct "Content-Type"

**Building a JMS Point-to-Point Test Plan**

The setup of the test is 1 thread group with 5 threads sending 4 messages each through a request queue. A fixed reply queue will be used for monitoring the reply messages. To construct the Test Plan, you will use the following elements: [Thread Group](http://jmeter.apache.org/usermanual/test_plan.html#thread_group), [JMS Point-to-Point](http://jmeter.apache.org/usermanual/component_reference.html#JMS_Point-to-Point), and [Graph Results](http://jmeter.apache.org/usermanual/component_reference.html#Graph_Results).

General notes on JMS: There are currently two JMS samplers. One uses JMS topics and the other uses queues. Topic messages are commonly known as pub/sub messaging. Topic messaging is generally used in cases where a message is published by a producer and consumed by multiple subscribers. A JMS sampler needs the JMS implementation jar files; for example, from Apache ActiveMQ.

**Adding a Thread Group**

The first step you want to do with every JMeter Test Plan is to add a [Thread Group](http://jmeter.apache.org/usermanual/test_plan.html#thread_group) element. The Thread Group tells JMeter the number of users you want to simulate, how often the users should send requests, and the how many requests they should send.

Go ahead and add the ThreadGroup element by first selecting the Test Plan, clicking your right mouse button to get the Add menu, and then select Add --> ThreadGroup.

You should now see the Thread Group element under Test Plan. If you do not see the element, then "expand" the Test Plan tree by clicking on the Test Plan element.

Next, you need to modify the default properties. Select the Thread Group element in the tree, if you have not already selected it. You should now see the Thread Group Control Panel in the right section of the JMeter window (see Figure 11.1 below)

[
Figure 11.1. Thread Group with Default Values](http://jmeter.apache.org/images/screenshots/webtest/threadgroup.png)

Start by providing a more descriptive name for our Thread Group. In the name field, enter Point-to-Point.

Next, increase the number of users (called threads) to 5.

In the next field, the Ramp-Up Period, leave set the value to 0 seconds. This property tells JMeter how long to delay between starting each user. For example, if you enter a Ramp-Up Period of 5 seconds, JMeter will finish starting all of your users by the end of the 5 seconds. So, if we have 5 users and a 5 second Ramp-Up Period, then the delay between starting users would be 1 second (5 users / 5 seconds = 1 user per second). If you set the value to 0, then JMeter will immediately start all of your users.

Clear the checkbox labeled "Forever", and enter a value of 4 in the Loop Count field. This property tells JMeter how many times to repeat your test. If you enter a loop count value of 0, then JMeter will run your test only once. To have JMeter repeatedly run your Test Plan, select the Forever checkbox.

**Note:** In most applications, you have to manually accept changes you make in a Control Panel. However, in JMeter, the Control Panel automatically accepts your changes as you make them. If you change the name of an element, the tree will be updated with the new text after you leave the Control Panel (for example, when selecting another tree element).

**Adding JMS Point-to-Point Sampler**

Start by adding the sampler [JMS Point-to-Point](http://jmeter.apache.org/usermanual/component_reference.html#JMS_Point-to-Point) to the Point-to-Point element (Add --> Sampler --> JMS Point-to-Point). Then, select the JMS Point-to-Point sampler element in the tree. In building the example a configuration will be provided that works with ActiveMQ 3.0.

| **Name** | **Value** | **Description** |
| --- | --- | --- |
| **JMS Resources** | | |
| QueueuConnectionFactory | ConnectionFactory | This is the default JNDI entry for the connection factory within active mq. |
| JNDI Name Request Queue | Q.REQ | This is equal to the JNDI name defined in the JNDI properties. |
| JNDI Name Reply Queue | Q.RPL | This is equal to the JNDI name defined in the JNDI properties. |
| **Message Properties** | | |
| Communication Style | Request Response | This means that you need at least a service running outside of JMeter and that will respond to the requests. This service must listen to the Request Queue and send messages to the queue referenced by themessage.getJMSReplyTo() |
| Content | test | This is just the content of the message. |
| JMS Properties |  | Nothing needed for active mq. |
| **JNDI Properties** | | |
| InitialContextFactory | org.apache.activemq.jndi.ActiveMQInitialContextFactory | The standard InitialContextFactory for Active MQ |
| **Properties** | | |
| queue.Q.REQ | example.A | This defines a JNDI name Q.REQ for the request queue that points to the queue example.A |
| queue.Q.RPL | example.B | This defines a JNDI name Q.RPL for the reply queue that points to the queue example.B |
| **Provider URL** | | |
| Provider URL | tcp://localhost:61616 | This defines the URL of the active mq messaging system. |

**Adding a Listener to View Store the Test Results**

The final element you need to add to your Test Plan is a [Listener](http://jmeter.apache.org/usermanual/component_reference.html#listeners). This element is responsible for storing all of the results of your JMS requests in a file and presenting a visual model of the data.

Select the Thread Group element and add a [Graph Results](http://jmeter.apache.org/usermanual/component_reference.html#Graph_Results) listener (Add --> Listener --> Graph Results). Next, you need to specify a directory and filename of the output file. You can either type it into the filename field, or select the Browse button and browse to a directory and then enter a filename.

[
Figure 11.2. Graph Results Listener](http://jmeter.apache.org/images/screenshots/graph_results.png)

s