

# Utilización del conector SOAP para llamar a un servicio web con WebSphere Lombardi Edition

## Documentación del producto

### Resumen

El conector de servicio web listo para usar en WebSphere Lombardi Edition es muy útil porque abstrae toda la complejidad de un servicio web y requiere muy pocas entradas para funcionar.

### Contenido

La desventaja de esta abstracción es que este conector es inflexible, por lo que no funciona bien con servicios web no estándar. En estas situaciones, tendrá que conectarse a los servicios web "a mano", utilizando el conector SOAP de WebSphere Lombardi Edition.

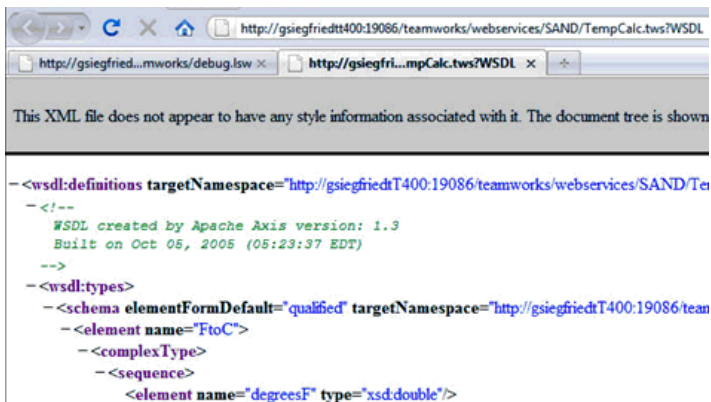
Estas instrucciones asumen que tiene acceso a un WSDL en funcionamiento y que puede realizar una llamada al servicio web utilizando soapUI u otra herramienta SOAP. Para ser exhaustivos, estas instrucciones describen cómo probar el servicio web en soapUI.

### Verifique que su servicio web esté funcionando

Verifique que tenga una URL WSDL activa.

Pegue la URL en una ventana del navegador y debería ver algo como la siguiente captura de pantalla.

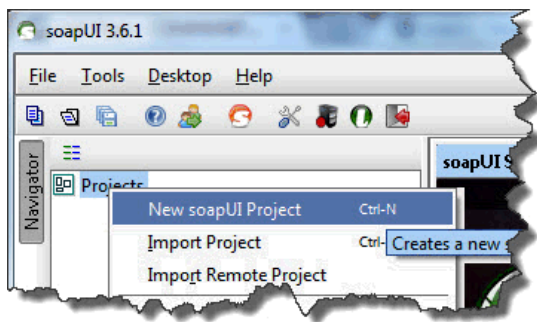
**Nota :** Es posible que deba agregar? WSDL al final de su URL.



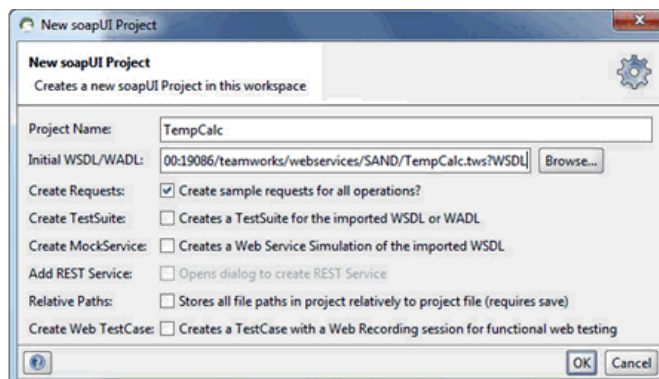
Si la URL no funciona, es probable que no tenga la configuración de red correcta para acceder al servicio web.

Instale **soapUI** usando la configuración predeterminada.

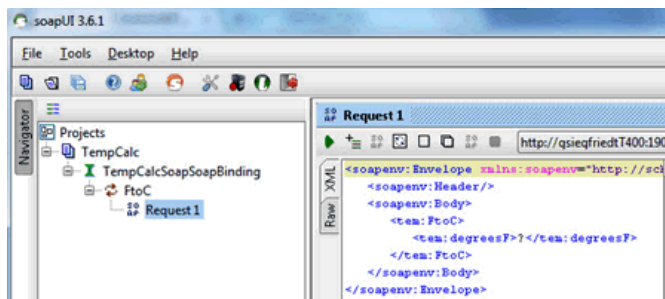
Haga clic con el botón derecho en **Proyectos** y haga clic en **Nuevo proyecto de soapUI** como se muestra en la siguiente captura de pantalla.



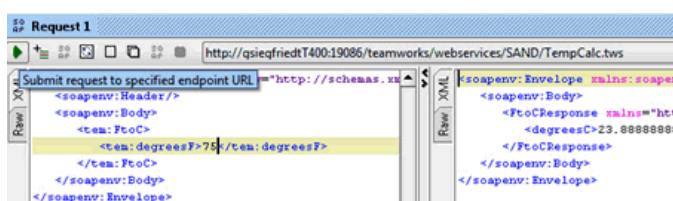
Pegue su URL de WSDL en el cuadro etiquetado **WSDL / WADL inicial** . No cambie los demás valores y opciones y haga clic en **Aceptar** .



Ahora debería tener una llamada de muestra para su servicio web donde puede realizar llamadas de muestra basadas en la entrada XML.

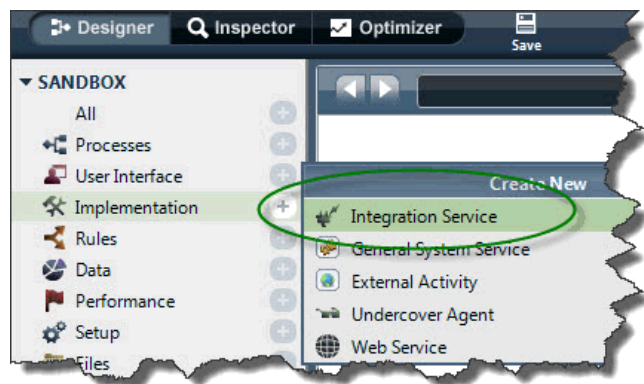


Pruebe el servicio web reemplazando los signos de interrogación con entradas de datos reales y presione **Reproducir** . Hay sugerencias que proporciona soapUI en lugares donde existen entradas repetidas o entradas opcionales. Para estas entradas, eliminar elementos no debería ser un problema.



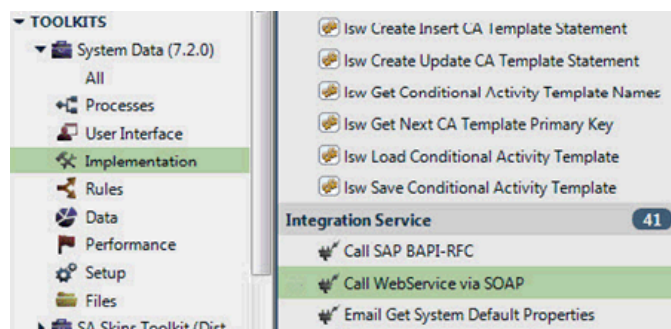
Llame a su servicio web dentro de WebSphere Lombardi Edition

Cree un nuevo servicio de integración.



Asígnele un nombre y haga clic en **Finalizar**.

Localice el componente Call WebService via SOAP en System Data Toolkit y arrástrelo a su diagrama.



Arrastre un Scriptlet de servidor desde su paleta del lado derecho a su diagrama justo antes del componente SOAP.

Conecte las líneas para que coincidan con el siguiente diagrama de captura de pantalla:

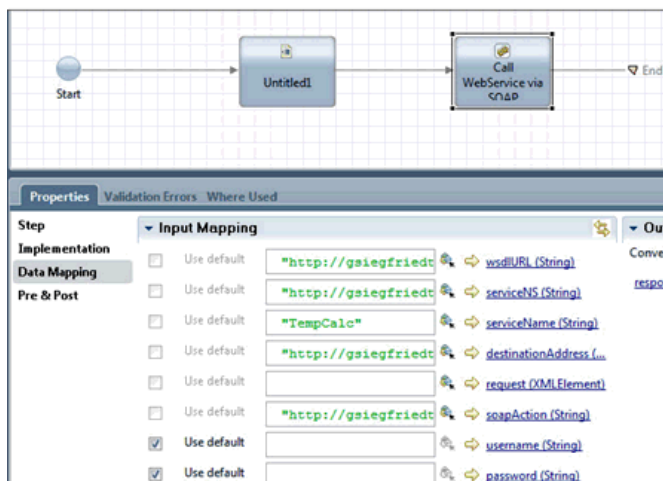


Seleccione el componente **Call WebService via SOAP** y abra la hoja de propiedades Data Mapping como se muestra en la siguiente captura de pantalla. Debería poder identificar todas las entradas del Mapeo de datos, excepto la solicitud, comparando su WSDL en una ventana del navegador con la siguiente captura de pantalla. Resultado: la entrada de la solicitud incluirá sus entradas variables. Este ejemplo utiliza el scriptlet del servidor para crear esta entrada XML. Complete los siguientes pasos:

```

<?xml version="1.0" encoding="utf-8" ?>
<definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
xmlns:s="http://www.w3.org/2001/XMLSchema" xmlns:s0="http://swanandmokashi.com"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
targetNamespace="http://swanandmokashi.com" >
  <types>
    <message name="GetQuotesSoapIn">
      <part name="GetQuotesSoapIn" type="s0:StockQuotesSoapIn" />
    </message>
    <message name="GetQuotesSoapOut">
      <part name="GetQuotesSoapOut" type="s0:StockQuotesSoapOut" />
    </message>
    <message name="GetQuotesHttpGetIn">
      <part name="GetQuotesHttpGetIn" type="s0:StockQuotesHttpGetIn" />
    </message>
    <message name="GetQuotesHttpGetOut">
      <part name="GetQuotesHttpGetOut" type="s0:StockQuotesHttpGetOut" />
    </message>
    <message name="GetQuotesHttpPostIn">
      <part name="GetQuotesHttpPostIn" type="s0:StockQuotesHttpPostIn" />
    </message>
    <message name="GetQuotesHttpPostOut">
      <part name="GetQuotesHttpPostOut" type="s0:StockQuotesHttpPostOut" />
    </message>
    <portType name="StockQuotesSoap">
      <binding name="StockQuotesSoap" type="s0:StockQuotesSoap" />
    </portType>
    <portType name="StockQuotesHttpGet">
      <binding name="StockQuotesHttpGet" type="s0:StockQuotesHttpGet" />
    </portType>
    <portType name="StockQuotesHttpPost">
      <binding name="StockQuotesHttpPost" type="s0:StockQuotesHttpPost" />
    </portType>
    <binding name="StockQuotesSoap" type="s0:StockQuotesSoap" />
    <binding name="StockQuotesHttpGet" type="s0:StockQuotesHttpGet" />
    <binding name="StockQuotesHttpPost" type="s0:StockQuotesHttpPost" />
    <service name="StockQuotes">
      <port name="StockQuotesSoap" binding="s0:StockQuotesSoap" />
      <port name="StockQuotesHttpGet" binding="s0:StockQuotesHttpGet" />
      <port name="StockQuotesHttpPost" binding="s0:StockQuotesHttpPost" />
    </service>
  </types>
  <operation name="GetStockQuotes">
    <input name="GetQuotes" type="s0:StockQuotesSoapIn" />
    <output name="GetQuotes" type="s0:StockQuotesSoapOut" />
    <binding name="StockQuotesSoap" type="s0:StockQuotesSoap" />
  </operation>
  <operation name="GetStockQuotesHttpGet">
    <input name="GetQuotesHttpGet" type="s0:StockQuotesHttpGetIn" />
    <output name="GetQuotesHttpGet" type="s0:StockQuotesHttpGetOut" />
    <binding name="StockQuotesHttpGet" type="s0:StockQuotesHttpGet" />
  </operation>
  <operation name="GetStockQuotesHttpPost">
    <input name="GetQuotesHttpPost" type="s0:StockQuotesHttpPostIn" />
    <output name="GetQuotesHttpPost" type="s0:StockQuotesHttpPostOut" />
    <binding name="StockQuotesHttpPost" type="s0:StockQuotesHttpPost" />
  </operation>
</definitions>

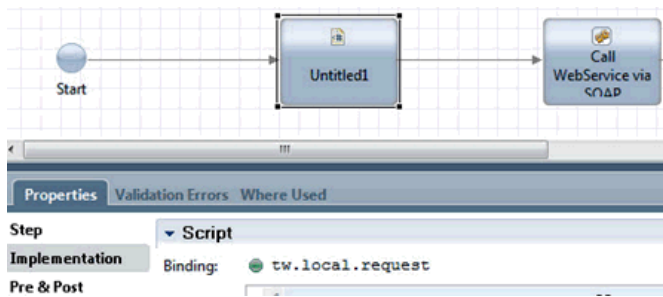
```



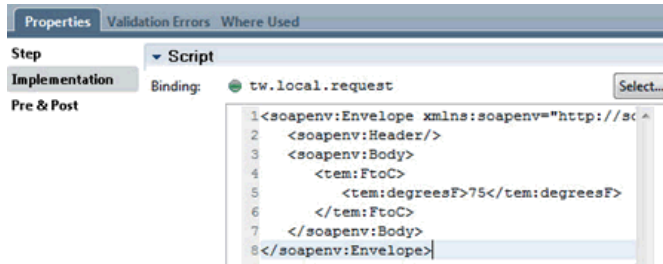
Abra la pestaña de variables y cree una nueva variable privada llamada solicitud, del tipo XMLElement.

Regrese a la vista de diagrama y cambie el nombre del scriptlet del servidor a Establecer solicitud.

Abra la hoja de propiedades de Implementación para el scriptlet del servidor y vincúlela a su variable de solicitud.

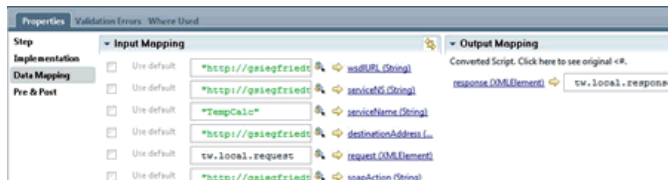


Copie toda la entrada XML de soapUI y péguela en la implementación del scriptlet del servidor.



Vincule su variable de solicitud a la entrada de solicitud del conector SOAP.

Cree una variable denominada respuesta de tipo XMLElement y vincúlela a la salida del conector SOAP.

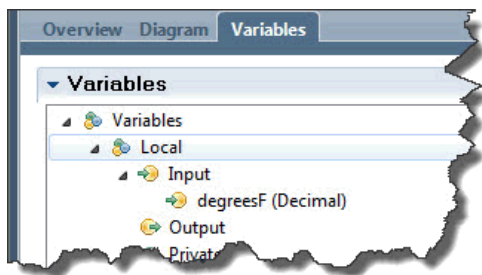


En este punto, debería poder probar su servicio utilizando sus valores codificados.

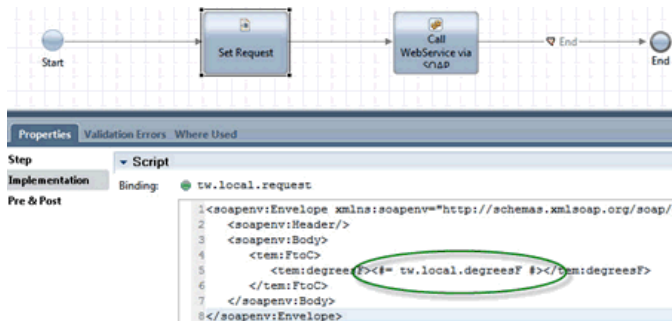
Ejecute el servicio en modo de depuración para ver la respuesta colocada en su variable de respuesta.

Si funcionó correctamente, está listo para agregar variables de entrada a su servicio y asignarlas a su variable de solicitud en el scriptlet del servidor. El siguiente ejemplo solo tiene una entrada:

Agregue entradas a su servicio.



Utilice la notación `<# = #>` para incluir JavaScript™ en el scriptlet de su servidor.

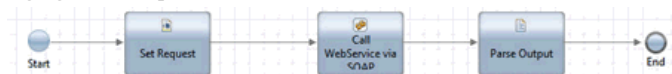


Ahora su entrada de servicio determinará la entrada al conector SOAP.

Utilice xpath para mapear su variable de respuesta en salidas adecuadas.

Este ejemplo utiliza una única variable de salida (\_degreesC).

Agregue un script de servidor al final de su servicio.



Utilice xpath para mapear su respuesta xml en sus variables de salida. Por ejemplo:

Devuelve una lista de nodos de todos los nodos 'Página' hasta el nodo VisioDocument / Pages:

```
xml.xpath ("VisioDocument / Pages / Page");
```

**Nota :** Es posible que deba experimentar con tener o no tener una barra al principio, dependiendo de la estructura del XML.

Devuelve una lista de nodos de todos los nodos 'Maestros' que tienen el atributo NameU igual a 'Soporte horizontal':

```
xml.xpath ("VisioDocument / Masters / Master [@ NameU = 'Soporte horizontal']");
```

En cualquier caso, debe conocer la ruta del nodo Y el espacio de nombres. El siguiente xpath ignora la profundidad e ignora los espacios de nombres.

Es lo mismo que i, excepto que ignora el espacio de nombres y la profundidad del nodo 'Página':

```
xml.xpath = "// * [nombre-local () = 'Página']";
```

En cualquier caso, el resultado es una lista de nodos que se puede utilizar de la siguiente manera:

```
var nodeList = xml.xpath (...);
tw.local.objArray = nuevo tw.object.listOf.myObj ();
for (var i = 0; i < nodeList.length; i ++) {
    var obj = new tw.object.myObj ();
    // Si el nombre del nodo siempre existe como hijo
    obj.name = nodeList [i] .name [0] .getText ();
    tw.local.objArray [tw.local.objArray.listLength] = obj;
}
```