Web Services & EAI
What & Why

WHAT

A web service is a collection of protocols and standards.

- Used for exchanging data between applications or systems.
- Various Programming languages and Software running on various platforms can use web services to exchange data over computer networks.
- Similar to inter-process communication on a single computer.
- Uses Common Network Protocols such as HTTP, SOAP, XML

WHY

- Web Services can be reused
- Web Services can be developed on any computer platform
- In Any development environment.

http://siebelunleashed.com
Important Terms and Definition

Integration Object:
Represent integration metadata for Siebel business objects, XML as common structures that the EAI infrastructure can understand.

- Structural conventions, they can be traversed and transformed programmatically
- Very similar to Siebel Business Object (BO)
- Can contain various integration components (BC)

XML (Extensible Markup Language):
- A flexible way to create common information formats
- Based on SGML (Standard Generalized Markup Language)
- Share both the format and the data on the World Wide Web, intranets etc.

http://siebelunleashed.com
**WSDL (Web Service Description Language)**:

- Standard format for describing a web service
- How To interface with XML-based services.
- Acts as an End point for a Web Service

**SOAP (Simple Object Access Protocol)**:

- SOAP is a standard for exchanging XML-based messages using HTTP.
- SOAP forms the foundation layer of the web services stack.
- Providing a basic messaging framework.

http://siebelunleashed.com
Siebel Web Service

Siebel Web Service can be used to perform two types of communication

- **Inbound Communication**

  Service that is setup to receive the requests from external application or in simple words it can be said that it acts as an *interface to* the Siebel application for the 3rd party applications.

- **Outbound Communication**

  Service that is setup to send data to 3rd party external applications which is acts as an *interface of* external applications for Siebel Application

http://siebelunleashed.com
Inbound Web Service Setup

Ground Work Needed to setup a Inbound Service

- Identify the information that needs to inserted in the Siebel Application
- Identify the which Siebel Business Components and Siebel Business Objects it involves
- Identify the class of Business Component
- Identify the User Properties

Actual Steps Involved in Creating Inbound Web Service

http://siebelunleashed.com
File Menu --> New Object --> EAI Tab --> Choose Integration object from the option (It will start Integration Object Builder Wizard)

http://siebelunleashed.com
Select the Project in which you want to create the Integration object
(* It is better to create a new project) http://siebelunleashed.com

Choose **EAI Siebel Wizard** from the dropdown menu on the business service in the second option. Click on Next.
Select the Business Object on which you want to base this Integration object or in simple words choose the Business Object which contains all the Business Components that include all the information we require.

Enter the Name of the Integration Object you want to create.

http://siebelunleashed.com
Click on Next

http://siebelunleashed.com
Select the business components that you want to include as a part of integration object, unselect the components which you don’t require. Click Next.

http://siebelunleashed.com
You need to make all the fields of all the integration components as inactive which you don’t require. And you can mark the Inactive fields as active which are required.

http://siebelunleashed.com
Next step is to create a Business Service. Which will include the methods you want to give as an Interface to your web service.

Business Service
Enter the Name of the project you created or you used for Integration Object

Enter the class as `CSSEAIDataSyncService`
Enter the Display name in Display name.
Create the methods that you want to include in the business service

http://siebelunleashed.com
## Business Service Method

Enter the name as **QueryById**

Enter Display name as **SBL_QUERY_BY_ID-1004225111-0M3** in Display Name

String Reference.

http://siebelunleashed.com
Business Service Method Arguments

You need to create two service method arguments for this method.

1st argument
Enter **PrimaryRowId** in the Name column.
Enter Data Type as **String**
Enter Type as **Input**
Enter Storage Type as **Property**
Enter Display Name String Override as **Id**

2nd argument
Enter **SiebelMessage** in the Name Column.
Enter Data Type as **Integration Object**
Enter the **Name** of Integration object you created in Integration object field
Enter Type as **output**
Enter Storage Type as **hierarchy**
Enter the Display name String Override as anything you like
Your configuration in Siebel Tools is Complete

You need to create Web Service Record in Application

http://siebelunleashed.com
In the Application go to Administration – WebServices – Inbound Service

Click on New in Inbound WebServices list applet.

Enter the URL of the web server in the Namespace column

Enter Name of the WebService in the Name column

Enter Status as Active

Enter any Comments you want to give
Click on New Button in Service Ports list applet
Enter the Name of the Port
Click on Pick Applet icon in Type Column
Click on New in the Inbound WebService Port type pick applet.
Give the Name in the Name column
Choose Business Service as Implementation Type
Select the Business Service you created in the tools from the Pick List.
Select the Newly created record in the pick applet and click ok.
(You should see your business service Name in the Business Service column)
Select HTTP as the Transport protocol.
Give the Address of the thin client in the address field.

http://<webserver>/eai_<lang>/start.swe?
SWEEExtSource=WebService&SWEEExtCmd=Execute&
Username=<username>&Password=<password>

Select SOAP_RPC_LITERAL in the Binding column.
Click on **New** button in Operations List applet

Click on **New** button in Operations pick Applet

Click on Pick Applet icon in Method Display name column

Select **QueryById** Method you created from the pick applet

Select **None** Instead of Username/Password from **Authentication Type** column

http://siebelunleashed.com
Outbound Webservice

Go to File Menu → New Object → EAI → Web Services

http://siebelunleashed.com
Select the project in which you want to create this service

Browse for the WSDL file that was provided by External Application

Next two fields will get auto populated.

First one will contain the path of Export file that is created as the result of Import

Second one will contain the path of the log file.
If everything is fine then you will get this prompt.

Click on finish button to finish the process
Compile the business Service Created by the Web Service Wizard

http://siebelunleashed.com
Click on Import button
Browse to the export file that was created by the Web Service Wizard
Click Import

This will create a Outbound WebService Record.
Sample Inbound Soap Request

<?xml version="1.0" encoding="UTF-8" ?>
<soapenv:Envelope xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <ns0:QueryById xmlns:ns0="http://st-cms.ps.ge.com/"/>
      <PrimaryRowId>1-A43E4</PrimaryRowId>
    </ns0:QueryById>
  </soapenv:Body>
</soapenv:Envelope>
Sample Inbound Soap Response

<SOAP-ENV:Envelope
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<SOAP-ENV:Body>
   <rpc:QueryByIdResponse xmlns:rpc="http://st-cms.ps.ge.com/">
      <SiebelMessage>
         <ListOfCmsquote
            xmlns="http://www.siebel.com/xml/CMSQuoteUpdate">
            <ListofQuote />
         </ListOfCmsquote>
      </SiebelMessage>
   </rpc:QueryByIdResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

http://siebelunleashed.com
<?xml version="1.0" encoding="UTF-8" ?>
<?Siebel-Property-Set EscapeNames="false"?>
<definitions xmlns="http://schemas.xmlsoap.org/wsdl/
xmlns:xsdLocal0="http://www.siebel.com/xml/CMSQuoteUpdate"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/
targetNamespace="http://3.209.221.101/
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:tns="http://3.209.221.101/">
<types>
<xsd:schema elementFormDefault="qualified"
xmlns:xsdLocal0="http://www.siebel.com/xml/CMSQuoteUpdate"
attributeFormDefault="unqualified"
targetNamespace="http://www.siebel.com/xml/CMSQuoteUpdate"
xmlns:xsd=http://www.w3.org/2001/XMLSchema>
<xsd:element name="ListOfCmsquote" type="xsdLocal0:ListOfCmsquote" />
<xsd:complexType name="ListOfCmsquoteTopElmt">
<xsd:sequence>
<xsd:element name="ListOfCmsquote" maxOccurs="1" minOccurs="1" type="xsdLocal0:ListOfCmsquote" />
</xsd:sequence>
</xsd:complexType>
<xsd:simpleType name="string100">
<xsd:restriction base="xsd:string">
<xsd:maxLength value="100" />
</xsd:restriction>
</xsd:simpleType>
</types>

http://siebelunleashed.com
<message name="RCSQuoteUpdate_QueryById_Input">
  <part name="PrimaryRowId" type="xsd:string" />
</message>

<portType name="RCSQuoteUpdate_Binding_RCSQuoteUpdate">

<operation name="QueryById">
  <input message="tns:RCSQuoteUpdate_QueryById_Input" />
  <output message="tns:RCSQuoteUpdate_QueryById_Output" />
</operation>

</portType>

<binding name="RCSQuoteUpdate_Binding" type="tns:RCSQuoteUpdate_Binding_RCSQuoteUpdate">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="rpc" />
  <operation name="QueryById">
    <soap:operation soapAction="rpc/http://3.209.221.101/:QueryById" />
    <input>
      <soap:body namespace="http://3.209.221.101/" use="literal" />
    </input>
    <output>
      <soap:body namespace="http://3.209.221.101/" use="literal" />
    </output>
  </operation>
</binding>

<service name="RCSInboundService">
  <port binding="tns:RCSQuoteUpdate_Binding_Binding_RCSQuoteUpdate" name="RCSQuoteUpdate">
  </port>
</service>
</definitions>

http://siebelunleashed.com