**HMH**

**Enterprise Web services Project**

**Migration Guide for Java WS Consumers**

**Version No.1.2**



|  |  |  |  |
| --- | --- | --- | --- |
|  | **Prepared By / Last Updated By** | **Reviewed by** | **Approved By** |
| **Name** | Ramesh Kumar Muthukumar | Andrew Trokhymenko |  |
| **Role** |  |  |  |
| **Signature** |  |  |  |
| **Date** | May 18, 2012 |  |  |

Table of Contents

[1.0 Introduction 3](#_Toc321386410)

[1.1 Purpose 3](#_Toc321386411)

[1.2 Intended Audience 3](#_Toc321386412)

[1.3 Definitions and Acronyms 3](#_Toc321386413)

[2.0 Guidelines for Java Web Service Consumers 4](#_Toc321386414)

[2.1.1 List of Apache Web Service Interfaces and wsdl URL 5](#_Toc321386415)

[2.1.2 How to retrieve interfaces using Locator 6](#_Toc321386416)

[2.1.3 EWSBindings Jar Location 8](#_Toc321386417)

[2.1.4 Host 8](#_Toc321386418)

# Introduction

### Purpose

The purpose of this document is to guide the developers on the changes required for Java Web Services Consumers for Apache Web Services.

### Intended Audience

* + *Team members*

### Definitions and Acronyms

| Acronym | ****Term**** |
| --- | --- |
| WSDL | Web Service Definition Language |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# Guidelines for Java Web Service Consumers

The Java Web Service consumers who are consuming the Tibco Services need to change the way of getting the client stub as explained in this section. We shall take the Pinger Test Drivers – Java classes – which are used to test the availability of the Tibco Web Services as example. Though this section explains the changes only for pinger test drivers, the same guidelines should be followed by all the Java Web Service consumers/clients.

By making small changes here and there in the Pinger source code we can make the chron jobs to test the availability of the Apache Web Services also.

For example, we shall take the AccountStatementPdfTest.java class which tests the availability of the AccountStatementPdf service.

Apache Web Service bindings provide a unique service locator using which we can get all the ports for the available services. But in the Tibco implementation we have a separate service locator for each service. So in the source code of the Test driver we need to change the service locator definition as below.

Tibco Implementation:

|  |
| --- |
| *AccountStatementPDFLocator locator = new AccountStatementPDFLocator();* |

After re-defining the service locator in New Apache Implementation,

|  |
| --- |
| *EnterpriseWebServicesServiceagentLocator locator = new EnterpriseWebServicesServiceagentLocator();* |

All the Pinger Test Drivers will be using the above mentioned service locator using which we can get all the endpoints using the WSDL URL.

Once we get the reference to the EnterpriseWebServicesServiceagentLocator we can get the binding for all the services by passing the valid WSDL URL of the Apache Web Services.

For Example, the WSDL URL for the customer interface looks like below.

|  |
| --- |
| *http://<HOST>/services/customer* |

Using the locator reference the binding can be retrieved and assigned to the respective service api as below.

Tibco Implementation:

|  |
| --- |
| *com.hmco.www.EAI.OTS.Customer.accountStatementPDFImpl.SOAPEventSourceBindingStub binding = null;* |

Binding declaration for Apache Services,

|  |
| --- |
| *com.hmco.www.EAI.Tibco.EnterpriseWebServices.CustomerEx binding =* ***null****;* |

Using the below code snippet the binding for the specific service can be retrieved.

|  |
| --- |
| *String url = “http://<HOST>/services/customer”;*  *binding = locator.getCustomerExEndpoint1(new URL(url));* |

*url* represents the WSDL url of the customer interface in this case.

### List of Apache Web Service Interfaces and wsdl URL

|  |  |  |
| --- | --- | --- |
| **Service Interface** | **Service Methods** | **WSDL URL** |
| Customer | accountStatementPDF | *http://<HOST>/services/customer* |
| createEtailCustomer |
| createSundryCustomer |
| createSundryCustomerAsync |
| createSundryCustomerAsyncResult |
| findEtailCustomer |
| getCustomerDetailAsyncPoll |
| getCustomerDetail |
| getCustomerDetailAsync |
| getCustomerDetailAsyncResult |
| getCustomerRelationships |
| getCustomerRelationshipsAsync |
| getCustomerRelationshipsAsyncResult |
| findSFDCCustomer |
| getSFDCRep |
| submitPackagingList |
| External | eskerTransaction | *http://<HOST>/services/esker* |
| Material | getMaterialATPasyncPoll | *http://<HOST>/services/material* |
| getMaterialATP |
| getMaterialATPAsync |
| getMaterialATPAsyncResult |
| getMaterialDetailasyncPoll |
| getMaterialDetail |
| getMaterialDetailEx |
| getMaterialDetailAsync |
| getMaterialDetailAsyncResult |
| getMaterialSubstitutions |
| Order | findSfdcBasisOrder | *http://<HOST>/services/order* |
| findUnsubmittedOrders |
| submitSample |
| submitSfdcOrder |
| OrderEx | getSubmitSaleEx | *http://<HOST>/services/orderEx* |
| submitSaleAsync |
| submitSaleAsyncResult |
| submitSaleasyncPoll |
| submitSale |
| Transaction | findTransactions | *http://<HOST>/services/transaction* |
| findTransactionDetailEx |
| getInvoices |
| getShipments |
| getTransactionDetail |
| getTransactionDetailEx |
| retrieveInvoicePDF |

### How to retrieve interfaces using Locator

|  |  |  |
| --- | --- | --- |
| **Service Interface** | **Service Methods** | **Locator Method** |
| Customer | accountStatementPDF | *binding = locator.getCustomerExEndpoint1(****new*** *URL(url))* |
| createEtailCustomer |
| createSundryCustomer |
| createSundryCustomerAsync |
| createSundryCustomerAsyncResult |
| findEtailCustomer |
| getCustomerDetailAsyncPoll |
| getCustomerDetail |
| getCustomerDetailAsync |
| getCustomerDetailAsyncResult |
| getCustomerRelationships |
| getCustomerRelationshipsAsync |
| getCustomerRelationshipsAsyncResult |
| findSFDCCustomer |
| getSFDCRep |
| submitPackagingList |
| External | eskerTransaction | *binding = locator. getEskerServicesEndpoint1 (new URL(url))* |
| Material | getMaterialATPasyncPoll | *binding = locator. getMaterialExEndpoint1 (****new*** *URL(url))* |
| getMaterialATP |
| getMaterialATPAsync |
| getMaterialATPAsyncResult |
| getMaterialDetailasyncPoll |
| getMaterialDetail |
| getMaterialDetailEx |
| getMaterialDetailAsync |
| getMaterialDetailAsyncResult |
| getMaterialSubstitutions |
| Order | findSfdcBasisOrder | *binding = locator. getOrderEndpoint1 (new URL(url))* |
| findUnsubmittedOrders |
| submitSample |
| submitSfdcOrder |
| OrderEx | getSubmitSaleEx | *binding = locator. getOrderExEndpoint1 (****new*** *URL(url))* |
| submitSaleAsync |
| submitSaleAsyncResult |
| submitSaleasyncPoll |
| submitSale |
| Transaction | findTransactions | *binding = locator. getTransactionExEndpoint1 (new URL(url))* |
| findTransactionDetailEx |
| getInvoices |
| getShipments |
| getTransactionDetail |
| getTransactionDetailEx |
| retrieveInvoicePDF |

An Example to get the client stub for Customer Interface,

|  |
| --- |
| ***public void inititalize(String urlEndpoint) throws ApplicationException {***  ***stub = getCustomerExStub(urlEndpoint);***  ***}***  ***//com.hmco.www.EAI.OTS.Order.submitSaleImpl.SOAPEventSourceBindingStub getSubmitSaleStub(String urlEndpoint) throws ApplicationException {***  ***private CustomerEx getCustomerExStub(String urlEndpoint) throws ApplicationException {***  ***logger.debug("Entered getSubmitSaleStub() method");***  ***//com.hmco.www.EAI.OTS.Customer.accountStatementPDFImpl.SOAPEventSourceBindingStub stub;***  ***com.hmco.www.EAI.Tibco.EnterpriseWebServices.CustomerEx stub;***  ***try {***    ***if (logger.isDebugEnabled())***  ***logger.debug("SubmitSale endpoint: " + urlEndpoint);***    ***//AccountStatementPDFLocator locator = new AccountStatementPDFLocator() EnterpriseWebServicesServiceagentLocator locator = new EnterpriseWebServicesServiceagentLocator();***    ***//locator.setSOAPEventSourceEndpointAddress(urlEndpoint);***  ***//stub = (com.hmco.www.EAI.OTS.Customer.accountStatementPDFImpl.SOAPEventSourceBindingStub) locator.getSOAPEventSource();***  ***stub = locator.getCustomerExEndpoint1(new URL(urlEndpoint));***  ***} catch (Exception e) {***  ***logger.error("Error getting SubmitSale stub", e);***  ***throw new ApplicationException(e);***  ***}***    ***logger.debug("Exited getSubmitSaleStub() method");***    ***return stub;***  ***}*** |

### EWSBindings Jar Location

**Name :** EwsBindings\_Apache.zip

**Location :** <https://eroom.dp.hmco.com/eRoom/Sunrise/SunriseTechnical/0_4b820>

### Host

Apache Staging (SAP Q01): **eaicamel-stg.hmco.com** (10.198.45.200)

Apache Production (SAP P01): **eaicamel-prd.hmco.com** (10.198.51.201)

Apache Dev: (SAP Dev01): NA