

WebService

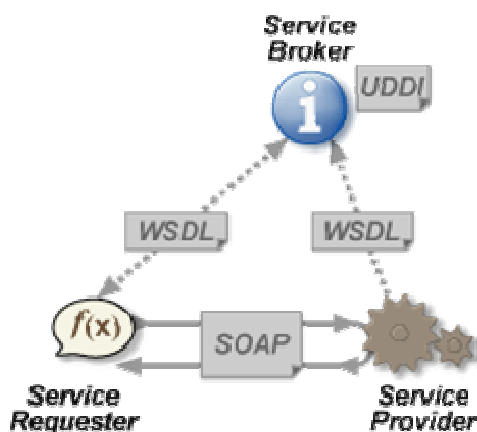
Summary

WebService is a library to request and provide WebService according to Integration service standard of e-government development framework.

Main Concept

Web Services

W3C defines the Web Service as the 'software system for interaction between computers generated in network'. In general, Web Service is the Web API executed in the remote system that provides requested service and approached in network such as Internet.



- See : http://en.wikipedia.org/wiki/Web_Services

Open Source Used

Apache CXF

WebService uses [Apache CXF](#) to implement Web Service.

Description

Since WebService is a library that is implemented according to the [Integration Service](#) standard, type of use such as API will not be explained in this chapter. This chapter describes additional setting information for WebService only and guides the setting method.

Metadata

WebService requires Web Service Client and Server information to request and provide associated service.

Physical ERD

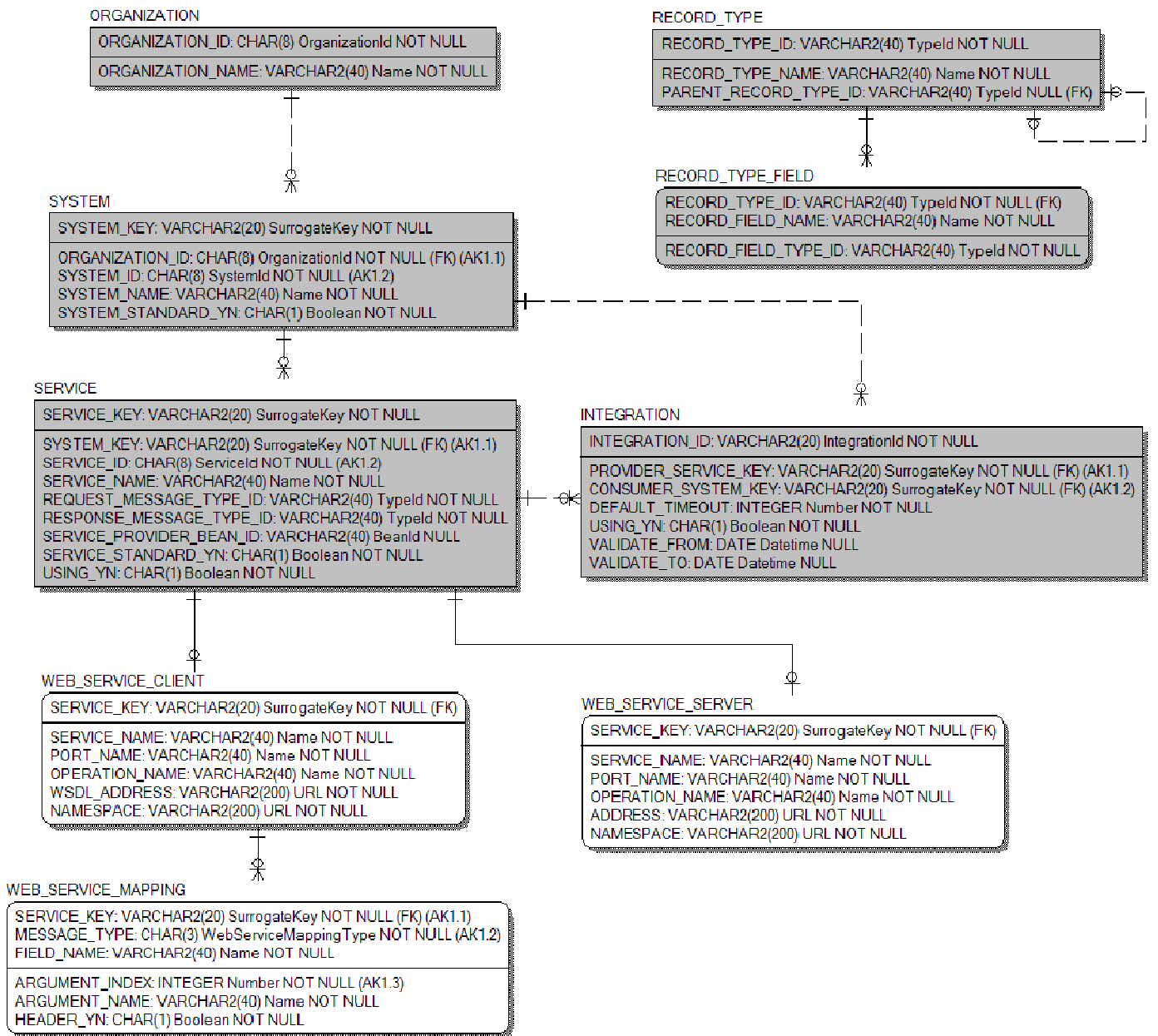


Table	Description
WEB_SERVICE_SERVER	This contains information required to publish the association service in the form of Web Service.
WEB_SERVICE_CLIENT	This contains information required to call the association service published in the form of Web Service.
WEB_SERVICE_MAPPING	This contains mapping information between Web Service messages and standard messages, in order to call the Web Service of existing Legacy system, not the service developed according to integration service standard of e-government.

Description on Physical Model Domain

Domain	Data Type	Description	Remarks
URL	VARCHAR2(200)	Indicate URL.	
WebServiceMappingType	CHAR(2)	Indicate Req/Res division.	'REQ' : Request 'RES' : Response

Description on Physical Model Table

WEB_SERVICE_SERVER

Table Name	WEB_SERVICE_SERVER						
Description	Contain information required to publish the associated service in the form of Web Service.						
Column							
Seq	PK	Column Name	Korean Name	Domain	Data Type	Null	Description
1	Y	SERVICE_KEY	Service KEY	SurrogateKey	VARCHAR2(20)	N	Service Key.
2		ADDRESS	Address	URL	VARCHAR2(200)	N	Address to publish service.
3		NAMESPACE	Name Space	URL	VARCHAR2(200)	N	Name space of service.
4		SERVICE_NAME	Service Name	Name	VARCHAR2(40)	N	Name of service to use when publishing.
5		PORT_NAME	Port Name	Name	VARCHAR2(40)	N	Name of port to use when publishing.
6		OPERATION_NAME	Function Name	Name	VARCHAR2(40)	N	Name of function to use when publishing.
Constraints							
PRIMARY KEY (SERVICE_KEY)							
FOREIGN KEY (SERVICE_KEY) REFERENCES SERVICE (SERVICE_KEY)							

WEB_SERVICE_CLIENT

Table Name	WEB_SERVICE_CLIENT						
Description	Contain information required to call the associated service published in the form of Web Service.						
Column							
Seq	PK	Column Name	Korean Name	Domain	Data Type	Null	Description
1	Y	SERVICE_KEY	Service KEY	SurrogateKey	VARCHAR2(20)	N	Service Key.
2		WSDL_ADDRESS	WSDL address	URL	VARCHAR2(200)	N	WSDL address of service to use.
3		NAMESPACE	Name space	URL	VARCHAR2(200)	N	Name space of service.
4		SERVICE_NAME	Service name	Name	VARCHAR2(40)	N	Name of service to use.
5		PORT_NAME	Port name	Name	VARCHAR2(40)	N	Name of port to use.
6		OPERATION_NAME	Function name	Name	VARCHAR2(40)	N	Name of function to use.
Constraints							
PRIMARY KEY (SERVICE_KEY)							
FOREIGN KEY (SERVICE_KEY) REFERENCES SERVICE (SERVICE_KEY)							

WEB_SERVICE_MAPPING

Table Name	WEB_SERVICE_MAPPING						
Description	This contains the mapping information between Web Service message and standard service to call the Web Service of existing Legacy system rather than the service developed according to Integration service standard of e-government.						
Column							
Seq	PK	Column Name	Korean Name	Domain	Data Type	Null	Description

1	Y	SERVICE_KEY	Service KEY	SurrogateKey	VARCHAR2(20)	N	Service key.
2	Y	MESSAGE_TYPE	Message type	WebServiceMappingType	CHAR(3)	N	Req/Res type.
3	Y	FIELD_NAME	Field Name	Name	VARCHAR2(40)	N	Name of standard message name field.
4		ARGUMENT_INDEX	Variable sequence	Number	Integer	N	Variable order of Web Service message.
5		ARGUMENT_NAME	Variable name	Name	VARCHAR2(40)	N	Variable of Web Service message.
6		HEADER_YN	Header or not	Boolean	CHAR(1)	N	Web Service header or not.
Constraints							
PRIMARY KEY (SERVICE_KEY, MESSAGE_TYPE, FIELD_NAME)							
FOREIGN KEY (SERVICE_KEY) REFERENCES WEB_SERVICE_CLIENT (SERVICE_KEY)							

How to Set

The following settings are required to use Webservice.

1. [Add dependency setting to pom.xml](#)
2. [Spring XML Configuration setting](#)

Add dependency to pom.xml

To use Webservice, add the following dependency to dependencies tag of pom.xml.

- Indicate the version of egovframework to use in `${egovframework.version}`, the value of `<version>` tag.

```

...
<dependencies>
    ...
    <dependency>
    <groupId>egovframework.rte</groupId>
    <artifactId>egovframework.rte.itl.webservice</artifactId>
    <version>${egovframework.version}</version>
    </dependency>
    ...
</dependencies>
...

```

Spring XML Configuration

Import the "context-webservice.xml" file containing the basic setting for Webservice to Spring XML Configuration file.

```
<import resource="classpath:/egovframework/rte/itl/webservice/context/context-webservice.xml"/>
```

Register Context and DataSource. (It is not required to set DataSource if there is one used by the project. Provided, however, that id must be "dataSource".)

```

<!--EgovWebServiceContext .
organizationIdandsystemIdshould enter agency ID and system ID of the current system. -->

```

```

<bean id="egovWebServiceContext"
class="egovframework.rte.itl.webservice.EgovWebServiceContext"
init-method="init">
<property name="organizationId" value="ORG_EGOV"/>
<property name="systemId" value="SYS00001"/>
<property name="defaultTimeout" value="5000"/>
<property name="integrationDefinitionDao" ref="integrationDefinitionDao"/>
<property name="webServiceServerDefinitionDao" ref="webServiceServerDefinitionDao"/>
<property name="webServiceClientDefinitionDao" ref="webServiceClientDefinitionDao"/>
<property name="typeLoader" ref="typeLoader"/>
<property name="classLoader" ref="classLoader"/>
</bean>

<!--DataSourcesetting. Recreate according to the system. Following is HSQL Sample. -->
<bean id="dataSource" class="org.apache.commons.dbcp.BasicDataSource" destroy-method="close">
<property name="driverClassName" value="net.sf.log4jdbc.DriverSpy"/>
<property name="url" value="jdbc:log4jdbc:hsqldb:hsqldb://localhost/test"/>
<property name="username" value="sa"/>
<property name="password" value=""/>
<property name="defaultAutoCommit" value="false"/>
<property name="poolPreparedStatements" value="true"/>
</bean>

```

Development of the Client Module

WebService Client module is called according to the integration service standard published to the Web Service. In this chapter, setting type will be explained (See [Associated Service API](#) for setting type.) Following process is required to set Client module.

1. [Add Metadata WEB_SERVICE_CLIENT setting](#)
2. [\(Optional\)Add Metadata WEB_SERVICE_MAPPING setting](#)

Add Metadata WEB_SERVICE_CLIENT Setting

To set a client module, it is required to add the setting to WEB_SERVICE_CLIENT Table of Metadata. Assume that association registration information is set in the INTEGRATION Table, the Metadata of Integration service as follows.(* information such as agency, system, service, message type are set. Assume that developed system is 'SYSTEM_CONSUMER')

INTEGRATION						
ID	PROVIDER_SERVICE_KEY	CONSUMER_SYSTEM_KEY	DEFAULT_TIMEOUT	USING_YN	VALIDATE_FROM	VALIDATE_TO
'INT_VERIFY_NAME'	'SERVICE_VERIFY_NAME'	'SYSTEM_CONSUMER'	5000	'Y'	NULL	NULL

To call Web Service 'SERVICE_VERIFY_NAME', add the setting that has 'SERVICE_VERIFY_NAME' as SERVICE_KEY in WEB_SERVICE_CLIENT.

WEB_SERVICE_CLIENT					
SERVICE_KEY	WSDL_ADDRESS	NAMESPACE	SERVICE_NAME	PORT_NAME	OPERATION_NAME
'SERVICE_VERIFY_NAME'	'http://192.168.0.1:8080/Sample/services/VerifyName?wsdl'	'http://itl/sample/'	'VerifyNameService'	'VerifyNamePort'	'service'

(Optional) Add Metadata WEB_SERVICE_MAPPING Setting

If the Web Service to call is not the service developed according to integration service standard of e-government, message header may be different. So that separate mapping information is required. In e-government Integration service standard, Attributes that comes in Web Service Header part are

defined in EgovIntegrationMessageHeader and the body part is defined in body of EgovIntegrationMessage, header and body part can be separated without separate mapping information. In case of Web Service that does not follow standard, some values defined in the body part of EgovIntegrationMessage should be included in the header.

The information of WEB_SERVICE_MAPPING Table is based on the message form defined in Integration service standard. Request Message of service 'SERVICE_VERIFY_NAME' has 'name', 'residentRegistrationNumber' fields and Response Message has 'result' field. Accordingly, WEB_SERVICE_MAPPING corresponding to 'SERVICE_VERIFY_NAME' should have the following information.

WEB_SERVICE_MAPPING					
SERVICE_KEY	MESSAGE_TYPE	FIELD_NAME	ARGUMENT_INDEX	ARGUMENT_NAME	HEADER_YN
'SERVICE_VERIFY_NAME'	'REQ'	'name'	1	'name'	Y
'SERVICE_VERIFY_NAME'	'REQ'	'residentRegistrationNumber'	2	'residentRegistrationNumber'	N
'SERVICE_VERIFY_NAME'	'RES'	'result'	1	'result'	N

Determine whether relevant field will be contained in the header of Web Service Envelop depending on the value of HEADER_YN column among above information. By applying the above setting value, 'name' field among request message is included in the header of Web Service Envelop.

Development of the Server Module

The followings are the process of developing Web Service Server module.

1. [Add EgovWebServiceServlet to web.xml](#)
2. [Add Metadata WEB_SERVICE_SERVER setting](#)

AddingEgovWebServiceServlet to web.xml

Add EgovWebServiceServlet setting to web.xml.

```

...
<servlet>
<description></description>
<display-name>EgovWebServiceServlet</display-name>
<servlet-name>EgovWebServiceServlet</servlet-name>
<servlet-class>egovframework.rte.itl.webservice.EgovWebServiceServlet</servlet-class>
<load-on-startup>1</load-on-startup>
</servlet>
<servlet-mapping>
<servlet-name>EgovWebServiceServlet</servlet-name>
<url-pattern>/services/*</url-pattern>
</servlet-mapping>
...

```

The value of <url-pattern> tag can be changed. See WEB_SERVICE_SERVER setting for more details.

Add Metadata WEB_SERVICE_SERVER setting

Assume that association registration information is set in Integration table, the metadata of Integration service as follows. (* information such as institution, system, service and message type is set, and assume that the service to publish is 'SERVICE_VERIFY_NAME')

INTEGRATION

ID	PROVIDER_SERVICE_KEY	CONSUMER_SYSTEM_KEY	DEFAULT_TIMEOUT	USING_YN	VALIDATE_FROM	VALIDATE_TO
'INT_VERIFY_NAME'	'SERVICE_VERIFY_NAME'	'SYSTEM_CONSUMER'	5000	'Y'	NULL	NULL

To publish Web Service 'SERVICE_VERIFY_NAME', add configuration that has 'SERVICE_VERIFY_NAME' as SERVICE_KEY in WEB_SERVICE_SERVER.

WEB_SERVICE_SERVICE					
SERVICE_KEY	ADDRESS	NAMESPACE	SERVICE_NAME	PORT_NAME	OPERATION_NAME
'SERVICE_VERIFY_NAME'	"/VerifyName"	'http://itl/sample/'	'VerifyNameService'	'VerifyNamePort'	'service'

As <url-pattern> tag value of <servlet-mapping> tag is the address for providing service, ADDRESS Column value of WEB_SERVICE_SERVER Table indicates the relative location for <url-pattern> tag value. For example, if IP of Web Application is 192.168.0.1, Port is 8080, Context Root is "Sample", url-pattern is "/services/*", WSDL Address of above 'SERVICE_VERIFY_NAME' is http://192.168.0.1:8080/Sample/services/VerifyName?wsdl.

Publish to WAS

Explain the method of deploying the application including e-government WebService to WAS. In case of Apache CXF, use the Web Service related library as provided by CXF. If WAS provides Web Service library by default, it may not operate properly. Accordingly, it is required to change the setting so as to use CXF library. Most resolutions are to change the sequence of Class Loading to load the library of WEB-INF of Web Application first.

This WebService uses JAX-WS 2.0 or over.

TmaxSoft JEUS 6.0

In case of the e-government WebService, CXF is used internally, but if it is published to JEUS 6.0, a problem occurs when publishing the server module. It is because Web Services related library that is included at JEUS 6.0 by default is not the same as the library used by e-government WebService. At present, following 2 problems were found.

- Publish Address Problem
Use the relative path for the path of EgovWebServiceServlet when publishing the Server module. In case of library that is used by Apache CXF, it converts to the actual address, however, the library included in JEUS 6.0 by default doesn't convert, resulting in IllegalArgumentException.
- Service Endpoint Interface Reference Problem
E-government WebService dynamically creates the Service Endpoint Interface of server module and implemented class according to Integration service standards. However, in case of library included in JEUS 6.0 by default, it fails to recognize the class created dynamically like this and Exception occurs.

The solution is to enable the library which is used by WebService of e-government to be loaded by ClassLoader first. JEUS 6.0 can set to load the library in WEB-INF/lib first through jeus-web-dd.xml setting.

```
<?xml version="1.0" encoding="UTF-8"?>
<jeus-web-dd xmlns="http://www.tmaxsoft.com/xml/ns/jeus" version="6.0">
<webinf-first>true</webinf-first>
</jeus-web-dd>
```

Locate the above jeus-web-dd.xml file in the WEB-INF folder where web.xml file exists. And if webinf-first is set to true, collision occurs for XML Parser. To avoid collision, remove following 2 files from WEB-INF/lib.

- 'xml-apis-1.0.b2.jar' (or upper version)
- 'stax-api-1.0.1.jar' (or upper version)

JBoss

In case of Jboss, add jboss-web.xml file.

```
<?xml version="1.0" encoding="UTF-8"?>
<jboss-web>
<class-loading java2ClassLoadingCompliance="false">
<loader-repository>
apache.cxf:archive=<WAR file name>
<loader-repository-config>
            java2ParentDelegation=false
</loader-repository-config>
</loader-repository>
</class-loading>
</jboss-web>
```

* <WAR file name> indicates the deploying war file name including the extension.

WebLogic

Since WebLogic 9.2 version supports up to J2EE 1.4 only, JAX-WS 2.0 is not supported. To use WebService at WebLogic, 10.x or over supporting JAX-WS 2.0 should be used.

Reference

- [Integration Service](#)
- <http://cxf.apache.org>