ExtraView™
Your Process. Your Workflow.

Web Services within ExtraView

Utilizing a service-orientated architecture interface within ExtraView to integrate with other applications

www.extraview.com

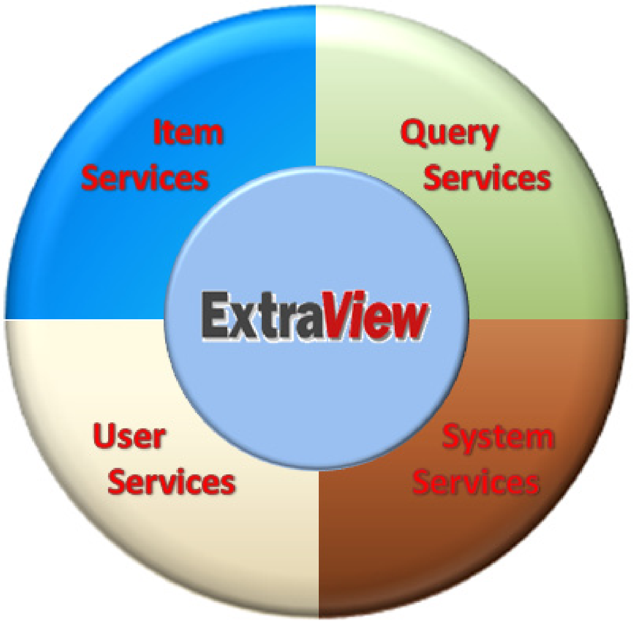
09-2008

Web Services within ExtraView

Utilizing a service-orientated architecture interface within ExtraView to integrate with other applications

Introduction

ExtraView provides a standardized set of methods to integrate with other applications using a service-orientated architecture (SOA). This complements the web-orientated architecture interface (WOA) that is also implemented within ExtraView. The interface is cross platform and can be accessed from any development platform, including Java and .Net.

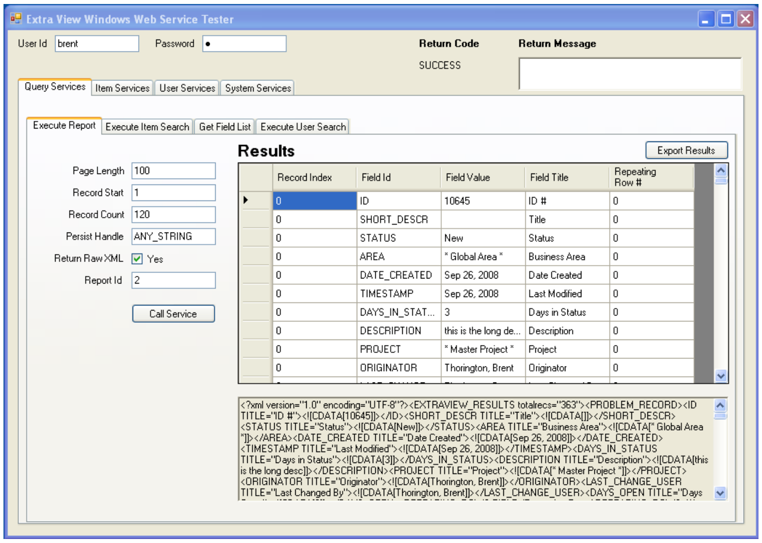


A service-oriented architecture is defined as a group of services, which communicate with each other. The process of communication involves either simple data passing or it involves two or more services coordinating some activity. Some means of connecting services to each other is needed.

SOAs applications are built out of software services. Services are intrinsically unassociated units of functionality, which have no calls to each other embedded in them. Within ExtraView they map to atomic functions to perform specific actions. Broadly, SOAs implement functionalities most humans would recognize as a service, such as filling out an online application for an account, viewing an online bank statement, inserting an item with a database or running a report. Instead of services embedding calls to each other in their source code, protocols are defined which describe how one or more services can talk to each other. This architecture then relies on your business process expert to link and sequence services to meet your business system requirement.

ExtraView's implementation of web services utilizes a service-orientated architecture to provide a full set of integration points between ExtraView and the consumer of its web services.

Implementation Details



User Id: brent
Password: •
Return Code: SUCCESS
Return Message:

Page Length: 100
Record Start: 1
Record Count: 120
Persist Handle: ANY_STRING
Return Raw XML: ☑ Yes
Report Id: 2

Results

Record Index	Field Id	Field Value	Field Title	Repeating Row #
0	ID	10645	ID #	0
0	SHORT_DESCR		Title	0
0	STATUS	New	Status	0
0	AREA	* Global Area *	Business Area	0
0	DATE_CREATED	Sep 26, 2008	Date Created	0
0	TIMESTAMP	Sep 26, 2008	Last Modified	0
0	DAYS_IN_STAT...	3	Days in Status	0
0	DESCRIPTION	this is the long de...	Description	0
0	PROJECT	* Master Project *	Project	0
0	ORIGINATOR	Thorington, Brent	Originator	0

```
<?xml version="1.0" encoding="UTF-8"?><EXTRAVIEW_RESULTS totalrecs="363"><PROBLEM_RECORD><ID
TITLE="ID #"><![CDATA[10645]]></ID><SHORT_DESCR TITLE="Title"><![CDATA[]]></SHORT_DESCR>
<STATUS TITLE="Status"><![CDATA[New]]></STATUS><AREA TITLE="Business Area"><![CDATA[* Global Area
*]]></AREA><DATE_CREATED TITLE="Date Created"><![CDATA[Sep 26, 2008]]></DATE_CREATED>
<TIMESTAMP TITLE="Last Modified"><![CDATA[Sep 26, 2008]]></TIMESTAMP><DAYS_IN_STATUS
TITLE="Days in Status"><![CDATA[3]]></DAYS_IN_STATUS><DESCRIPTION TITLE="Description"><![CDATA[this
is the long desc]]></DESCRIPTION><PROJECT TITLE="Project"><![CDATA[* Master Project *]]></PROJECT>
<ORIGINATOR TITLE="Originator"><![CDATA[Thorington, Brent]]></ORIGINATOR><LAST_CHANGE_USER
TITLE="Last Changed By"><![CDATA[Thorington, Brent]]></LAST_CHANGE_USER><DAYS_OPEN TITLE="Days
```

Sample interface utilizing ExtraView Web Services

For maximum flexibility, the Web Service Software Development Kit (SDK) contains a fully functional set of services written for both Java and the .Net platforms. This allows development on the major platforms utilized within most major corporations. The SDK contains a full set of working examples.

Summary of Services

A wide range of services is exposed with a complete Web Services Description Language (WSDL) to standardize communications with other SOA applications.

Item Services

The principal purpose of the item services is to provide the means to *insert*, *update* and *delete* items within the ExtraView database.

Query Services

The query services allow the creation of services that perform querying on the ExtraView database. This includes running reports and setting up queries where you provide the search filters at runtime.

User Services

Web services allow the development of an interface to manage users within ExtraView. For example, you can *add* a new user, you can manage a user's role(s), you may change a user's password or you may deactivate a user.

System Services

This group of services views and manipulates the metadata within ExtraView. For example, you may look at the metadata of a field in the data dictionary, you may upload field values into a list or you may import item records into ExtraView.

Security

In addition to the high level of security afforded within the HTTPS protocol, the implementation offers an additional encryption level at the message level. This optional feature encrypts transmission of all messages at both the client and the server.

SDK Sample

As mentioned above, the implementation accommodates both Java and .Net interfaces. To expand upon this the SDK contains examples which implement the range of Web Services within both Java and .Net, using C#. The .Net implementation example provides a Windows GUI to all the services.

The source code is provided giving you full working examples of all the individual web services.

Java Example Call

This example was created with Java 1.5. Java 1.5 should be used within both the client and server side of implementations with ExtraView's web services.

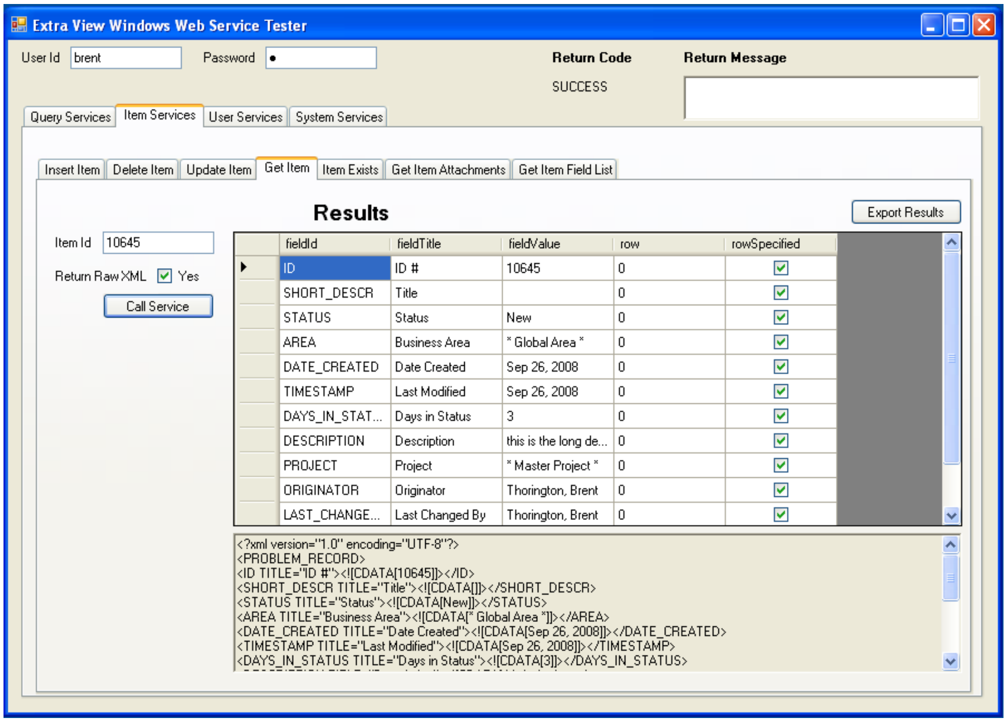
```
public class EVItemServiceClient {
   public static void main(String args[]) {
      ...
      EVUserServiceStub userStub = new ↵
         EVUserServiceStub("http://extraview.net/WS-TEST/services/EVUserService");
      GetItemDocument reqEnvelope = GetItemDocument.Factory.newInstance();
      GetItemRequest request = reqEnvelope.addNewGetItem().addNewRequest();
      request.setUserId("ADMIN");
      request.setPassword("PASSWORD");
      request.setItemId(1443);
      request.setReturnRawXML(true);
      GetItemResponseDocument resEnvelope = stub.getItem(reqEnvelope);
      GetItemResponse response = resEnvelope.getGetItemResponse().getReturn();
      ...
    }
}
```

.Net Example Call

This example was composed with C# using Visual Studio 2008 on the client side.

```
using Client.ItemServiceReference;
namespace Client
{
   ...
   EVItemServicePortTypeClient itemClient = new
      EVItemServicePortTypeClient("EVItemServiceHttpSoap11Endpoint");
   GetItemResponse response = itemClient.getItem(new GetItemRequest()
   {
      userId = "ADMIN",
      password = "PASSWORD",
      itemId = 1443,
      itemIdSpecified = true,
      returnRawXML = true,
      returnRawXMLSpecified=true
   });
   ...
}
```

Example Windows GUI



User Id: brent
Password: •

Return Code: SUCCESS

Return Message:

Results

Item Id: 10645
Return Raw XML: ☑ Yes

	fieldId	fieldTitle	fieldValue	row	rowSpecified
▶	ID	ID #	10645	0	☑
	SHORT_DESCR	Title		0	☑
	STATUS	Status	New	0	☑
	AREA	Business Area	* Global Area *	0	☑
	DATE_CREATED	Date Created	Sep 26, 2008	0	☑
	TIMESTAMP	Last Modified	Sep 26, 2008	0	☑
	DAYS_IN_STAT...	Days in Status	3	0	☑
	DESCRIPTION	Description	this is the long de...	0	☑
	PROJECT	Project	* Master Project *	0	☑
	ORIGINATOR	Originator	Thorington, Brent	0	☑
	LAST_CHANGE...	Last Changed By	Thorington, Brent	0	☑

```
<?xml version="1.0" encoding="UTF-8"?>
<PROBLEM_RECORD>
<ID TITLE="ID #"><![CDATA[10645]]></ID>
<SHORT_DESCR TITLE="Title"><![CDATA[]]></SHORT_DESCR>
<STATUS TITLE="Status"><![CDATA[New]]></STATUS>
<AREA TITLE="Business Area"><![CDATA[* Global Area *]]></AREA>
<DATE_CREATED TITLE="Date Created"><![CDATA[Sep 26, 2008]]></DATE_CREATED>
<TIMESTAMP TITLE="Last Modified"><![CDATA[Sep 26, 2008]]></TIMESTAMP>
<DAYS_IN_STATUS TITLE="Days in Status"><![CDATA[3]]></DAYS_IN_STATUS>
```

Example service to retrieve an item from the database displaying the results

SOA Interface versus a WOA Interface

ExtraView offers multiple API's. The SOA interface is complemented by a WOA interface that utilizes direct HTTP calls to access the API. The range of functionality with each interface is identical, but each interface is used quite differently.

- The SOA interface is standards-based, making it easier to integrate with other SOA-ready applications
- The WOA interface is easier to develop with, but not standards-based
- Within the SOA implementation, calls to ExtraView are converted to HTTP requests. Therefore there is an overhead associated with this task. This is typical of all SOA implementations and the overhead is small, but distinct.

Summary

ExtraView offers a full SOA implementation for integrations with other enterprise systems. The implementation allows for cross-platform development with a rich set of web services that expose client WSDL's.

For more information, please contact ExtraView Corporation at (831) 461-7100, or by email to info@extraview.com.