# SOA Gateway Resource Access

- Overview

Overview

This section describes the operations exposed by SOA Gateway to access data sources. These operations are described in a WSDL which is defined for each web service. A client program connecting to a web service can read the WSDL to determine what operations are available on the server. These operations can be invoked using SOAP or RESTful query.

SOA Gateway also supports MTOM for binary data, raw XML, and HTML. WSDLs can be registered in a UDDI server for service look-up.
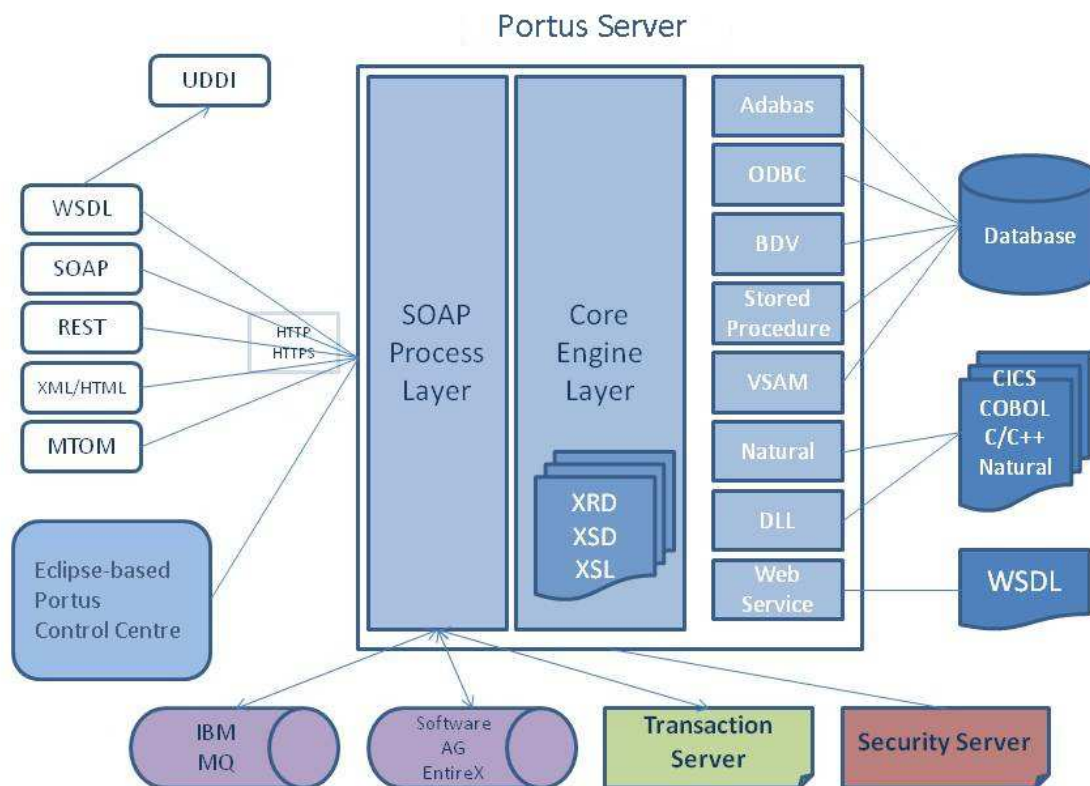
Supported protocol Versions

WSDL 1.1

SOAP 1.1

MTOM 1.0

HTTP 1.1

ODBC Version 3

SOA Gateway can also connect to the Messaging systems such as IBM MQ and Software AG EntireX via their C-interfaces.

Web Service Security can be handled by HTTP Basic Profile authorization (via Apache), or connectivity with an external security manager, such as RACF.

Further Reading

Prerequisites

Retrieve WSDL

Web Service Operations

Providing Key Information

Prerequisites

At this point your SOA Gateway Server should be installed, configured and started.

If you have not yet configured any web services, please refer to the SOA Gateway Control Center section and add at least one web service

Retrieving the WSDL for your resource.

All web services defined will have a WSDL associated with them. This is the starting point for using the operations provided by SOA Gateway. The WSDL describes the operations that may be carried out, and how they are used. This includes a description of valid parameters, data and responses for each of the operations.

In order to get the WSDL for a particular resource you simply issue a standard http request, specifying:

- the server name or *IP-address* of the server where SOA Gateway is installed [and running]

- the *TCP Port number* that SOA Gateway is listening on (as provided in the installation)

- the *name* of the service. (this is the value of the "*Name*" field in the web service properties)

The following example shows the URL required to retrieve the WSDL for an Adabas "Employees" file.

Server Name	PortusGateway
Port Number	56000
Name	adabas_Employees
URL for WSDL	http://PortusGateway:56000/adabas_Employees?WSDL

SOA Gateway Web Services Operations for Data Resources

The operations provided by SOA Gateway for accessing data resources [files, databases, programs, etc.] are now explained.

Parameters are required unless otherwise stated.

Operation	*list* (SOAP)
	LIST (REST)
Description	The *list* operation returns a list of records or rows from your data source.
	The data returned can be limited or restricted by providing *key* information.
	This operation will only be available when the web service is a "database" type, for example, Adabas or MySQL
Parameters	● *Key data*
	Key data must be entered for at least one of the fields defined as a key.
	Refer to the section Specifying Key Data for more information.
	● *Options*
	None
Result	SOAP
	The result will be either :
	● an XML document wrapped in a SOAP message and containing the requested data
	● a SOAP fault message
	REST
	The result will be either :
	● an XML document containing the requested data
	● a fault message

Operation	*select* (SOAP)
	SELECT (REST)

Description	The *select* operation returns a list of records or rows from your data source. The maximum number of rows/records returned can be set via the SOA Gateway Control Center.
	This operation will only be available when the web service is a "database" type, for example, Adabas or MySQL
	The data returned can be limited or restricted by providing *key* information. The select operation extends the capability of the list operation by enabling searches on a larger set of criteria.
	The key information for a select is wrapped in a condition block and can be repeated several times within that block. Each key entry represents an 'AND' condition. Condition blocks can also be repeated several times. Each condition block represents an 'OR' condition. The condition block accepts the following specifiers:
Conditions	• *Less than a specific value (LT).*
	• *Less than or equal to a specific value (LE).*
	• *Equal to a specific value (EQ).*
	• *Greater than a specific value (GT).*
	• *Greater than or equal to a specific value (GE).*
	• *Not equal to a specific value (NE).*
	• *Starting with a specific value (START). Character based fields only.*
	• *Ending with a specific value (ENDS). Character based fields only.*
	• *Containing a specific value (CONTAINS). Character based fields only.*

Example	SOAP
| | ```
    <soapenv:Body>
       <nos:adabasEmployeeSelectElement>
          <!--1 or more repetitions:-->
          <condition>
             <!--Zero or more repetitions:-->
             <personnel_id Condition="GT">50012100</personnel_id>
             <personnel_id Condition="LE">50012700</personnel_id>
          </condition>
          <condition>
             <!--Zero or more repetitions:-->
             <personnel_id Condition="EQ">50012900</personnel_id>
          </condition>
       </nos:adabasEmployeeSelectElement>
    </soapenv:Body>
``` |
| | REST |
| | ```
http://localhost:56005/adabas_Employees_9?
SELECT
&condition[1].personnel_id>50012100
&condition[1].personnel_id<=50012700
&condition[2].personnel_id=50012900
``` |
	The example above specifies 2 condition blocks. This will return data where the (personnel_id > 50012100 and personnel_id <= 50012700) or personnel_id = 50012900
Parameters	● *Key data* Key data must be entered for at least one of the fields defined as a key. Refer to the section Specifying Key Data for more information. ● *Options* None
Result	SOAP The result will be either : ● an XML document wrapped in a SOAP message and containing the requested data ● a SOAP fault message REST The result will be either : ● an XML document containing the requested data ● a fault message

Operation	selectNext
Description	The *selectNext* operation returns a list of records or rows from your data source. A selectNext operation may only be called following a select and subsequently other selectNext calls. For this functionality the initial select operation has to initiate a new Conversation. See Conversational Processing. The resultant conversation id must be passed in any associated selectNext calls. selectNext calls may be issued until end of file is reached or may be terminated by a selectEnd call. The maximum number of rows/records will be that set for set via the SOA Gateway Control Center. This operation will only be available when the web service is a "database" type, for example, Adabas or MySQL There is no key information for a selectNext operation as this will have been passed in by the initiating select operation.

Parameters	• *Key data*
	
	  None
	
	• *Options*
	
	  None

Result	SOAP The result will be either : • an XML document wrapped in a SOAP message and containing the requested data • a SOAP fault message REST Not available

Operation	selectEnd
Description	The *selectEnd* operation terminates a sequence of select and/or selectNext calls with a conversation. A selectEnd operation may only be called following select or selectNext operations. For this functionality the select operation has to initiate a new Conversation. The resultant conversation id must be passed in the selectEnd call. This operation will only be available when the web service is a "database" type, for example, Adabas or MySQL There is no key information for a selectEnd operation as this will have been passed by the initiating select operation.

Parameters	• *Key data*
	
	   None
	
	• *Options*
	
	   None

Result	SOAP The result will be either : • an XML document wrapped in a SOAP message and containing the requested data • a SOAP fault message REST Not available

Operation	selectCount
Description	The *selectCount* operation returns a count of the records or rows that match the criteria set in the condition block(s). This operation will only be available when the web service is a "database" type, for example, Adabas or MySQL The selectCount operation is identical to that of the select operation in terms of its search capabilities. The key information for a selectCount is wrapped in a condition block and can be repeated several times within that block. Each key entry represents an 'AND' condition. Condition blocks can also be repeated several times. Each condition block represents an 'OR' condition. The condition block accepts the following specifiers:

Conditions	• *Less than a specific value (LT).*
	• *Less than or equal to a specific value (LE).*
	• *Equal to a specific value (EQ).*
	• *Greater than a specific value (GT).*
	• *Greater than or equal to a specific value (GE).*
	• *Not equal to a specific value (NE).*
	• *Starting with a specific value (START). Character based fields only.*
	• *Ending with a specific value (ENDS). Character based fields only.*
	• *Containing a specific value (CONTAINS). Character based fields only.*
Parameters	• *Key data*
	Key data must be entered for at least one of the fields defined as a key.
	Refer to the section Specifying Key Data for more information.
	• *Options*
	None
Result	SOAP
	The result will be either :
	• an XML document wrapped in a SOAP message and containing the requested data
	• a SOAP fault message
	REST
	The result will be either :
	• an XML document containing the requested data
	• a fault message

Operation	*get* (SOAP)
	GET (REST)
Description	The *get* operation returns a single record or row from your data source.
	The data returned is specified by providing *unique key* information identifying a single record / row.
	This operation will only be available when the web service is a "database" type, for example, Adabas or MySQL
Parameters	• *Key Data*
	Key data must be entered for at least one of the fields defined as a primary key field, so that a single record can be identified.
	Refer to the section Specifying Key Data for more information.
	• *Options*
	None
Result	SOAP
	The result will be either :
	• an XML document wrapped in a SOAP message and containing the requested data
	• a SOAP fault message
	REST
	The result will be either :
	• an XML document containing the requested data
	• a fault message

Operation	*add* (SOAP)
	ADD (REST)
Description	The *add* operation adds a single record or row of data to your data source.
	This operation will only be available when the web service is a "database" type, for example, Adabas or MySQL
Parameters	● *Add Data*
	Provide values for each of the fields defined. These are the fields / columns of the data on your data source.
	You must add data for at least one of the fields specified as being *primary key* fields.
	Refer to the section Specifying Key Data for more information.
	● *Options*
	None
Result	SOAP
	The result will be either :
	● an XML document wrapped in a SOAP message and containing the requested data
	● a SOAP fault message
	REST
	The result will be either :
	● an XML document containing the requested data
	● a fault message

Operation	*update* (SOAP) *UPDATE (REST)*
Description	The *update* operation updates a single record or row of data. This operation will only be available when the web service is a "database" type, for example, Adabas or MySQL

Parameters	• *Update Data*
	
	     Provide values (NULL or otherwise) for each of the fields defined. These are the fields / columns of the data on your data source.
	
	     Key data must be entered for at least one of the fields defined as a *primary key* field.
	
	     Refer to the section Specifying Key Data for more information.
	
	     Fields that have been left empty or NULL, e.g. `<data></data>` will be set to this value accordingly.
	
	 • *Options*
	
	     None

Result	SOAP The result will be either : • an XML document wrapped in a SOAP message and containing the requested data • a SOAP fault message REST The result will be either : • an XML document containing the requested data • a fault message

Operation	*delete (SOAP)*
	DELETE (REST)
Description	The *delete* operation deletes a single record or row of data from the data source.
	This operation will only be available when the web service is a "database" type, for example, Adabas or MySQL
Parameters	● *Key Data*
	Key data must be entered for at least one of the fields defined as a *primary key* field, so that a single record can be identified.
	Refer to the section Specifying Key Data for more information.
	● *Options*
	None
Result	SOAP
	The result will be either :
	● an XML document wrapped in a SOAP message and containing the requested data
	● a SOAP fault message
	REST
	The result will be either :
	● an XML document containing the requested data
	● a fault message

Operation	*invoke (SOAP)* *INVOKE (REST)*
Description	The *invoke* operation makes a call to a function or program. This operation will only be available when the web service is a "program" type, for example, NATURAL, COBOL or when the web service describes a database stored procedure.
Parameters	● *Parameters* Each of the fields defined can be input, output, or input+output. Provide values for each of the fields defined as input or input+output. These match the function/program parameters that are required on the call.
Result	SOAP The result will be either : ● an XML document wrapped in a SOAP message and containing the requested data ● a SOAP fault message REST The result will be either : ● an XML document containing the requested data ● a fault message

Specifying Key Data

Keys or *Key Fields* in SOA Gateway terms are simply fields or columns on your data source (file, table, etc) that may be used to narrow a search for a record or row of data.

Note:
key fields may or may not be *indexed* at your data source. Indexing provides better response times from your data source. If you are unsure what, if any, fields are key fields (in Adabas terms: Descriptors) or allowed to be used for searching, contact your data source administrator (DBA, etc.).

Key fields are specified to SOA Gateway by setting the appropriate value for the attribute "*key type*" in the Resource Description.

Key Types

There are two types of *key fields* for data sources:

Primary Key Fields

Primary Key Fields are fields in a data source that must always contain unique values.

They are required on all data source operations except the *list* operation, where they are optional. Secondary keys may be used instead or in addition on a list request.

Their values will not be altered in the *update* operation.

Primary Keys cannot contain *wild card* symbols, except on the *list* operation.

Secondary Key Fields

Secondary Key Fields are fields that may be used in narrowing a search.

They are not required in any operation, and their values can be updated.

They may contain wild card symbols.

Using Wild card Symbols and other Generic Search criteria

Wild cards are used where you do not wish to specify an exact value in a key field, but use a generic specification that will match for a range of different values.

These search modifiers may only be used on the *list* operation.

SOA Gateway currently supports the following generic search criteria :

Wild card for one or more characters

The character "*" may be used as a wild card for all characters.

It may only be used on fields defined as "*string*" fields.

Where it appears, it will match any character or group of characters.

Note:
for Adabas resources, this wildcard may only appear at the *end* of the string data supplied.

Example

"Ga*" would match "Gat", "Gate", "Gateway", etc.