

Rollbase SOAP API

The Rollbase SOAP API is designed to enable integration between the Rollbase platform and external systems via standards-based SOAP calls. In order to use this API the client must have a valid Rollbase user account with login credentials: username and password. These credentials must be supplied as parameters to the `login` method, the first SOAP call to Rollbase, which is required to establish an API session. This call will return a session id, which must be used as the first parameter in all subsequent API calls. To end an API session, the `logout` method should always be used.

The Rollbase SOAP API uses the same permissions mechanism as the standard Rollbase user interface. API users must have permission to view, create, update, or delete records in order to perform these actions via API calls. Permissions cannot be set via the API. They can only be set by an Administrator using the Rollbase user interface in the Setup area.

Working with field data: the DataField container

When working with records using the Rollbase SOAP API, you will need to make use of a special container called `DataField`. This serves as a container for data stored in a single record field, and it includes the following attributes:

- **Integration name (mandatory):** This is the integration name of the field the data belongs to
- **Mutually exclusive data value(s):** This is the actual data that belongs to this field in this record. Only one of the following data types can be used for a given field.
 - String
 - Long integer
 - Array of long integers (used for lookup fields and multi-select picklists)
 - Decimal number
 - Boolean
 - Date

SOAP API URL

All Web Services API calls must be made using the following URL:

<http://www.rollbase.com/webapi/services/rpcrouter>

SOAP API WSDL

The Rollbase SOAP API WSDL is located at:

<http://www.rollbase.com/webapi/wSDL/rollbase.wsdl>

API Documentation

login

Performs a user login and initiates an API session. This method must be called prior to any other API call.

API method name	login
Parameters	
String loginName	Login name for an active Rollbase user account
String password	User password
Output	Session ID, which must be used in all subsequent API calls during this session

logout

Terminates the specified API session. This method should be called to explicitly end each Web API session.

API method name	logout
Parameters	
String sessionId	Session ID obtained at login
Output	None

getObjectDef

Retrieves a full description of a Rollbase Object Definition as an XML document. The XML schema for this document can be found at:

<http://www.rollbase.com/webapi/wsd/objDefDescription.xsd>

API method name	getObjectDef
Parameters	
String sessionId	Session ID obtained at login
String objDefName	Integration name for the object definition to retrieve
Output	XML document with object definition description

search

Performs a full-text search in the Rollbase database. The search query has the same syntax as that used in the standard Rollbase interface. (For more information on search syntax, see the Rollbase User Guide or the online Help system.)

API method name	search
Parameters	
String sessionId	Session ID obtained at login
String query	Query for full-text search
String objDefName	Integration name for selected object definition. This parameter is optional and allows you to narrow the search to a specified object.
Output	Ids of all records found in the search, returned as an array of longs

getUpdated

Retrieves an array of record ids of a specified object type, which were either created or updated within the given date/time interval.

API method name	getUpdated
Parameters	
String sessionId	Session ID obtained at login
Date from	Beginning of date/time interval
Date till	End of date/time interval
String objDefName	Integration name for object definition to search
Output	Ids of all records found in the search, returned as an array of longs

getDataObj

Retrieves all field data for a given record.

API method name	getDataObj
Parameters	
String sessionId	Session ID obtained at login
long id	Record's id. (This id can be obtained using the search method)
Output	All of the field data for the specified object record.

	Each field's value is wrapped in a <code>DataField</code> container.
--	--

getDataField

Retrieves the value of a single field from a specific record.

API method name	<code>getDataField</code>
Parameters	
<code>String sessionId</code>	Session ID obtained at login
<code>long id</code>	Record's id. (If necessary this id can be obtained using the <code>search</code> method)
<code>String fieldName</code>	Field's integration name
Output	Field's value wrapped in a <code>DataField</code> container.

getBinaryData

Retrieves the binary data (file attachment) associated with a particular file field from a specific record.

API method name	<code>getBinaryData</code>
Parameters	
<code>String sessionId</code>	Session ID obtained at login
<code>long id</code>	Record's id. (If necessary this id can be obtained using the <code>search</code> method)
<code>String fieldName</code>	File field's integration name
Output	Binary data (file attachment) from specified file field wrapped in a <code>ByteArr</code>

getRelationships

Retrieves an array of related record ids.

API method name	<code>getRelationships</code>
Parameters	
<code>String sessionId</code>	Session ID obtained at login
<code>long id</code>	Record's id. (If necessary this id can be obtained using the <code>search</code> method)
<code>String relName</code>	Relationship's integration name

Output	Ids of related records wrapped in a LongArr
---------------	---

create

Creates a new record.

API method name	create
Parameters	
String sessionId	Session ID obtained at login
String objDefName	Integration name for object definition to search
DataFieldArr arr	Values for fields of newly created object. (Note: Record id and auto-numbers will be assigned automatically.)
Output	Id of newly created record as a long

update

Updates an existing record.

API method name	update
Parameters	
String sessionId	Session ID obtained at login
long id	Record's id. (If necessary this id can be obtained using the search method)
DataFieldArr arr	Values for fields to be updated.
Output	None

setDataField

Sets the value of a single field for a specified record.

API method name	setDataField
Parameters	
String sessionId	Session ID obtained at login
long id	Record's id. (If necessary this id can be obtained using the search method)
DataField df	Field's new value wrapped in a DataField container.
Output	None

setBinaryData

Sets the binary data (file attachment) associated with a particular file field from a specific record

API method name	setBinaryData
Parameters	
String sessionId	Session ID obtained at login
long id	Record's id. (If necessary this id can be obtained using the search method)
String fieldName	File field's integration name
String contentType	MIME content type of data. For example: text/html text/plain text/plain application/pdf application/msword application/excel application/vnd.ms-powerpoint application/wordperfect5.1 application/rtf
String suggestedFileName	Suggested file name (with valid file extension)
ByteArr binData	File's binary data wrapped in a ByteArr
Output	None

setRelationships

Assigns an array of related records to the specified record.

API method name	setRelationships
Parameters	
String sessionId	Session ID obtained at login
long id	Record's id. (If necessary this id can be obtained using the search method)
String relName	Relationship's integration name
LongArr arr	Ids of related records
Output	None

delete

Moves the specified record to the Recycle Bin.

API method name	delete
------------------------	--------

Parameters	
String sessionId	Session ID obtained at login
long id	Record's id. (If necessary this id can be obtained using the search method)
Output	None