EWCreateAndRead

Adds a record to your knowledgebase data and reads the result.

Syntax

```
EWWSBaseUserObject o = ew.EWCreateAndRead(String sessionId, String table,
              EWWSBaseUserObject object)
or for the table-specific call:
WSCase case = ew.EWCreateAndRead_WSCase(String sessionId, WSCase object)
```

Usage

Use EWCreateAndRead to add records to the tables, such as Support Case or People, in your knowledgebase and read the result in one call, potentially retrieving values for the fields set by business rules and other back-end logic.

The EWCreateAndRead call is analogous to an INSERT statement in SQL immediately followed by SELECT.

Rules and Guidelines

When creating records, consider the following rules and guidelines:

- The username that was used to obtain the specified session token must have sufficient access rights to create individual records within the specified table. Please verify specific permissions via **Setup > Access > Manage Groups > [Edit Group] > Table > [Select Table] > Permissions**.
- Agiloft allows specifying fine-grained access permissions on the table field level. The username that was used to obtain the specified session token must have sufficient access rights to set every field being specified in the data object supplied in the call when creating the record. Consider this if you are using a record data obtained from or just used in another call like EWRead or EWUpdate - read and update permissions of a particular user may not match the created ones. Please verify specific permissions via **Setup > Access > Manage Groups [Edit Group] > Table > [Select Table] > Field Permissions**.
- The identifier of the new record is calculated automatically unless the field-level permissions allow the client to supply the pre-calculated one and the client supplied one.

- Certain fields can be defined to have default values. If permissions allow, these may be overwritten by the data supplied in the call.
- In the context of overwriting the default values the API and WSDL distinguish between an empty (null) value set explicitly, i.e. the one that should overwrite the default, and a value not set at all in the EWCreate call, i.e. where default should remain as is.
- For required fields that do not have a pre-configured default value, you must supply a value. For all other fields in the record that do not have a pre-configured default value, if you do not explicitly specify a value, then its value is empty (null).
- Agiloft allows one to establish relationships between the records in different tables and ensures the data integrity once the links are forged. See Linked Fields.
- When creating the records in the table it is required to explicitly specify the type of the record. This is especially important in the case when the record is created for the true subtype, such as Contacts.Customer. In this case the table specified in the call will be "Contacts" and the type of the record will be "Customer". For the top-level subtypes the type of the record is equal to the table name. Please consult the Table wizard for the specific names of the tables/subtypes in your KB, available via **Setup > Tables > [Select Table to Edit]**.
- Fields that are present in the table directly are filled in as simple values.
- Some environments like .NET require a special property set for simple-type fields to properly handle empty values - .<name>Specified = true for .NET
- Fields from the Linked Fields relationships that allow values not present in the donor tables are present both in the table, for the case when the value is non-source, and in the linking classes for the case when the value is truly imported and the link is forged.

Linked Fields

Fields that are imported into the table from another table as a part of a Linked Fields set have to be filled via special linking classes.

In WSDL a Linked Fields set takes form of DAO_Dao3_Link<N> field in the complex data structure that corresponds to the target table, where N is a sequential number assigned automatically at the time of the set creation, e.g. WSCase.DAO_Dao3_Link3.

As a value such fields can take one or more WS<Table1><Table2>_Dao3_Link<N> data structures - linking classes, where Table1 is the target table and Table2 is the donor table of the Linked Fields relationship, e.g. WSCaseTeams_Dao3_Link3.

Unfortunately, at this moment one has to rely on investigating the actual sets of fields inside the classes to trace the fields, visible in the Field wizard in the GUI, back to the main object property.

Non-source values for Linked Field sets that allow them are present directly in the table itself additionally to those in the Linking Classes, if the link was in fact forged.

To forge a link the client has to create one or more instances of the linking class and select one of 3 options:

1. Specify the identifier previously obtained through the call to EWSelectFromTable, EWCreate or EWUpdate in the "id" field of the linking class. Please note: some environments like .NET require a special property set for simple-type fields to properly handle empty values - .<name>Specified = true for .NET.
2. Supply example data in the fields of the linking class that will be used to run a Query-by-Example.
3. Supply the SQL statement similar in form to the one used for EWSelectFromTable in the special field "searchSQL". Such SQL is evaluated prior to the creation of the record to identify the actual value for the linked record identifier(s). This evaluation is done in exactly the same manner as when selecting from the table. Normally this method would be used if the query is indirect, such as "where full_name like 'John%'.

See examples of all of the above in the sample code below.

If any of the query mechanisms are used to identify the related records, one has to take into account if the relationship allows multiple records to be linked. If the query returns multiple results and the relationship doesn't allow multiple records to be linked an EWOperationException is thrown.

Choice Fields

The values for choice columns should be supplied as instance(s) of the enumerated types described in the WSDL.

The original text values have undergone the following transformations:

1. Spaces replaced by "_"
2. Dashes replaced by "*MINUS*"
3. Pluses replaced by "*PLUS*"
4. Prefixed with "OPTION_"
5. Converted to upper case

One has to perform the reverse transformation to get to the text value.

Unsupported Types of Fields

Related tables and embedded search results are not supported by the SOAP interface.

Basic Steps for Creating and Reading Records in One Call

Creating records involves the following basic steps:

1. Instantiate the concrete subclass of EWWSBaseUserObject such as WSCase or WSContact. Populate its fields with the data that you want to add.
2. Call EWCreateAndRead, passing in the table name and the data map or the table-specific variant with the data map only.
3. Process the results.

Example Task

In MyKB knowledgebase as user A create a case describing a lost present, with High priority, linked to contact #125, owned by a Support Person and assigned to the Support Team. Return the value from "date_created" field.

The task is completed by performing the following steps:

1. Login to MyKB with "A" and "password" and English as the local language.
2. Create a new data structure, fill in the fields.
3. Specify a customer link to the Contacts record 124.
4. Specify the assignee Support Team from the Team table.
5. Specify an ownership link to the Contact record identified by the full name "Support Person".
6. Create the record.
7. Get date_created value from the returned object.
8. Logout.

Sample Code - Java

```
public Date create() {
             EWServiceAPI binding = new EWServiceAPIServiceLocator().getMyKB();
             try {
             String sessionId = binding.EWLogin("MyKB", "A", "password", "en");
             WSCase wsCase = new WSCase();
             wsCase.setSummary("A case of a lost present");
             wsCase.setProblem_description("People keep losing things"+
             +" bought for their dearest.");
             WSCaseContacts_Dao3_Link5 contact = new WSCaseContacts_Dao3_Link5();
             contact.setId(125L); // forge link by specifying the identifier
             wsCase.setDAO_Dao3_Link5(contact);
             WSCaseTeams_Dao3_Link3 assignedTo = new WSCaseTeams_Dao3_Link3();
             assignedTo.setAssigned_To("Support Team"); //forge link by QBE
             wsCase.setDAO_Dao3_Link3(assignedTo);
             wsCase.setPriority(WSChoice_Priority.OPTION_HIGH);
             WSCaseContacts_Dao3_Link7 ownedBy = new WSCaseContacts_Dao3_Link7();
             ownedBy.setSearchSQL("full_name='Support Person'"); //link by SQL
             wsCase.setDAO_Dao3_Link7(ownedBy);
             wsCase.setType("case");
             WSCase wsCase = (WSCase) binding.EWCreateAndRead(sessionId, "case",
                                               wsCase);
             return wsCase.getDate_Created();
             } finally {
             binding.EWLogout(sessionId);
             }
   }
```

You can generate sample a Web Services code for any table by selecting **Setup > Tables > [Edit Table] > API > Download Sample**.

Arguments

Name	Type	Description
sessionId	String	Session token
tableName	String	The name of the table where the record is to be created (only for generic methods).

obj	EWWSBaseUserObject or a specific descendent like WSCase or WSContact for table-specific calls.	The descendant of EWWSBaseUserObject - one of the complex types described in the WSDL that correspond to the tables on the Agiloft side

Response

The record data as a descendant of EWWSBaseUserObject - a complex structure described in WSDL.

Faults

EWSessionException - client not logged in or the session has expired; client should re-login.

EWPermissionException - username used to create the session lacks sufficient privileges to perform the record modification.

EWWrongDataException - client has supplied the wrong data.

EWOperationException - creation operation has been blocked by one of Agiloft functionalities.

EWIntegrityException - specified table cannot be found or its primary key cannot be identified.

EWUnexpectedException - an unexpected exception has happened; user should report this for investigation.