

EWRead

Retrieves the record data from your knowledgebase.

Syntax

```
EWWSBaseUserObject o = ew.EWRead(String sessionId, String  
    tableName, long id);  
or for the table-specific call:  
WSCase wsCase = ew.EWRead_WSCase(String sessionId, long id);
```

Usage

Use the EWRead call to retrieve individual records from the table. The client application passes the tableName and an identifier of the record to be read in the generic call; and just the record identifier in the table-specific call.

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 read individual records within the specified table. Please verify specific permissions by navigating to **Setup > Access > Manage Groups**, editing the relevant group, clicking the Table tab, editing the relevant table and clicking Permissions.
- Agiloft allows specifying fine-grained access permissions on the field level. The username that was used to obtain the specified session token must have sufficient access rights to be able to read field content. Please verify specific permissions via by navigating to **Setup > Access > Manage Groups**, editing the relevant group, clicking the Table tab, editing the relevant table and clicking Field Permissions..
- This call does not return records that have been deleted.
- This call never returns null when the record is not found.

- In general, you use EWRead when you know in advance the identifiers of the records to retrieve. The client application may use the EWSelectFromTable call to obtain record identifiers beforehand or take the identifiers from the id field of the data structures and the linking classes.
- Client applications can use EWRead to perform a client-side join. For example, a client application can run a query to obtain a set of opportunity records, iterate through the returned opportunity records, obtain the accountId for each opportunity, and then call EWRead to obtain account information for those accountIds.

Linked Fields

Values for the fields that are imported into the target table from the donor table as a part of a Linked Fields set are available via special linking classes.

In WSDL a Linked Fields set takes form of a 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, such as 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, such as 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.

Choice Fields

The values for choice columns are returned 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 Reading Records

Reading records involves the following basic steps:

1. Determine the id of the record you want to update. You may want to use the EWSelectFromTable call to get the identifiers of the records based on some search condition or get the identifier from a previous EWCreate call, or get the id of a linked record from the linking class after performing another EWRead or EWUpdate.
2. Call EWRead.
3. Process the results.

Example Task

In MyKB knowledgebase as user A, read case #456 and return the value from the "date_created" field.

This task is completed by performing the following steps:

1. Login to MyKB with "A" and "password" and English as the local language.
2. Read record 456 from table Case.
3. Get date_created value.
4. Logout.

Sample Code - Java

```
public Date read() throws Exception {
    EWServiceAPI binding = new EWServiceAPIServiceLocator().getDemo();
    try {
        String sessionId = binding.EWLogin("MyKB", "A", "password", "en");
        WSCase wsCase = (WSCase) binding.EWRead(sessionId, "case", 456);
        return wsCase.getDate_Created();
    } finally {
```

```
binding.EWLogout(sessionId);  
}  
}
```

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

Arguments

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

Response

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

Faults

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

EWPermissionException - user used to create the session lacks the sufficient privileges to read the record.

EWWrongDataException - client has supplied the wrong data.

EWOperationException - the operation has been blocked by an Agiloft function such as a table-level lock.

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

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