

EWSelectAndRead

Executes a query against the specified table and returns an array of record data for the records that match the specified criteria.

Syntax

```
EWWSBaseUserObject[] os = ew.EWSelectFromTable(String sessionId, String
        tableName, String whereClause);
or for the table-specific call:
WSCase[] cases = ew.EWCreateAndRead_WSCase(String sessionId, String whereClause)
```

Usage

Use EWSelectFromTable call to search for records in the specified table.

The SQL query supplied should only include the "where" clause from a construct similar to:

```
select primary-key-field-name from table-name where ...
```

Rules and Guidelines

When querying 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 via **Setup > Access > Manage Groups > [Edit Group] > Table > [Select Table] > 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 **Setup > Access > Manage Groups [Edit Group] > Table > [Select Table] > Field Permissions**.
- This call does not return records that have been deleted.

- The queries operate on the database level and should be written in terms of names visible in the Agiloft GUI in the Table wizard in the dbname column. Also see EWGetChoiceLineId for special handling of choice values.
- This method never returns null. In the case when no records are found an empty array is returned.
- Special note on memory management: As WS integration implies pass-by-value semantics, the memory allocated for the resulting array will be released once the data is sent to the client and the server-side JVM's garbage collector considers it eligible for discarding.
- If the client-side environment is one with a garbage collector, the memory used by client-side array will be cleared once the client-side garbage collector considers it eligible,
- When the client environment uses explicit memory management, the client is responsible for freeing up the used memory explicitly.
- It is possible to limit the number of records returned through means available to the underlying database used by your instance of Agiloft e.g. "limit 0,200" for MySQL will limit the number of the returned records to 200.
- It is not possible currently to influence the sequence in which records are returned. They are returned in the effectively arbitrary default order returned by the underlying database.
- When using EWSelectAndRead method all fields of the resulting records are read and passed to the client.

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 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 - not visible in the GUI.

Non-source values for Linked Field sets that allow them are present directly in the table itself and 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 Selecting and Reading Records in One Call

1. Compose the whereClause.
Note: "Prepared" statements are not supported, all parameters have to be passed inside of the whereClause string.
2. Perform the call.
3. Handle results, specifically the situations where there are no elements, one element, or more than one element in the returned array.

Example Task

In MyKB knowledgebase, as user A, find all cases assigned to John Doe. Return summaries as a String array.

The task is completed by performing the following steps:

1. Login to MyKB with "A" and "password" and English as the local language.
2. Search for cases assigned to John Doe.
3. Logout

Sample Code - Java

```
public String[] select() throws Exception {
    EWServiceAPI binding = new EWServiceAPIServiceLocator().getDemo();
    try {
        String sessionId = binding.EWLogin("MyKB", "A", "password", "en");
        EWWSBaseUserObject[] records = binding.EWSelectAndRead(sessionId,
                                                                "case", "assignee='John Doe'");
        String[] result = new String[records.length];
        for (int i=0; i<records.length; i++) {
            result[i] = records[i].getSummary();
        }
        return result;
    } finally {
        binding.EWLogout(sessionId);
    }
}
```

You can generate sample 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 query has to be performed - only for generic methods
where	string	The where clause of the SQL select construct

Response

An array of the records as descendants of EWWSBaseUserObject - a complex structure described in WSDL.

Faults

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

EWPermissionException - user used to create the session lacks sufficient privileges to run the query.

EWWrongDataException - client has supplied wrong data.

EWOperationException - the operation has been blocked by an Agiloft function, e.g. table-level lock.

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

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