EWSearchTablePaginated

Executes a pre-configured named Saved Search against the specified table and returns and array of record data for the records that match the search. This operation variant allows iteration through results page by page, also called pagination.

Syntax

```
EWWSBaseUserObject[] os = ew.EWSearchTablePaginated(String sessionId, String
              tableName, String[] fieldNames, String searchName, int page, int
limit);
```

Usage

Use the EWSearchTablePaginated call to search for records in the specified table based on a Saved Search pre-configured in the GUI. Specify page and limit parameters to retrieve data page by page.

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 > [*Select Group to Edit*] > Table > [*Select Table*] > Field Permissions**.
- This call does not return records that have been deleted.
- 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 server-side JVM's garbage collector considers it eligible for discarding.
- If the client-side environment is the 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.
- When using the EWSearchTablePaginated method, only the fields explicitly listed in the call are read. Within the values returned for the fields requested explicitly, a null value, `nillable="true"`, means the actual null value was retrieved. However, the rest of the fields – those not listed in the fieldNames array – will also appear on the wire as `nillable="true"` elements due to limitations of the underlying Web Services stack.
- To read all fields, use "*" string constant as the only element in the fieldNames array.
- Page numbers start with 0 (zero). A limit – or page size – value of zero indicates "all records" and so all records are returned on page 0 when limit 0 is specified; otherwise, with a non-zero page number, an empty result is returned – no records.
- When a page is not found, empty result is returned.
- A call using pagination always returns a page worth of data. However, to truly take advantage of pagination all other parameters must remain the same. If the table, fields, saved search or limit on a subsequent call is different from the previous one, the underlying query is automatically rebuilt and re-run.
- As such only one "open" query is allowed per client session. If the client requires multiple queries to be iterated in parallel, the client code should create multiple sessions using the same login credentials.
- This method doesn't support multi-threading – the client is responsible for restricting access to a single thread i.e. one client thread = one session = one open query.
- The query will remain "open" until the session is closed – an explicit logout is performed by the client or session timeout occurs – or the application server discards the underlying low-level objects as a result of resource management. In this case the query will be rebuilt and re-run automatically on the next call.
- If the query is rebuilt and re-run, the result of the next call may not be fully consistent with the results of the previous call, as the underlying data may have changed, for example a record was deleted, or the sort order for the search in question was changed. Therefore, the dataset may appear to have "holes" and/or the logical page boundaries may shift when iterating the query page by page.

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 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, for example WSCase.DAO_Dao3_Link3.

As a value such fields can take one or more WS_Dao3_Link data structures - linking classes, where Table1 is the target table and Table2 is the donor table of the Linked Fields relationship, for example 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, which are visible in the Field wizard in the GUI, back to the main object property, which is 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 instances 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

Perform the reverse transformation to get to the text value.

Unsupported Types of Fields

Embedded search results are not supported by the SOAP interface.

Basic Steps for Searching Records with a Pre-configured Saved Search

1. Create a Saved Search in the GUI.

2. Perform the call using the name of the search as the "searchName" parameter.
3. Specify page and limit values.
4. Handle the results, specifically the situations where there are no elements, one element, or more than one element in the returned array.
5. Iterate to the next page as required.

Example Task

In MyKB knowledgebase, as user A, find all cases assigned to the user used to login. Return the first 40 summaries as two String arrays, up to 20 elements each.

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 using My Assigned search.
3. Logout.

Sample Code - Java

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

```
 public String[][] search() throws Exception {
  EWServiceAPI binding = new EWServiceAPIServiceLocator().getDemo()
  try {
    String sessionId = binding.EWLogin("MyKB", "A", "password", "en");
    String[][] result = new String[2][];
    EWWSBaseUserObject[] records1 = binding.EWSearchTablePaginated(sessionId,
"case",
      new String[] {"summary"}, "My Assigned", 0, 20);
    result[0] = new String[records1.length];
    for (int i=0; i<records1.length; i++) {
      result[0][i] = records1[i].getSummary();
    }
    EWWSBaseUserObject[] records2 = binding.EWSearchTablePaginated(sessionId,
"case",
      new String[] {"summary"}, "My Assigned", 1, 20);
    result[1] = new String[records2.length];
    for (int j=0; j<records2.length; j++) {
      result[1][j] = records2[j].getSummary();
    }
```

```
    return result;
  } finally {
    binding.EWLogout(sessionId);
  }
}
```

Arguments

Name	Type	Description
sessionId	String	Session token.
tableName	String	The name of the table where the query has to be performed.
fields	String array	The list of fields to read.
searchName	String	The name of the Saved Search to run.
page	int	The page number.
limit	int	The records per page limit, or page size.

Response

An array of the records as descendants of 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 run the query.

EWWrongDataException - client has supplied the wrong data.

EWOperationException - the operation has been blocked by an Agiloft function, for example a table-level lock.

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

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