

ETV Access SOAP

User Documentation

General Information

| | |
|-------------------|--------------------------------------------|
| File names | ETV-Access-SOAP-Documentation-V1.4-EN.docx |
| Version | 1.4 |
| author(s) | Antonino Artese |

Change notices

| Version | Date | Who | Remarks/reason for change |
|---------|------------|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1.0 | 22.12.2020 | Antonino Artese | First version |
| 1.1 | 06.01.2021 | Antonino Artese | Chapter 3.3.1 / Table 8: Adaption object name «state» to «stateCode» and Addition object «type-Code» Chapter 3.3.2 / Table 12: Adaption description search-object «value» |
| 1.2 | 16.02.2021 | Antonino Artese | Chapter 3.3.1 / Table 8 : Adaption object name «geoSearch» to «geo» and example «geo» |
| 1.3 | 21.04.2021 | Antonino Artese | Chapter 3.3.1 / Table 11: Introduction parameter «Precision-Group-Type» Chapter 4 / Tables 16/17: Addition «Authenticata-tion» error message |
| 1.4 | 29.03.2023 | Bernard Ljuljdjurovic | Chapter 2.2.1 / Table 2: Adaption SOAP Authentica-tion XML example |

I. Table of contents

| | | |
|----------|-------------------------------------|-----------|
| I. | TABLE OF CONTENTS..... | II |
| II. | LIST OF ABBREVIATIONS..... | III |
| 1 | INTRODUCTION | 4 |
| 2 | AUTHENTICATION..... | 5 |
| 2.1 | OPEN USER ACCOUNT | 5 |
| 2.2 | OPEN AUTHENTICATION..... | 5 |
| 2.2.1 | <i>Access Points Login</i> | 6 |
| 2.2.2 | <i>WSDL Authentication</i> | 7 |
| 3 | SOAP SERVICE..... | 8 |
| 3.1 | ACCESS POINTS SEARCH SERVICE..... | 8 |
| 3.2 | WSDL SEARCH SERVICE..... | 8 |
| 3.3 | FUNCTIONALITIES SEARCHSERVICE | 9 |
| 3.3.1 | <i>Search</i> | 9 |
| 3.3.2 | <i>Lookup</i> | 13 |
| 3.3.3 | <i>GetAddress</i> | 13 |
| 3.3.4 | <i>SearchLocation</i> | 13 |
| 4 | ERROR CODE LIST | 14 |
| 5 | INFORMATIONS-CODE LIST | 16 |
| 6 | SUPPORT | 17 |
| 7 | LIST OF FIGURES | 18 |
| 8 | LIST OF TABLES | 19 |

II. List of abbreviations

| | |
|-----------|------------------------------------|
| API | Application Programming Interface |
| CORS | Cross-Origin Resource Sharing |
| ETV | Electronic telephone directory |
| HTTPS | Hypertext Transfer Protocol Secure |
| OAuth 2.0 | Open Authentication Version 2 |
| LV03 | National survey |
| SOAP | Single Object Access Protocol |
| TLS | Transport Layer Security |
| URL | Uniform Resource Locator |
| WCF | Windows Communication Foundation |
| WSDL | Web Service Description Language |

1 Introduction

The ETV Access Webservices offer professional access to the daily updated directory data of Swisscom Directories AG.

All ETV Inside functionalities are available both as RESTful and WCF SOAP web services. This documentation describes the SOAP Webservices.

2 Authentication

Opening an OAuth2 user account is a prerequisite for using the ETV Access APIs. The user account (e-mail address/password) is required to log in to the Open Authentication Server and to technically call the APIs in the production and integration environments. The user account is valid for both environments.

2.1 Open user account

The following is the procedure for creating an account:

- a. Call up the website <https://api.multisource.ch/admin>
- b. Open an account (register with an e-mail address)
- c. Activate your account via the mail sent to you
- d. Call up user profile via URL <https://auth.local.ch/secure/me>



Figure 1: OAuth User Profile with «Identification» and «Username»

- e. Report the information «username» (email address of the OAuth account) and «Identification» to etv@directoriesdata.ch for activation

Afterwards directoriesDATA will activate the account and inform you via e-mail.

2.2 Open Authentication

The ETV Access Search Services are accessed via the Open Authentication Server from localsearch.

By submitting a request to the OAuth server, you will receive a valid 'Access Token' to access the ETV Access functionality. The following figure shows the process for obtaining a valid Access Token.

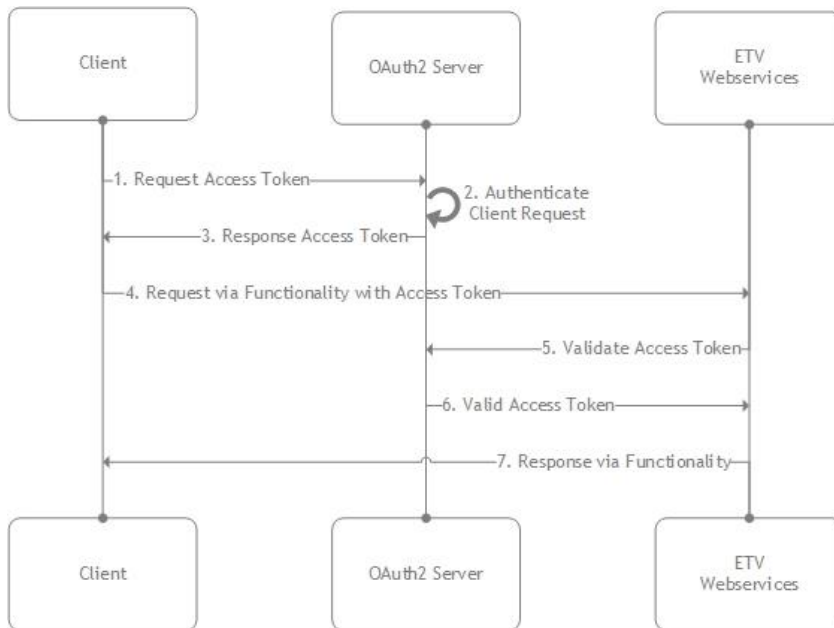


Figure 2: Open Authentication process

2.2.1 Access Points Login

The following are the access points for obtaining a valid Access Token.

Table 1: Access points for authentication

| Access points | Description | Method |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|--------|
| https://apitep.multisource.ch/soap/Services/AuthenticationService.svc | Integration | POST |
| https://api.multisource.ch/soap/Services/AuthenticationService.svc | Production | POST |

Binding: basicHttpBinding

Security Mode: Transport clientCredentialType=None

Table 2: Reference Access Token

| Function | Description | Code Example | XML |
|-------------|------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| AccessToken | Obtaining Access Token | <pre> AuthenticationClient _Auth = new AuthenticationClient(); string _Token = _Auth.AccessToken("xxx", "xxx"); Console.WriteLine(string.Format("Token: {0}", _Token)); _Auth.Close(); </pre> | <pre> <s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"><s:Body><AccessToken xmlns="https://api.multisource.ch/soap"> <Username>ENTER USER </Username> <Password>ENTER PASSWORD</Password> </AccessToken></s:Body></s:Envelope> </pre> |

The parameter values marked in red must be replaced with the e-mail address and password of the previously created OAuth2 user account (see chapter 2.1).

If the API is integrated via JavaScript, the request domain must be stored as a valid access domain (CORS).

You will receive a valid Access Token as a response.

Table 3: Response Token Query

| XML | Description |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre><s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"> <s:Body> <AccessTokenResponse xmlns="http://tempuri.org/"> <AccessTokenResult>253ab17e-fee-4882-8518-09076862736a</AccessTokenResult> </AccessTokenResponse> </s:Body> </s:Envelope></pre> | <p>The Access Token is valid for 6 hours. The validity of the token is extended by 6 hours if it is used in the last 60 minutes before the expiry time.</p> |

2.2.2 WSDL Authentication

Following are the access points for obtaining the Authentication Service description:

Table 4: WSDL ETV Authentication

| WSDL | Description |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------|
| svcutil.exe https://apitep.multisource.ch/soap/Services/AuthenticationService.svc?wsdl | Service description from Client Integration |
| https://apitep.multisource.ch/soap/Services/AuthenticationService.svc?singleWsdL | Service description from Single File Integration |
| svcutil.exe https://api.multisource.ch/soap/Services/AuthenticationService.svc?wsdl | Service description from client production |
| https://api.multisource.ch/soap/Services/AuthenticationService.svc?singleWsdL | Service description from single file production |

3 SOAP Service

All ETV Access functionalities are available via a SOAP web service.

3.1 Access Points Search Service

The ETV Access Search Service is available through the following access points:

Table 5: Search Service access points

| Access points | Description | Method |
|-------------------------------------------------------------------------------------------------------------------------------------------------|-------------|--------|
| https://apitep.multisource.ch/soap/Services/ETVSearchService.svc | Integration | POST |
| https://api.multisource.ch/soap/Services/ETVSearchService.svc | Production | POST |

Access is only possible via https (TLS 1.3). The downgrading of TLS is not allowed. The number of queries per functionality is limited in the **integration** to **100 transactions** per day.

3.2 WSDL Search Service

Below are the access points for obtaining the web service descriptions:

Table 6: WSDL ETV Search Service

| WSDL | Description |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------|
| svcutil.exe https://apitep.multisource.ch/soap/Services/ETVSearchService.svc?wsdl | Service description from Client Integration |
| https://apitep.multisource.ch/soap/Services/ETVSearchService.svc?singleWsd | Service description from Single File Integration |
| svcutil.exe https://api.multisource.ch/soap/Services/ETVSearchService.svc?wsdl | Service description from client production |
| https://api.multisource.ch/soap/Services/ETVSearchService.svc?singleWsd | Service description from single file production |

3.3 Functionalities SearchService

The following API functionalities are available for ETV Access for queries to the published entry data in the telephone book:

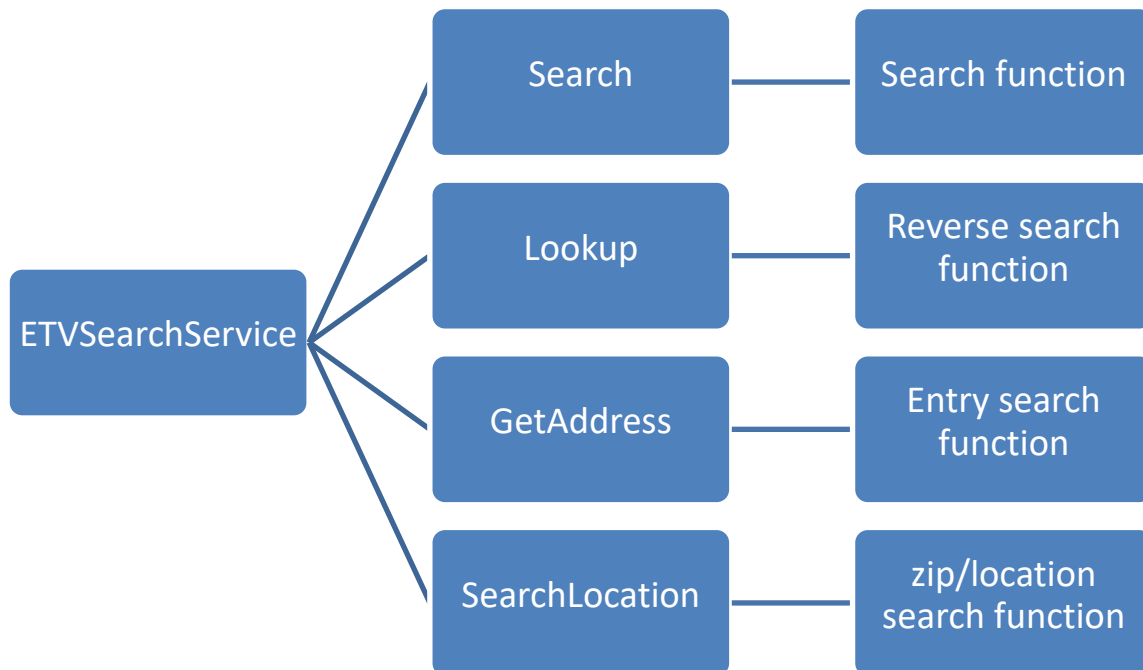


Figure 3: Overview ETV Access Search Service functionalities

Optionally, the searched entries can be enriched with geographic country coordinates (LV03).

The ETV Access Search Service functionalities are optimized for different use cases in terms of performance. The functionalities have to be used according to the intended use.

The following functionalities are available in production and integration:

Table 7: Functionalities ETV Search Service

| Functionality | Description |
|----------------|-------------------------------------------------------------------|
| Search | Search function for obtaining the ETV entry data |
| Lookup | Obtaining the ETV entry data via the "value" object |
| GetAddress | Obtaining the ETV entry data via the "EntryId" object |
| SearchLocation | Obtaining the ETV entry data via the "ZIP" or "Location" objects. |

3.3.1 Search

The Search functionality is used to search the entry data database. The following table lists all objects that can be used for search queries.

Table 8: Search Objects

| Objects | Example | Description |
|------------|-------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| firstname | | first name |
| lastname | localsearch | Last name / Company name |
| femalename | | Ledignames |
| street | Förrlibuckstrasse | Street name |
| houseNo | 62 | House number |
| zip | 8005 | Zip code Location |
| location | Zurich | Place |
| stateCode | ZH | Canton |
| typeCode | BUS | Search restriction according to private (RES) or business (BUS) entries. |
| service | 0800 86 80 86 | Fixnet/mobile numbers, e-mail, URL |
| profession | | Profession |
| category | | Category |
| geo | {"x":"681265;681265","y":"249578;249578"} | <p>Transfer of top-left and bottom-right coordinates.</p> <p>Restriction of search results by coordinates only possible in combination with at least one of the following objects:</p> <ul style="list-style-type: none"> • Load name • First name/Company name • Femalename • Profession • Service • Category |

With the following parameters the Search-Type can be defined individually for each object. On the system side, the search type "Fix-Match" is defined as standard.

Important note: The search types 1 and 2 can influence the response times of the functionality and the effectiveness of the search results.

Table 9: Parameter Search-Type

| Objects | Example | Description |
|----------------------|---------|---------------------------------------------------------------------------------------------------|
| firstnameSearchType | 1 | 0 = Fix-Match Search (Standard) 1 = Like Search 2 = Phonetic Search 3 = Fix-Match Search |
| lastnameSearchType | | |
| streetSearchType | | |
| zipSearchType | | |
| locationSearchType | | |
| professionSearchType | | |

The following parameter can be used to individually limit the amount of results returned by the functionality. The system default is a result set of 5 entries. The maximum allowed result set is 200 entries.

Important note: For customers who are billed with the price model "data elements", the increase of the maximum result quantity can influence the monthly billing.

Table 10: Parameter Result set restriction

| Objects | Example | Description |
|----------------|---------|---------------------------------------------------------------------|
| maxResultCount | 10 | In this example, the functionality returns a maximum of 10 entries. |

With the following parameters the Precision-Group-Type can be defined individually per object. The Precision-Group-Type enables the search for an input value in several database fields.

Important note: The system default is Precision Group Type 0. Precision Group Type 1 may affect the response times of the functionality and the effectiveness of the search results.

Table 11: Parameter Precision-Group-Type

| Objects | Value | Description |
|-------------|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
| firstnamePg | 0 | Search input in field: <ul style="list-style-type: none"> • firstname |
| | 1 | Search input in fields: <ul style="list-style-type: none"> • firstname • lastname • femalename • category |
| lastnamePg | 0 | Search input in field: <ul style="list-style-type: none"> • lastname |

| | | |
|--------------|---|-----------------------------------------------------------------------------------------------------------------------------------------------------|
| | 1 | Search input in fields: <ul style="list-style-type: none"> • firstname • lastname • femalename • category |
| femalenamePg | 0 | Search input in field: <ul style="list-style-type: none"> • femalename |
| | 1 | Search input in fields: <ul style="list-style-type: none"> • firstname • lastname • femalename • category |
| categoryPg | 0 | Search input in field: <ul style="list-style-type: none"> • category |
| | 1 | Search input in fields: <ul style="list-style-type: none"> • firstname • lastname • femalename • category |
| professionPg | 0 | Search input in field: <ul style="list-style-type: none"> • profession |
| | 1 | Search input in fields: <ul style="list-style-type: none"> • profession • category |

Below is a code example for calling the Search function.

Table 12: Example Code Search Function

| Example code search function |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> ETVSearchClient _Etv = new ETVSearchClient(); var _ReturnSearch = _Etv.Search(_Token, new EntryAdd() { Lastname = "Swisscom Directories AG", Street = "Förrlibuckstrasse", Zip = "8005", Location = "Zürich"}); Console.WriteLine(string.Format("IsError: {0} ErrorCode: {1} ErrorText: {2}", _ReturnSearch.IsError, _ReturnSearch.ErrorCode, _ReturnSearch.ErrorText)); _ReturnSearch.Data.ForEach(f => f.EtvRecords.ForEach(q => q.EntryAdds.ForEach(e => { </pre> |

```

        Console.WriteLine(string.Format("Firstname: {0} Lastname: {1} Street: {2} Zip: {3} Location: {4}",
e.Firstname, e.Lastname, e.Street, e.Zip, e.Location));
    }));
ETVSearchClient .Close();

```

3.3.2 Lookup

The Lookup functionality is used to break down fixnet and mobile numbers and thus display the corresponding entry information. The following table lists all objects that can be used in lookup queries.

Table 13: Lookup Objects

| Objects | Description | Examples | mandatory fields |
|---------|-----------------------|---------------|------------------|
| value | Fixnet/mobile numbers | 0800 00 00 24 | Yes |

3.3.3 GetAddress

The GetAddress functionality is used to search for specific entries based on the entry identifier. The following table lists all objects that can be used with GetAddress queries.

Table 14: GetAddress Objects

| Objects | Description | Examples | mandatory fields |
|---------|----------------------------|----------|------------------|
| entryId | Identifier entry record | 30731710 | Yes |
| from | Valid from date yyyy-mm-dd | | |
| to | Valid to date yyyy-mm-dd | | |

3.3.4 SearchLocation

The functionality SearchLocation is used for the targeted search of specific entries based on the entry identifier. The following table lists all objects that can be used in SearchLocation queries.

Table 15: SearchLocation Objects

| Objects | Description | Examples | mandatory fields |
|---------|-------------|----------|------------------|
| zip | Zip code | 8005 | |
| name | Place | Zurich | |

4 Error Code List

In case of an error, the ETV Access functionalities return error codes. The errors are used for troubleshooting or for localizing the cause of the error. It is recommended to check every response for errors.

In the following table the error code categories are listed.

Table 16: Error Code Categories

| Code | Category |
|------|--------------------------|
| 200 | Authentication |
| 401 | Authorization |
| 500 | Internal execution error |

The following table describes the error codes.

Table 17: Error List

| Group | Code | Sub-Code | Message | Description |
|-----------------|------|----------|-----------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------|
| Authen-tication | 200 | | { "error": "invalid_grant", "error_description": "Es existiert kein Konto mit dieser E-Mail-Adresse" } | Wrong oAuth username |
| Authen-tication | 200 | | { "error": "invalid_grant", "error_description": "Das Passwort ist falsch" } | Wrong oAuth password |
| General | 401 | 1 | Requested LocalUserId is not Connected to Multisource Webservice | API: API request was made with a valid token, but the corresponding Local-ID is not yet associated with a multisource webservice user. |
| General | 401 | 2 | No Access to the current Functionality | The user has no authorization to access this functionality. |
| General | 401 | 3 | No Configuration for this User | The user is not configured. |
| General | 401 | 4 | No Product Access. | The user has no access to the product. |
| General | 401 | 5 | The user ist deactivated. | The user is deactivated. |
| General | 500 | 0 | "Exception Message" | An unintercepted error was triggered. A system exception is thrown. |

| | | | | |
|---------|-----|---|-----------------|-----------------------------------------------------------|
| General | 500 | 1 | No Request Data | The functionality requires additional request parameters. |
|---------|-----|---|-----------------|-----------------------------------------------------------|

5 Informations-Code List

The information codes provide general additional information on the query result.

In the following table the info code categories are listed.

Table 18: Info-Code Categories

| Code | Category |
|------|----------------------|
| 100 | General Informations |

The following table describes the information codes.

Table 19: Info-Code List

| Group | Source | Code | SubCode | Message | Description |
|---------------|---------|------|---------|-------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Action Source | General | 100 | 6 | Search returned too much data. Result is reduced to 5. | The search has exceeded the applicable limit for the number of results. The result quantity was restricted to the specified value. The standard result set is 5 entries. |
| Action Source | General | 100 | 7 | Search returned more data than the internal maximum of 200. | The search has exceeded the maximum result set of 200 entries. |

6 Support

If you have technical problems, such as not being able to access a function or if you have questions about your user name or password, please contact etv@directoriesdata.ch for assistance.

7 List of Figures

| | |
|--------------------------------------------------------------------------------|----------|
| <i>Figure 1: oAuth User Profile with «Identification» and «Username»</i> | <i>5</i> |
| <i>Figure 2: Open Authentication process.....</i> | <i>6</i> |
| <i>Figure 3: Overview ETV Access Search Service functionalities.....</i> | <i>9</i> |

8 List of tables

| | |
|---------------------------------------------------------|-----------|
| <i>Table 1: Access points for authentication.....</i> | <i>6</i> |
| <i>Table 2: Reference Access Token.....</i> | <i>6</i> |
| <i>Table 3: Response Token Query</i> | <i>7</i> |
| <i>Table 4: WSDL ETV Authentication.....</i> | <i>7</i> |
| <i>Table 5: Search Service access points</i> | <i>8</i> |
| <i>Table 6: WSDL ETV Search Service.....</i> | <i>8</i> |
| <i>Table 7: Functionalities ETV Search Service.....</i> | <i>9</i> |
| <i>Table 8: Search Objects.....</i> | <i>10</i> |
| <i>Table 9: Parameter Search-Type.....</i> | <i>11</i> |
| <i>Table 10: Parameter Result set restriction</i> | <i>11</i> |
| <i>Table 11: Parameter Precision-Group-Type.....</i> | <i>11</i> |
| <i>Table 12: Example Code Search Function.....</i> | <i>12</i> |
| <i>Table 13: Lookup Objects.....</i> | <i>13</i> |
| <i>Table 14: GetAddress Objects.....</i> | <i>13</i> |
| <i>Table 15: SearchLocation Objects.....</i> | <i>13</i> |
| <i>Table 16: Error Code Categories.....</i> | <i>14</i> |
| <i>Table 17: Error List</i> | <i>14</i> |
| <i>Table 18: Info-Code Categories.....</i> | <i>16</i> |
| <i>Table 19: Info-Code List</i> | <i>16</i> |