Location Intelligence
Infrastructure Asset Management

Confirm®

Confirm Web API

v26.10b.AM

Open Source Attribution Notice

The Confirm suite of products contain the following open source software:

- Feature Data Objects v 3.5.0, which is licensed under GNU Lesser General Public License, Version 2.1, February 1999 with the unRAR restriction. The license can be downloaded from: **http://fdo.osgeo.org/licenceAndGovernance.html**. The source code for this software is available from **http://fdo.osgeo.org/content/fdo-350-downloads**
- MrSID software (specifically the mrsid32.dll) is used under license and is Copyright © 1995-2002, LizardTech, Inc., 1008 Western Ave., Suite 200, Seattle, WA 98104. All rights reserved. MrSID is protected by U.S. Patent No. 5,710,835. Foreign patents are pending. Unauthorized use or duplication prohibited.

  Patented technology in the Software was developed in part through a project at the Los Alamos National Laboratory, funded by the U.S. Government and managed by the University of California. The U.S. Government has reserved rights in the technology, including a non-exclusive, nontransferable, irrevocable, paid-up license to practice or have practiced throughout the world, for or on behalf of the United States, inventions covered by the patent, and has other rights under 35 U.S.C. § 200-212 and applicable implementing regulations.

  For further information, contact Lizardtech.

- NodaTime, version number 1.3.10, which is licensed under the Apache license, version number 2.0. The license can be downloaded from http://www.apache.org/licenses/LICENSE-2.0 . The source code for this software is available from http://nodatime.org/.
- Chromium Embedded Framework, version 3, which is licensed under the New BSD License. The license can be downloaded from http://opensource.org/licenses/BSD-3-Clause. The source code for this software is available from http://code.google.com/p/chromiumembedded/downloads/list.
- Xilium.CefGlue, version 3, which is licensed under the MIT License (with portions licensed under the New BSD License). The licenses can be downloaded from http://opensource.org/licenses/MIT and http://opensource.org/licenses/BSD-3-Clause. The source code for this software is available from http://xilium.bitbucket.org/cefglue/.
- D3 Data Driven Documentation, version 3.4.1, which is licensed under the New BSD License. The license can be downloaded from from https://github.com/mbostock/d3/blob/master/LICENSE. The source code for this software is available from http://d3js.org/.
- OpenLayers, version 8.1, which is licensed under the BSD 2-Clause Licence. The license which can be downloaded from https://github.com/openlayers/openlayers/blob/master/LICENSE.md The source code for this software is available from https://github.com/openlayers/openlayers.
- Proj4js, version 1+, which is licensed under the Apache License, Version 2, January 2004. The license can be downloaded from http://www.apache.org/licenses/LICENSE-2.0.html. The source code for this software is available from http://trac.osgeo.org/proj4js/.
- requireJS, version 2.1.2, which is licensed under the MIT License or the New BSD License. The license can be downloaded from https://github.com/jrburke/requirejs/blob/master/LICENSE. The source code for this software is available from http://requirejs.org/.
- Apache Cordova, version 11.1.0, which is licensed under the Apache License, Version 2, January 2004. The license can be downloaded from http://www.apache.org/licenses/LICENSE-2.0.html. The source code for this software is available from http://phonegap.com/download/.
- Xilium.CefGlue, version 75.1, which is unlicensed. The source code for this software is available from https://gitlab.com/xiliumhq/chromiumembedded/cefglue.

- Chromium Embedded Framework, version 75.0, which is licensed according to the following criteria:

  Copyright (c) 2008-2014 Marshall A. Greenblatt. Portions Copyright (c) 2006-2009 Google Inc. All rights reserved.

  Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

  * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

  * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

  * Neither the name of Google Inc. nor the name Chromium Embedded Framework nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

  THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

  The source code for this software is available from http://opensource.spotify.com/cefbuilds/index.html#

April 16, 2026

Table of Contents

Specifications

Specifications

The following sections outline all the Specifications that exist within the Confirm functionality.

In this section

Confirm Web API

Introduction

The Confirm Web API is a set of RESTful web APIs, which allows retrieval and modification of data in the Confirm database. The APIs are available automatically with the Confirm web interface.

The APIs can be categorized into two types - Create/Update API and Query API, each of which is described below.

Create/Update API

This API can be used to create new records and to modify existing records.

The following entities and operations are currently available:

Entity	Operations
Features	Create a Feature with Feature Conditions, Feature Attributes, Feature Measurements and Feature Dates.
	Update a Feature with Feature Conditions, Feature Attributes, Feature Measurements and Feature Dates.
Jobs	Create a Job with Job Items.
	Update a Job with Job Items.
	Commit a Job.
Defects	Create a Defect with Defect Attributes.
	Update a Defect with Defect Attributes.
CentralEnquiries	Add Images and Documents to an Enquiry.
Payments	Create Payment Batch from supplied Jobs. Generates Items automatically, according to Job's current outstanding Items and the supplied Job Value.

For more detailed information on how to use the Confirm Web API, refer to the schema definition:

- ConfirmWebApi.yaml

Query API

This API is designed to allow 3rd party systems to get data from Confirm.

It is developed using GraphQL language, which gives more flexibility and efficiency. It allows users to ask for what they need and nothing more.

The GraphQL query can be generated either via **Confirm Web - Reporting interface** or directly using GraphQL Introspection.

The query can be captured from Confirm Web - Reporting interface using the browser once a report is run, as described here: **Query API - Query Capture** .

One use of the Query API is to allow an exernal GIS to synchronise data with Confirm, as described here: **Query API - GIS Synchronisation** .

Confirm Web - API Authentication

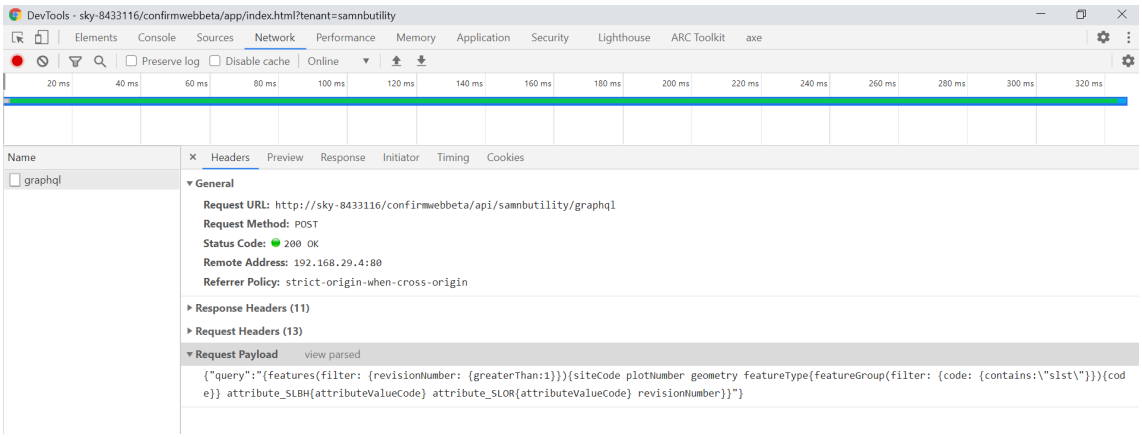
Authentication for the Confirm Web API can be achieved using either OAuth or an API Key linked with a Confirm User. See below for more details of each option:

- **Generating an OAuth token**
- **Authenticating with an API Key**

Query API - Query Capture

The query can be captured from Confirm Web - Reporting interface using the browser once a report is run.

This can be done in multiple ways, developer console in browsers being one of them. Sample screen-shot from chrome browser is as below:

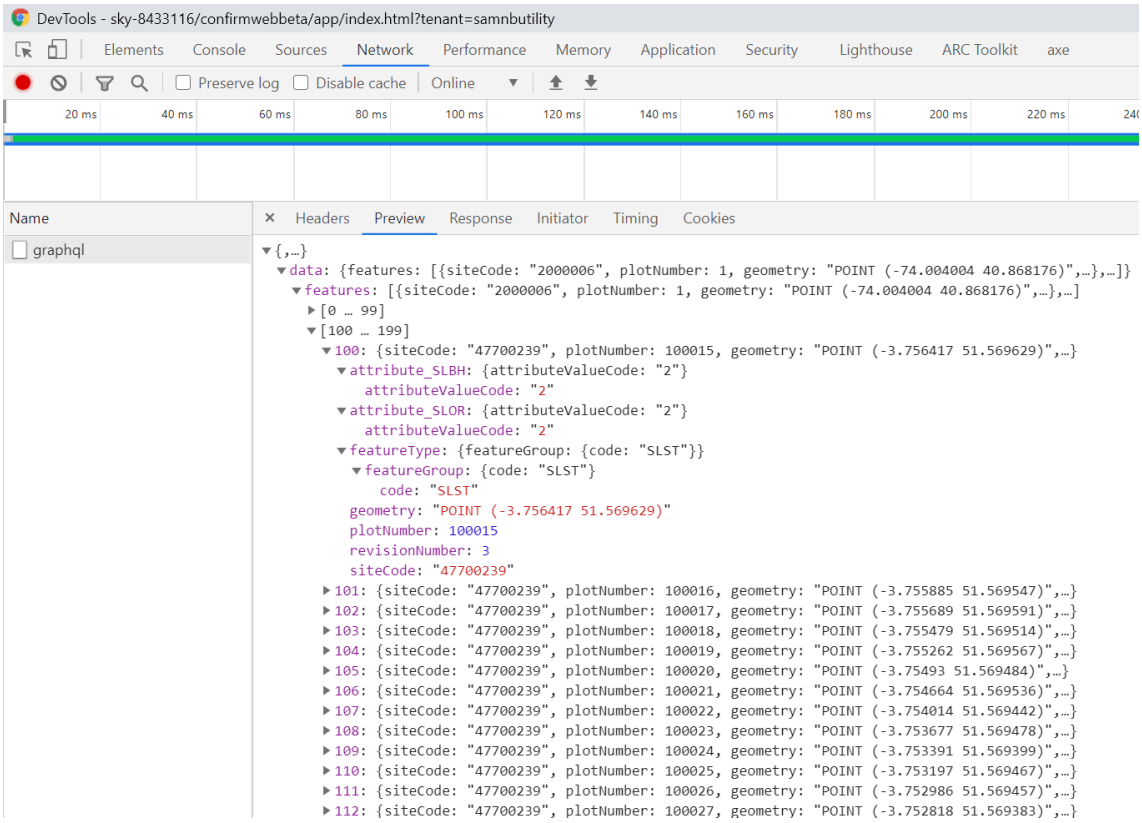


The value of query property (without quotes) within ‘Request Payload’ is the GraphQL query.

Sample GraphQL query is below:

{features(filter: {revisionNumber: {greaterThan:1}}){siteCode plotNumber geometry featureType{featureGroup(filter: {code: {contains:"slst"}}){code}} attribute_SLBH{attributeValueCode} attribute_SLOR{attributeValueCode} revisionNumber}}

The results are returned in the JSON format and can be seen in the developer console of the browser as well. Sample screen-shot from chrome browser below:



The captured query can also be run in a REST API client (like Postman), GraphQL query needs to be supplied as a query parameter in the GET request. Sample URL below:

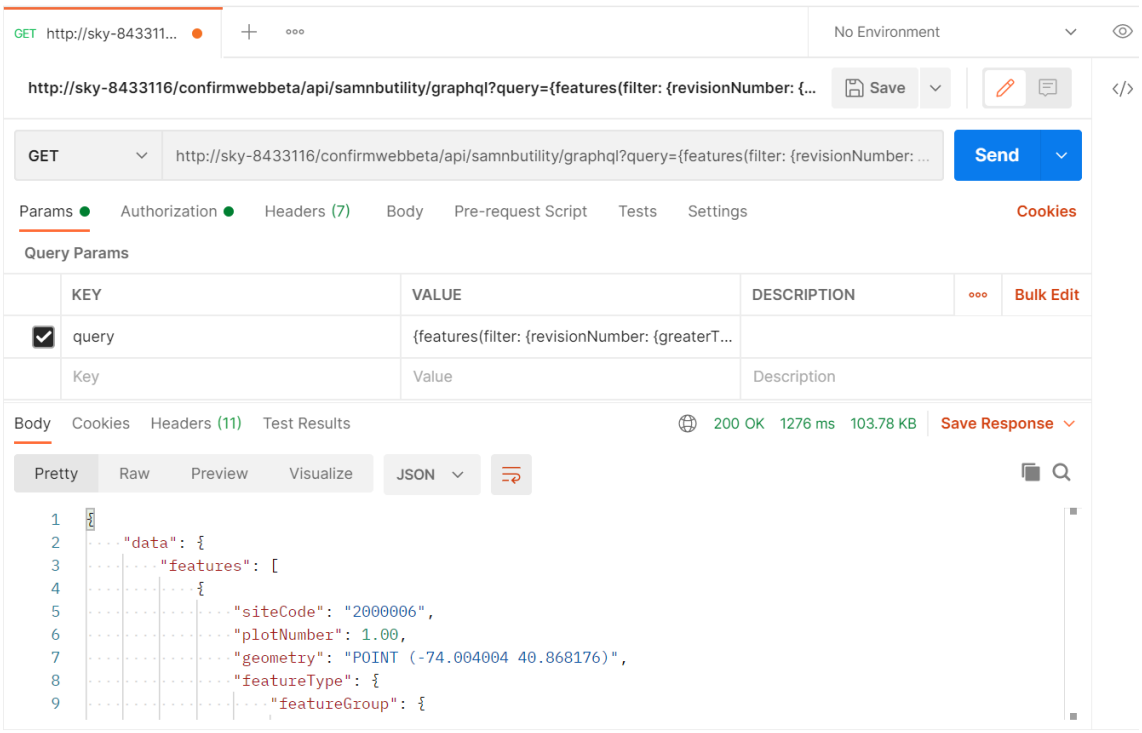
https://ConfirmWebServer/ConfirmWeb/api/DatabaseName/graphql

Query API - GIS Synchronisation

One use of the Query API is GIS Synchronisation, where an external GIS can use the Query API to obtain up to date Feature data from Confirm.

GIS synchronization is based on Revision Number field hence it needs be part of API requests. Revision Number is used to keep track of when a feature was last updated so that only Confirm Features that have changed since the last synchronisation are fetched.

The response of this would be JSON list of features with specified columns filtered by the criteria specified in the query. Below is the sample screen-shot of GET API run in the Postman tool:



Note: A revision number could include up to 1000 features. Hence it is recommended to have a maximum revision number on GraphQL queries when performing initial synchronization in order to limit the results.

Query API - Pagination

Pagination is used to divide a record set into discrete pages with the following parameters:

- pageNumber
- pageSize

The pageNumber specifies the page number, while the pageSize specifies the number of items(records) per page. For instance, if a user has 100 items and wishes to display 10 items per page, there will be a total of 10 pages.

A Summary entity can be included in the request which contains two properties:

- queryName
- totalCount

The queryName refers to the entity name queried by the user, while totalCount represents the total number of records present in the database for the queried entity.

Below is a sample API request which is asking for the the second page of results, where there are 5 items per page:

```
query Features {
    features(
        pagination: { pageNumber: "1", pageSize: "5" }
        filter: { wardCode: { equals: "PT" } }
    ) {
        plotNumber
        siteCode
        startDate
    }
    summary {
```

```
        queryName
        totalCount
    }
}
```

The following is a sample response to the above request:

```
{
    "data": {
        "features": [
            {
                "plotNumber": 7,
                "siteCode": "47700294",
                "startDate": "1990-01-01T00:00:00"
            },
            {
                "plotNumber": 7,
                "siteCode": "47703504",
                "startDate": "1990-01-01T00:00:00"
            },
            {
                "plotNumber": 7,
                "siteCode": "47712347",
                "startDate": "1990-01-01T00:00:00"
            },
            {
                "plotNumber": 7,
                "siteCode": "47712617",
                "startDate": "1990-01-01T00:00:00"
            },
            {
                "plotNumber": 7,
                "siteCode": "47712327",
                "startDate": "1990-01-01T00:00:00"
            }
        ],
        "summary": [
            {
                "queryName": "features",
                "totalCount": 1226
            }
        ]
    }
}
```

Generating an OAuth token

Follow the steps to generate an OAuth token:

1. Obtain your *API Key (Username)* and *Secret (password)* from Confirm system administrator
2. To generate the OAuth Token, encode your credentials (API Key and Secret) using base64 computation mechanism. To do this, provide API KEY and Secret to the base64 encoder (online encoder can be used), and generate the encoded 'base64value'.
3. The following format should be used while computing the {BASE64VALUE}:

   {API KEY}:{SECRET}

4. Enter the generated 'base64value' in the header of the request and call the token URI as shown in Figure 1 below:

   Here, {tenant} is the tenant name and {Confirm web url} is the URL where Confirm web is deployed, like https://ConfirmWebServer/ConfirmWeb/.

5. The access token is returned as Figure 2 below

```
Figure 1:
          Authorization: Basic {base64Value}
          Content-Type: application/x-www-form-urlencoded
          POST {Confirm web url}/api/{tenant}/oauth/token
          grant_type=client_credentials
```

```
Figure 2:
          {
          "access_token": "{your access token as a Base64 encoded
value}",
          "token_type": "bearer",
          "expires_in": {The expiry time in seconds}
          }
```

Authenticating with an API Key

Follow the steps to generate an API Key for a Confirm User and use in a Confirm Web API:

1. On the User Security screen use the API Key button to generate an API for the User. Make sure you store this key securely, since it will not be accessible directly in Confirm again.
2. The following format should be used while computing the {BASE64VALUE}:

   {username}:{api key}

3. Enter the generated 'base64value' in the header of the request and call the Confirm Web URI as shown below:

```
Authorization: Basic {base64Value}
```