



Electronic Data Platform Technical requirements

ELECTRONIC DATA PLATFORM	1
TECHNICAL REQUIREMENTS	1
1. GENERAL INFORMATION	1
1.1 Prerequisites	1
1.1.1 <i>Browser</i>	1
1.1.2 <i>Authentication – User Management</i>	1
1.2 Base URL's	1
2. CONNECTION FOR STANDARD USERS	1
2.1 Accessing the Electronic Data Platform.....	1
2.2 Logging in	1
3. CONNECTION FOR CLIENT SYSTEMS	2
3.1 URL parts for application and download page.....	2
3.2 Authentication Parameters	2
3.2.1 <i>Basic Authentication Method (deprecated approach)</i>	2
3.2.2 <i>Api Gateway (securised approach)</i>	2
3.3 Examples	3
3.3.1 <i>By coding</i>	3
3.3.2 <i>Using Google Chrome ARC</i>	5
3.4 Server Certificate's Root Certification Authority must be trusted.....	7

1. General Information

1.1 Prerequisites

1.1.1 Browser

- For public menu items
 - 2 latest versions of Internet Explorer (right now 10 en 11)
 - Mozilla Firefox 10
 - Chrome 16
 - Safari 5
- For private menu items Fluxys Belgium only supports Internet Explorer 10 or higher
- JavaScript must be enabled
- Cookies must be accepted

1.1.2 Authentication – User Management

- Contact details of SPOC users (i.e. an employee of the customer) must have been supplied to Fluxys Belgium. This information will be used to perform the setup for each Business Party.
- Customers are advised to designate at least two different SPOC's – to be able to deal with absences.
- The customer's SPOC will be able to define and manage users online. For example, when a user would have locked his account, he can contact the SPOC to unlock it and/or to reset the password.
- For an application to be able to use an URL for automatic downloads, the SPOC should create an Automatic Download Agent account. Such an account is best set up with the mobile phone and email address of someone in the technical team. A welcome email is sent once this account has been created by the SPOC, but even more important is that an SMS is sent to the user with his temporary password. This password should be reset manually before gaining access to the Electronic Data Platform. The username and password should be passed as authentication parameters each time a report is automatically downloaded. All information about external user management (XUMA) is available in the XUMA User Manual.

Note: The server will use a Server Certificate to setup a secure connection (https). However, a regular user should normally not worry about this.

1.2 Base URL's

The base URL for the Electronic Data Platform is :

- The EDP of the TSO : <https://gasdata.fluxys.com>

2. Connection for Standard Users

2.1 Accessing the Electronic Data Platform

Standard users access the Electronic Data Platform directly by entering the base URL <https://gasdata.fluxys.com> in a browser.

Initially only public content of the Electronic Data Platform will be displayed.

2.2 Logging in

As mentioned under *1.1 Prerequisites*, users must first be registered and assigned one or more roles by their company's SPOC.

Registered users can login through the "Sign in" link in the top right corner of the screen. This will display a login dialog box where username and password can be provided.

After logging in, the user will return to the Electronic Data Platform home screen, but will see

additional items under the various menus – according to the assigned user rights.

For example, any user that is assigned a 'Reader' role on <https://gasdata.fluxys.com> will be able to see a 'Metering' item with several sub-items like 'Measurements', "Topology"...

Unless the user performs an explicit logout, he will stay logged in as long as he does not close his browser, with a maximum of 12 hours.

3. Connection for Client Systems

Most of the information available manually via the Electronic Data Platform and Electronic Booking System can also be retrieved directly and automatically, which is especially useful for client systems that perform 'automatic downloads'.

3.1 URL parts for application and download page

When using the Electronic Data Platform user interface, the underlying applications are not visible to the end user.

However, for technical interactions such as automatic downloads, it remains necessary to specify the name of the underlying application, followed by the name of the download page.

For downloads (related to the api getaway method), the general format of the URL is thus:

[https://api.gasdata.fluxys.com/\[ApplicationName\]/\[DownloadPageName\]?\[Parameters\]](https://api.gasdata.fluxys.com/[ApplicationName]/[DownloadPageName]?[Parameters])

3.2 Authentication Parameters

3.2.1 Basic Authentication Method (deprecated approach)

NB : This method with credentials passed in the URL is deprecated for security reason and is replaced by the new API Gateway approach. This method will be obsolete and deactivated in the course of 2018. All reports are already available through the API gateway. All users must therefore switch to the new method.

While a user will be presented with a login dialog when accessing a private section of the Electronic Data Platform, applications need to pass their credentials as parameters in the URL. This can be done by specifying username and password as the first two parameters in the URL.

So in practice, any URL that is used for automatic downloads will have the following format:

[https://gasdata.fluxys.com/\[ApplicationName\]/\[DownloadPageName\]?usr=User&pwd=Password&\[OtherParameters\]](https://gasdata.fluxys.com/[ApplicationName]/[DownloadPageName]?usr=User&pwd=Password&[OtherParameters])

Note: Within the context of this document the username and password are always assumed to be "User" and "Password", which of course must be replaced by the username and password of a user that is created by a customer SPOC.

3.2.2 Api Gateway (secured approach)

All reports are now available through this method.
And some reports are only available through this method.

3.2.2.1 Procedure

The following is the procedure to follow to download reports from <https://api.gasdata.fluxys.com>. The implemented method is OAuth2 Client Credentials Grant. Using this standard, a download from <https://api.gasdata.fluxys.com> takes 2 distinct requests to our servers.

3.2.2.2 Authenticate and retrieve a bearer token

In order to retrieve a bearer token, the following https request needs to be made to <https://xumais.gasdata.fluxys.com/connect/token>.

```
POST /connect/token HTTP/1.1
Host: xumais.gasdata.fluxys.com
Authorization: Basic czZCaGRSa3F0MzpnWDFmQmF0M2JW
Content-Type: application/x-www-form-urlencoded

grant_type=client_credentials&scope=fluxys_api
```

As you can see above, the request is authenticated using an 'Authorization: Basic' header. The credentials are encoded in base64 in the format username:password.

After successfully validating the provided Authorization header, the server will return a response similar to the following:

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Cache-Control: no-store
Pragma: no-cache

{
  "access_token": "2YotnFZFEjrlzCsicMWpAA",
  "expires_in": 3600,
}
```

The value associated with the JSON key "access_token" will be used as a Bearer authorization header in the next step. It must be used as is without any transformation.

3.2.2.3 Download a report by providing the bearer token to the api gateway

To download the actual report, make a https GET request to the report url (eg. <https://api.gasdata.fluxys.com/TransmissionHandler/Reports/WebTrack/flowmeasurement/node/new/hourly/get>) making sure to include the bearer token retrieved in step 1 in an 'Authorization: Bearer' header

```
GET
/TransmissionHandler/Reports/WebTrack/flowmeasurement/node/new/hourly/get
HTTP/1.1
Host: api.gasdata.fluxys.com
Authorization: Bearer 2YotnFZFEjrlzCsicMWpAA
Accept: application/xml
```

After validating your access token, the report will be returned. The main accepted format reports are xml and csv to define in the "Accept" header.

3.3 Examples

Those examples illustrate two ways to authenticate and download a report using the API gateway : by coding (C#) or using a API testing tool (Chrome ARC).

3.3.1 By coding

The following snippet C# code demonstrates the procedure described in the previous sections.

```
using System;
using System.Collections.Generic;
using System.Net.Http;
using System.Net.Http.Headers;
using System.Text;
```

```

using Newtonsoft.Json.Linq;

namespace Example
{
    class Program
    {
        static void Main(string[] args)
        {
            var httpClient = new HttpClient();

            //
            // Step 1: retrieve an access token from our token server
            //

            var tokenRequest = new HttpRequestMessage(
                HttpMethod.Post,
                "https://xumais.gasdata.fluxys.com/connect/token");
            tokenRequest.Content = new FormUrlEncodedContent(
                new Dictionary<string, string>()
                {
                    { "grant_type", "client_credentials" },
                    { "scope", "fluxys_api" }
                });

            // pass username and password with a 'Authorization: Basic' header
            tokenRequest.Headers.Authorization = new AuthenticationHeaderValue(
                "Basic",

Convert.ToBase64String(Encoding.UTF8.GetBytes("myusername:mypassword")));

            var tokenResponse = httpClient.SendAsync(tokenRequest).Result;
            tokenResponse.EnsureSuccessStatusCode();

            var tokenEndpointResult =
tokenResponse.Content.ReadAsStringAsync().Result;
            var accessToken =
JObject.Parse(tokenEndpointResult)["access_token"].ToString();

            //
            // Step 2: Get the report passing the access token
            //

            var requestUri =
"https://api.gasdata.fluxys.com/TransmissionHandler/Reports/WebTrack/flowmeasurement/n
ode/new/hourly/get?periodfrom=2016-04-02&periodto=2016-04-02&type=default";
            var reportRequest = new HttpRequestMessage(HttpMethod.Get, requestUri);
            // note: the token retrieved in step 1 is added as an 'Authorization:
Bearer' header
            reportRequest.Headers.Authorization = new
AuthenticationHeaderValue("Bearer", accessToken);
            // note: the type is depending of the report, here an xml report
            reportRequest.Headers.Accept.ParseAdd("application/xml");
            var reportResponse = httpClient.SendAsync(reportRequest).Result;
            reportResponse.EnsureSuccessStatusCode();

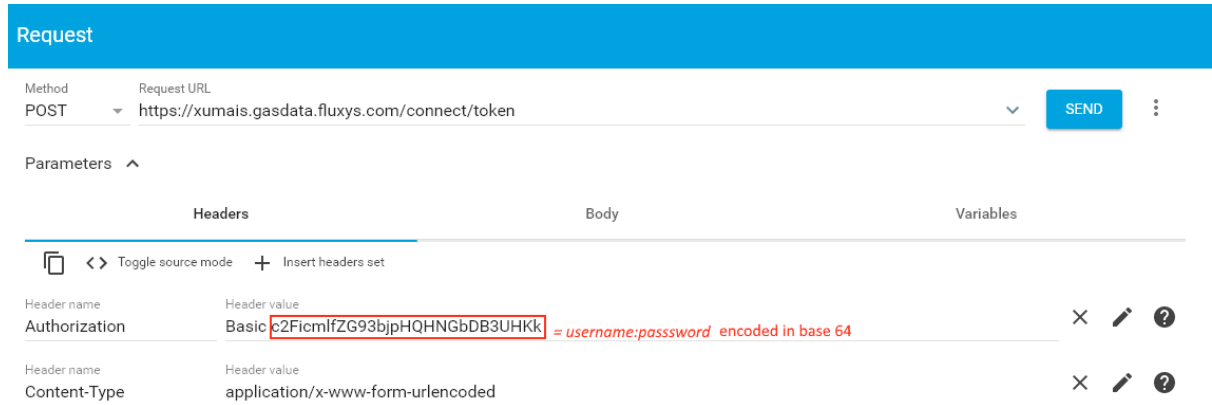
            // reportContent can be used to copy the report content to a file or for
any processing
            var reportContent = reportResponse.Content.ReadAsStreamAsync().Result;
        }
    }
}

```

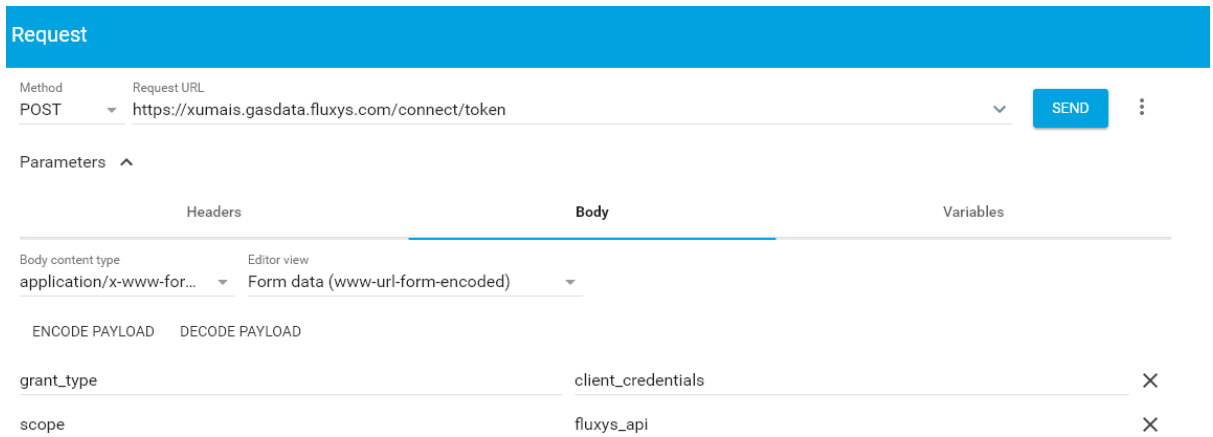
3.3.2 Using Google Chrome ARC

3.3.2.1 Authentication

1. Fill the POST request with <https://xumais.gasdata.fluxys.com/connect/token>
2. Insert header attributes
 - authorization: Basic *username:password* (base64 encoded) . The used credentials are the ones of the automated download agent user created on the EDP platform
 - content-Type: application/x-www-form-urlencoded



3. Insert body
 - grant_type : client_credentials
 - scope : fluxys_api



4. Click on 'Send' and get the token in the access token attribute of the response

3.4 Server Certificate's Root Certification Authority must be trusted

As currently, the Root Certification Authority's Certificate (CA) of the Server Certificate must be trusted on the client for automatic downloads to work. Normally the Root CA will be GlobalSign.