REST API V2 Security

To access the AgileAssets REST API client applications must authenticate using OAuth2.

OAuth2 is an internet standard that secures endpoints using tokens, consumer authorization, and client secrets. The web API uses OLTU on top of OAuth2 to provide authentication.

OAuth2 must authenticate over HTTPS. The token can be intercepted between the client and the server if HTTPS is not used.

OAuth2 Overview

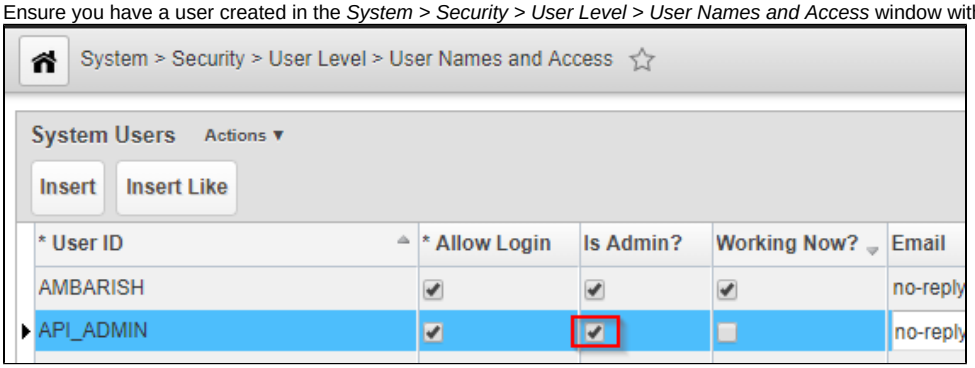
The basic steps are:

1. Establish Client ID and Secret. The Secret is known only to the two parties involved. This is done once only
2. The client obtains an access token and a refresh token.
3. Use the access token on REST API request.
4. When the access token expires, use the refresh token to get a new one.

Create a Client Id and Secret

To use OAuth2 REST API developers will need a client id and secret to be configured first.

1. Ensure you have a user created in the *System* > *Security* > *User Level* > *User Names and Access* window with admin privileges configured.



2. Create a client id and secret

A REST API is provided to create the client id and secret in the AgileAssets system.

Call the following URI with a POST request <base app url>**/rest/oauth2/secret** using the following form parameters x-www-form-urlencoded encoded.

Parameter	Description
redirect_uri	the redirect URI - must be valid
client_id	Textual name of client id e.g. API_VIEWER
grant_type	one of more (comma separated) of : • authorization_code • password • refresh_token

In the header of the request send the following Authorization header:

> To create the header value - Base64 encode your user name and password from Step 1 above(with admin rights). Add a ":" separator between username and password.
>
> Hint: Use an online application like http://www.utilities-online.info/base64 to encode your username password:
>
> e.g. MyUserName:MyPassword equals TXlVc2VyTmFtZTpNeVBhc3N3b3Jk

The POST request should return a result as follows when successful.

```
{
"client_secret": "$2a$12$94oWHSS5lkqNJChC.6JNFOQXBpVmlld2VyWFlaWlk=",
"client_id": "ApiViewer"
}
```

The following shows an example of this in JavaScript:

JavaScript Example

```
var data = "redirect_uri=https%3A%2F%2Fakdot.agileassets.com%
2FAMS_AK_DEV&client_id=ApiViewer&grant_type=authorization_code%2Cpassword%2Crefresh_token";

var xhr = new XMLHttpRequest();
xhr.withCredentials = true;

xhr.addEventListener("readystatechange", function () {
  if (this.readyState === 4) {
    console.log(this.responseText);
  }
});

xhr.open("POST", "https://akdot.agileassets.com/AMS_AK_DEV/rest/oauth2/secret");
xhr.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");
xhr.setRequestHeader("Authorization", "QVBJX0FETUladjdreU1KOTQ5cDNITExmVko=");
xhr.setRequestHeader("Cache-Control", "no-cache");

xhr.send(data);
```

Get Access Token

To use REST API resources and access token is required to use with each request.

For non-interactive application clients using the REST API they need to request the access token using the following methodology.

Call the following URI with a POST request <base app url>**/rest/oauth2/token** using the following form parameters x-www-form-urlencoded encoded.

Parameter	Description
grant_type	Value to use is "password" (don't include quotes)
client_id	Client Id from above steps
client_secret	Client secret return in Json on above steps
username	Valid application user. Configured in *System* > *Security* > *User Level* > *User Names and Access* window Note: A different use than above can be created/used that doesn't have admin rights
password	Valid user password

The request would return the following JSON:

```
{
"access_token": "$2a$12$YT9WGAVMcSP09.qisYMp6OP/VpAUtynPQUqIRtTm9dU6A3lfWMKRW",
"refresh_token": "$2a$12$YT9DGAVMaSP09.qisYMp6OU8GkF5jW7Ay1xuHOXdxoxG3gIeSyjhe",
"token_type": "BEARER",
"expires_in": 43200
}
```

The following shows an example of this in JavaScript:

JavaScript Example

```
var data = "grant_type=password&client_id=ApiViewer&client_secret=%242a%2412%2494oWHSS5lkqNJChC.
6JNFOQXBpVmlld2VyWFlaWlk%3D&username=API_VIEW&password=9XDWm96FLzZfVwqY";

var xhr = new XMLHttpRequest();
xhr.withCredentials = true;

xhr.addEventListener("readystatechange", function () {
if (this.readyState === 4) {
console.log(this.responseText);
}
});

xhr.open("POST", "https://akdot.agileassets.com/AMS_AK_DEV/rest/oauth2/token");
xhr.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");
xhr.setRequestHeader("Cache-Control", "no-cache");
xhr.send(data);
```

Access REST API Resource

Once the access token is received it can be use to access the various REST API resources.

For a list of resources - access the wadl file on each API version.

V1 - <base app url>**/rest/v1/applicationPretty.wadl**

V2 - <base app url>**/rest/v2/applicationPretty.wadl**

When accessing one of the REST API endpoint in the HTTP header include the Authorization header with the value "Bearer <Access Token>".

e.g. for access token above it will be

Authorization : Bearer $2a$12$YT9WGAVMcSP09.qisYMp6OP/VpAUtynPQUqIRtTm9dU6A3lfWMKRW

For example access data via a database view called ROUTE_ID_VIEW in REST API, use a GET request to <base app url>/rest/v1/lookup/view/ROUTE_ID_VIEW

Note: For database views to be visible in the REST API they need to be added in the System > Utilities > Allowed Web API Views window.

The following shows an example of this in JavaScript:

JavaScript Example

```
var data = null;

var xhr = new XMLHttpRequest();
xhr.withCredentials = true;

xhr.addEventListener("readystatechange", function () {
if (this.readyState === 4) {
console.log(this.responseText);
}
});

xhr.open("GET", "https://akdot.agileassets.com/AMS_AK_DEV/rest/v1/lookup/view/ROUTE_ID_VIEW");
xhr.setRequestHeader("Authorization", "Bearer $2a$12$YT9VGAVMcSP09.qissMp6OP/VpAUtynPQUqIRtTm9dU6A3lfWMKRW");
xhr.setRequestHeader("Cache-Control", "no-cache");
xhr.send(data);
```

Refresh Token

The access token expires, and then the refresh token is used to generate a new access token.

Call the following URI with a POST request <base app url>**/rest/oauth2/token** using the following form parameters x-www-form-urlencoded encoded.

Use the Authorization header - base 64 encoding of client id and secret with ":" separator.

eg Basic UG9zdG1hbjpzZWNyZXQ=

Parameter	Description
grant_type	Value to use is "refresh_token" (don't include quotes)
refresh_token	refresh token returned from initial token request JSON

The request would return the following JSON:

```
{
"access_token": "$2a$12$sDsoih.v7C9kA0Va2LhioOTN.lqKg3i1tZuz3GM6e10xodY8lPQLW",
"refresh_token": "$2a$12$YT9WGAVFcSG09.qisYMp6OU8GkF5jW7Ay1xuHOXdxoxG3gIeSyjhe",
"token_type": "BEARER",
"expires_in": 43200
}
```

Use the new access_token to access any further API resources.

The following shows an example of this in JavaScript:

JavaScript Example

```
var data = "grant_type=refresh_token&refresh_token=%242a%2412%24YT9DGAVMaSP09.
qisYMp6OU8GkF5jW7Ay1xuHOXdxoxG3gIeSyjhe";
var xhr = new XMLHttpRequest();
xhr.withCredentials = true;

xhr.addEventListener("readystatechange", function () {
if (this.readyState === 4) {
console.log(this.responseText);
}
});

xhr.open("POST", "https://akdot.agileassets.com/AMS_AK_DEV/rest/oauth2/token");
xhr.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");
xhr.setRequestHeader("Authorization", "Basic
QXBpVmlld2VyOiQyYSQxMiRiLzBwd0RDWGVJWXEzd29XUzN2dmUuUVhCcFZtbGxkMlZ5V0ZsYVdsaz0=");
xhr.setRequestHeader("Cache-Control", "no-cache");
xhr.send(data);
```