USER GUIDE
OAUTH AUTHENTIFICATION

Version 2.0

Effective date: 8 July 2016

CONTENTS

1 Introduction

This document describes the procedures for authenticating RTE's APIs using the OAuth protocol.

1.1 Reference documents

Short reference	Name of the document	Complete reference
[R1]	Terms of use for RTE's APIs	Access link

1.2 Definitions

The terms used in this User Guide (the first letters of which are always capitalised) are defined below. Otherwise, their definitions are given in the General Conditions of Use **[R1]**:

API	Application Programming Interface (Interface de programmation applicative)
Authentification	Protection Mode for ensuring that the identity of the Sender or Receiver has been checked by RTE, and that they are authorised to access the IT system and use the Applications.
URL	Uniform Resource Locator: character string based on a specific format used to locate a resource on a network and specify what protocol should be used on this resource.
User(s)	Legal entity which has agreed to RTE's General Terms and Conditions for Using APIs and which has been granted access to RTE's IT system for the purposes of using the APIs it has made available.
Web Service	Computer programme designed to enable heterogeneous systems to communicate with one another in distributed environments
OAuth 2	OAuth 2.0 (Open Authorisation) is a delegated authorisation framework which enables applications to get tokens for accessing data hosted by third parties.

1.3 Technical support

In the event of difficulties accessing or using an API, Users can contact the telephone support services provided by RTE in accordance with the technical conditions detailed in the General Terms and Conditions of Use.

2 Accessing the APIs

2.1 Authorisation Method

The authorisation method in place is underpinned by the OAuth 2.0 (Open Authorisation) delegated authorisation framework. This enables a client application to access a resource exposed as an API on behalf of its owner via an access token for data hosted by a third party.

OAuth 2.0 RFC: http://tools.ietf.org/html/rfc6749

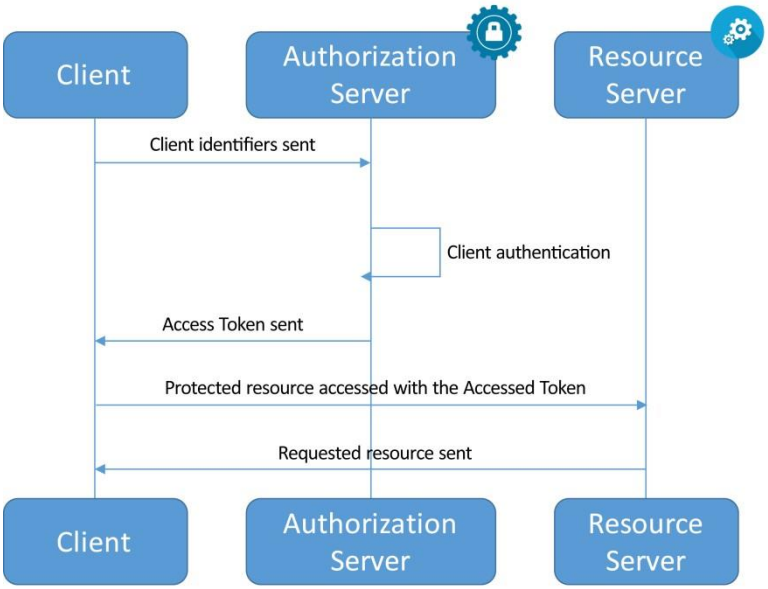
4 roles are distinguished in OAuth scenarios:

- **Resource owner**: the end user
- **Client**: Application Server requesting access to a protected resource on the Resource Server (the client can be a server-side Java application, a client-side JavaScript application or a native mobile application, for example).
- **Resource Server**: Server on which the protected data is hosted
- **Authorisation Server**: Token-issuing server. These (access) tokens will be used when the client sends requests to the resource server

OAuth defines 4 possible scenarios for granting an access token to a client:

- Authorization Code
- Resource Owner Password Credentials
- Client Credentials
- Implicit Grant

The scenario used by the APIs is "Client Credentials":



2.1.1 Access token request from the authorisation server

The access token is requested by the client application directly from the authorisation server.

<table>
<tr><td>Endpoint</td><td colspan="4">digital.iservices.rte-france.com/token/oauth/</td></tr>
<tr><td>Method</td><td colspan="4">HTTP Method: POST
Content-Type: application/x-www-form-urlencoded</td></tr>
<tr><td rowspan="2">Parameters</td><td>Name</td><td>Type</td><td>Value</td><td>Cardinality</td></tr>
<tr><td>Authorization</td><td>HEADER</td><td>“Basic”(1) followed by the client_id:client_secret pair encoded in base64 (these elements – client_id and client_secret – are provided after the API has subscribed to a generated application in RTE’s API portal)</td><td>Required</td></tr>
<tr><td>Example</td><td colspan="4">POST https://digital.iservices.rte-france.com/token/oauth/ HTTP/1.1
Authorization: Basic ZjdiZmExZWQtNmY0Py00YThkLTkxZjMtNGQ4NDczYmUwMTZjOjNhMkEyOTEyLWVlNTYtNDI1Zi1lMzZiLWI5NDY3OTFjMzQzYg==
Content-Type: application/x-www-form-urlencoded</td></tr>
</table>

The **grant_type** and **scope** attributes are not given in the token call request.

The **scope** is automatically managed based on the user profile (public user, partner user)

The **grant_type** is determined by the **application type** (**Web/Server** or **Mobile**) generated from the DATA portal and including the connection information (ClientID/ClientSecret), used to invoke the API.

(1) Note: the content of the “Authorization” header is case-sensitive. The case defined should therefore be adhered to (in the "Basic" example, the word should begin with a “capital B”).

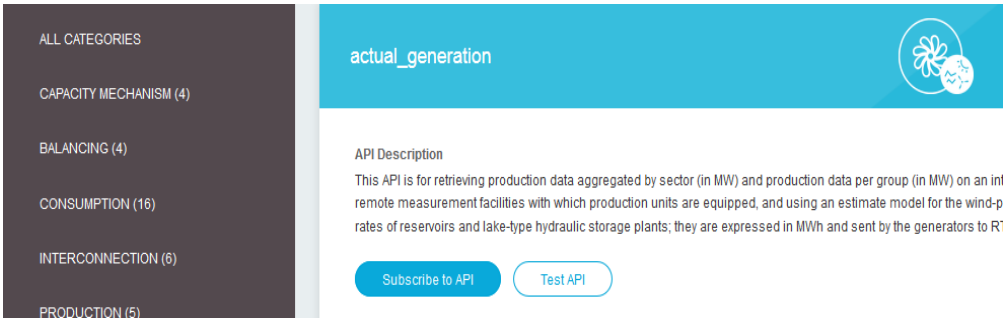
When an application is created within the portal, the user is prompted to select its type (**Web/Server** or **Mobile**).
Each application type corresponds to an Oauth2 flow

Currently, only the **"client_credentials"** flow is available via the "**Web/Server" application type**.

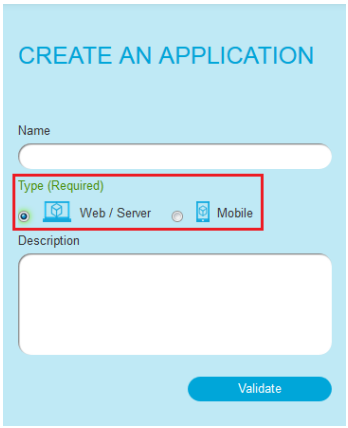
The **Mobile type** is currently also associated with the "**client_credentials"** Oauth2 flow, but may later be switched to another flow ("Resource Owner Password Credentials", for example).

It is **therefore advisable to create "Web/Server" type applications** when subscribing to an API.

To retrieve your **« client_id »** and **« client_secret »** IDs, you must first subscribe to the desired API by clicking on the button below.



Then create an application by selecting the **« Web / Server »** type. Once a first application is created, this one can be used to access different APIs.



The IDs are accessible in the "MY APPLICATIONS" menu, by clicking on the application previously created

If security problems occurs, IDs can be reset.

Several applications can be created, each one with its proper IDs.

2.1.2 Access token response from the authorisation server

The access token is sent back in JSON format in the body of the HTTP response.

<table>
<tr><td>General</td><td colspan="3">Content-Type: application/json; charset=UTF-8</td></tr>
<tr><td rowspan="5">Parameters</td><td>Name</td><td>Content</td><td>Cardinality</td></tr>
<tr><td>Access_token</td><td>The access token.</td><td>Required</td></tr>
<tr><td>Token_type</td><td>E.g.: bearer</td><td>Required</td></tr>
<tr><td>Expires_in</td><td>Number of seconds before the access token expires</td><td>Required</td></tr>
<tr><td>scope</td><td>List of API scopes for which the access token is valid.</td><td>Optional</td></tr>
<tr><td>Example</td><td colspan="3">
<pre>
HTTP/1.1 200 OK
Date: Thu, 24 Dec 2015 14:26:59 GMT
X-CorrelationID: Id-b3007c56350554276222509d 0
Cache-Control: no-store
Content-Type: application/json
Pragma: no-cache
{
      "access_token":"kZBwyADEDgjYw4rADIWA0rPOtc9ULQ7FHdQZ
      2yWz9vxWseaihQU0IL",
      "token_type":"Bearer",
      "expires_in":7200,
}
</pre>
</td></tr>
</table>

2.1.3 Invoking the resource:

Access to the resource is requested directly by the client application from the resource server with the retrieved token.

In this section, only the http HEADERs needed for the OAUTH call are shown.

<table>
<tr><td>Endpoint</td><td colspan="4">digital.iservices.rte-france.com/[service endpoint]</td></tr>
<tr><td>Method</td><td colspan="4">HTTP Method: POST
Content-Type: application/soap+xml;charset=UTF-8;</td></tr>
<tr><td rowspan="2">Parameters</td><td>Name</td><td>Type</td><td>Value</td><td>Cardinality</td></tr>
<tr><td>Authorization</td><td>HEADER</td><td>"Bearer" followed by the token supplied when the authorisation server is called</td><td>Required</td></tr>
<tr><td>Example</td><td colspan="4">
<pre>
POST /privateapi/sandbox/getDonneesPhysiques/V1 HTTP/1.1
Content-Type: application/soap+xml;charset=UTF-8;
action="getDonneesPhysiquesAction"
Authorization: Bearer MUpX2RtZq5Gj1yAX1fhdsZ3qDr5dKhdVlfvpS
Host: 10.132.7.158:8066
Content-Length: 1103

<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
<soap:Body>
….
….
</soap:Body>
</soap:Envelope>
</pre>
</td></tr>
</table>

2.1.4 Possible errors

The following types of error can be encountered:

Possible error codes	Description
invalid_request	The request is missing a required parameter, includes an invalid parameter value, is malformed, etc.
invalid_client	Client application authentication failed
invalid_grant	The grant_type is invalid, the authorisation code or refresh_token is invalid or has expired, or the redirect_uri is invalid
unauthorized_client	The client application is not authorised to request an authorisation code
unsupported_grant_type	The grant_type entered is not supported
invalid_token	Bad token or expired token provided when calling the resource
invalid_scope	The scope is invalid, unknown or malformed

Examples (No Authorisation HTTP header in the call to the resource)	`HTTP/1.1 401 Unauthorized`
	`Date: Tue, 06 Oct 2015 19:49:28 GMT`
	`Allow: OPTIONS, POST`
	`Server:`
	`Content-Length: 0`
	`Connection: close`
	`X-CorrelationID: Id-c8251456c9b142fd925aeb6b 0`
	`WWW-Authenticate: Bearer realm="DefaultRealm"`

Examples (In the "Authorisation" HTTP header, there is no "bearer" in the call to the resource)	`HTTP/1.1 400 Bad Request`
	`Date: Tue, 06 Oct 2015 19:52:59 GMT`
	`Allow: OPTIONS, POST`
	`Server:`
	`Content-Length: 0`
	`Connection: close`
	`X-CorrelationID: Id-9b26145635b274c6337385f7 0`
	`WWW-Authenticate: Bearer realm="DefaultRealm",error="invalid_request",error_description="Problem parsing the Authorization Header`

Examples (Bad token or expired token provided when calling the resource)	`HTTP/1.1 401 Unauthorized`
	`Date: Tue, 06 Oct 2015 19:51:11 GMT`
	`Allow: OPTIONS, POST`
	`Server:`
	`Content-Length: 0`
	`Connection: close`
	`X-CorrelationID: Id-2f261456feb15af315740ec1 0`
	`WWW-Authenticate: Bearer realm="DefaultRealm",error="invalid_token",error_description="Unable to find the access token in persistent storage."`

Examples (No grant_type=client_credentials definition when calling the resource)	`HTTP/1.1 401 Unauthorized`
	`Date: Tue, 06 Oct 2015 20:10:16 GMT`
	`Server:`
	`Content-Encoding: gzip`
	`Connection: close`
	`X-CorrelationID: Id-a82a145637b3794468a87abd 0`

	Authorization: Basic ZWI4ODY3NTItNjJiZi00YjliLWExMjYtN2JhNTYyMzkyNjY3OjJmZDljNjFhLWFjZWItNGEyYi1hODk5LTg2OTI2OWMwMmQzMg== Host: 10.132.7.157:8089 User-Agent: Jakarta Commons-HttpClient/3.1 Content-Type: text/html <!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN"> <HTML><HEAD> <TITLE>401 Authorization Required</TITLE> </HEAD><BODY> <H1>Authorization Required</H1> This server could not verify that you are authorized to access the document requested. Either you supplied the wrong credentials (e.g., bad password), or your browser doesn't understand how to supply the credentials required.<P> <HR> </BODY></HTML>

Examples
(Bad Authorisation (ClientID:ClientSecret), No basic or no Authorisation header defined when calling the resource)
	HTTP/1.1 401 Unauthorized
	Date: Tue, 06 Oct 2015 20:13:34 GMT
	Server:
	Content-Encoding: gzip
	Connection: close
	X-CorrelationID: Id-6e2b145669b322f541bb840c 0
	WWW-Authenticate: Basic realm="DefaultRealm"
	Content-Type: application/json
	{
	"error" : "invalid_client",
	"error_description" : "Client authentication failed (e.g. unknown client, no client authentication included, or unsupported authentication method). The authorization server MAY return an HTTP 401 (Unauthorized) status code to indicate which HTTP authentication schemes are supported. "
	}

~°~°~°~°~°~°~°~°~° End of document ~°~°~°~°~°~°~°~°~°~°