API

ProjeQtOr provides an API to interact with its elements. It is provided as REST Web Service.
It is possible to read (method GET), create (methods PUT, POST), update (methods PUT, POST) and delete (method DELETE) elements.

First, the API must be enabled : for security reasons it is not enabled by default.

- Generate a **.htpasswd** file (see related topics on the net on how to do this)
  A template is provided in /api/.htpasswd, referring to user projeqtor, password projeqtor.
  It is provided only for test purpose.
  Do not use it on a production environment as it would expose all your data.
- Update .htaccess file to specify location of your .htpasswd file :
  AuthUserFile "/pathToFile/.htpasswd"
  Default location is Apache directory.

Use of API :

- Since V4.4, user used (defined in .htpassword) must exist as a User in the database.
  Then, access rights (read, create, update, delete) to the access defined for this user.
  This allows you to provide some access to external users and control the visibility they get on your data
- Available methods are GET (read), PUT (create, update), POST (create, update) and DELETE (delete)
- For PUT, PUSH and DELETE methods, data must be encrypted with AES-256 algorythm, with key as the API key defined for the user. Admin must provide this API Key to the API consumer.
  You can use AESCRT library provided in /external directory for the encryption.
- PUT and PUSH methods are similar and can both be used to create or update elements.
  Difference is only in the way to send data : as a Post array for POST, as a file for PUT.
- DELETE method requires data, formatted as for a PUT, but only "id" is required.
- For PUT, POST and DELETE, you can provide :
  - a single item : `{"id":"1", …}`
  - a list of items : `{"identified":"id", "items":[{"id":"1", …}, {"id":"2", …}]}`
- Json format retrieved from GET can be used for PUT, POST and DELETE.

API

Method	Url	Result
GET	http://myserver/api/{object class}/{object id} Ex : *http://myserver/api/Project/1*	The complete description, in Json format, of the object of the given class with the given id
GET	http://myserver/api/{object class}/all Ex : *http://myserver/api/Project/all*	The complete description, in Json format, of all the objects of the given class
GET	http://myserver/api/{object class}/filter/{filter id} Ex : *http://myserver/api/Project/filter/1*	The complete description, in Json format, of all the objects of the given class corresponding to the given stored filter • id of the filter can be retrieved when saving filter • Using filter of a different class may lead to unexpected results.
GET	http://myserver/api/{object class}/search/{crit1}/{critN} Ex : *http://myserver/api/Activity/search/idProject=1/name like '%error%'*	The complete description, in Json format, of all the objects of the given class corresponding to the given criteria. • you can provide as many criteria as you wish, they will be included in where clause with AND operator. • unlike example, criteria must be "url encoded" (use PHP urlencode() function for instance)
GET	http://myserver/api/{object class}/updated/{start date}/{end date} Ex : *http://myserver/api/Project/updated/20130101000000/20131231235959*	The complete description, in Json format, of all the objects updated between start date en end date • Date format is YYYYMMDDHHMNSS • Date >= "start date" and Date < "end date"
POST	http://myserver/api/{object class} data provided in json format as POST value	The complete description, in Json format, of updated or created objects, with 2 additional fields : •apiResult : status of update •apiResultMessage : detailed message
PUT	http://myserver/api/{object class} data provided in json format as a file	
DELETE	http://myserver/api/{object class} data provided in json format as a file	

API

Here is an example of PHP code calling the API for **GET** request (read) :

```
$fullUrl="http://myserver/api/Ticket/list/all";
$curl = curl_init($fullUrl);
curl_setopt($curl, CURLOPT_HTTPAUTH, CURLAUTH_BASIC);
curl_setopt($curl, CURLOPT_USERPWD, "projeqtor:projeqtor");
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);
$curl_response = curl_exec($curl);
echo $curl_response;
curl_close($curl);
```

This request lists all Tickets

Here is an example of PHP code calling the API for **DELETE** request (create, update):

```
$fullUrl="http://myserver/api/Ticket";
$data='{"id":"1"}';
$data=AesCtr::encrypt($data, 'ApiKeyForUserProjeqtor', 256);
$curl = curl_init($fullUrl);
curl_setopt($curl, CURLOPT_HTTPAUTH, CURLAUTH_BASIC);
curl_setopt($curl, CURLOPT_USERPWD, "projeqtor:projeqtor");
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);
curl_setopt($curl, CURLOPT_CUSTOMREQUEST, "DELETE");
curl_setopt($curl, CURLOPT_POST, true);
curl_setopt($curl, CURLOPT_POSTFIELDS, $data);
$curl_response = curl_exec($curl);
echo $curl_response;
curl_close($curl);
```

This request deletes Ticket #1

API

Here is an example of PHP code calling the API for **PUT** and **POST** request (create, update):

```
$fullUrl="http://myserver/api/Ticket";
$data='{"id":"1", "name":"name to be changed for Ticket 1"}';
$data=AesCtr::encrypt($data, 'ApiKeyForUserProjeqtor', 256);
$curl = curl_init($fullUrl);
curl_setopt($curl, CURLOPT_HTTPAUTH, CURLAUTH_BASIC);
curl_setopt($curl, CURLOPT_USERPWD, "projeqtor:projeqtor");
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);
curl_setopt($curl, CURLOPT_CUSTOMREQUEST, "PUT");
curl_setopt($curl, CURLOPT_POST, true);
curl_setopt($curl, CURLOPT_POSTFIELDS, $data);
$curl_response = curl_exec($curl);
echo $curl_response;
curl_close($curl);
```

```
$fullUrl="http://myserver/api/Ticket";
$data='{"id":"1", "name":"name to be changed for Ticket 1"}';
$data=AesCtr::encrypt($data, 'ApiKeyForUserProjeqtor', 256);
$curl = curl_init($fullUrl);
curl_setopt($curl, CURLOPT_HTTPAUTH, CURLAUTH_BASIC);
curl_setopt($curl, CURLOPT_USERPWD, "projeqtor:projeqtor");
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);
curl_setopt($curl, CURLOPT_CUSTOMREQUEST, "POST");
curl_setopt($curl, CURLOPT_POST, true);
curl_setopt($curl, CURLOPT_POSTFIELDS, array('data'=>$data));
$curl_response = curl_exec($curl);
echo $curl_response;
curl_close($curl);
```

These requests update name of Ticket #1