

B/ Les points intéressants

L'API permet de CRUD (**Create, Read, Update, Delete**) les ressources. Vous trouverez ci-dessous une liste des ressources concernées.

- Documents (avec le path ou par id)
- Users
- Groups
- Directories
- Automation (voir plus bas, Nuxeo Content Automation)
- Document Types & Schemas
- Workflow & Task (reste encore à implémenter)

C/ Comment utiliser l'API REST ?

Nous avons donc essayé d'utiliser cette API pour faire de la recherche de document dans le dépôt Nuxeo. L'application que nous avons réalisée est codée en PHP et est hébergée en local sur une de nos machines. Notons que le SDK pour PHP est encore en cours d'implémentation mais il propose déjà les opérations principales. Nous allons donc montrer comment invoquer des opérations à distances avec le client PHP. Avant cela il est nécessaire de dire quelques mots sur Nuxeo Content Automation.

Le Nuxeo Content Automation est un service qui implémente des actions exécutables dans la plateforme. Ces actions sont des opérations dites atomiques. Il est également possible d'assembler ces opérations en chaine pour ainsi créer des règles complexes (rendre un document obsolète au bout d'un an, déplacer un document après validation, ...). On peut donc utiliser ce service directement sur le serveur dans l'application Nuxeo en Java ou alors à distance au travers de l'API REST. Elle met aussi à disposition des développeurs (avec Nuxeo IDE) les moyens d'écrire de nouvelles opérations en Java.

Voyons comment utiliser l'API avec le SDK PHP :

```

1  <?php
2  require_once('./nuxeo-automation-php-client-master/NuxeoAutomationClient/NuxeoAutomationAPI.php');
3
4  // $client = new NuxeoPhpAutomationClient("http://pfe-nuxeo.univ-avignon.fr:8080/nuxeo/site/automation");
5  $client = new NuxeoPhpAutomationClient("http://pfe-nuxeo.univ-avignon.fr:8080/nuxeo/site/automation");
6
7  $session = NULL;
8  if (isset($_REQUEST['name']) AND isset($_REQUEST['word']) AND ($_REQUEST['name'] != NULL AND $_REQUEST['word'] != NULL)) {
9      $session = $client->getSession($_REQUEST['name'], $_REQUEST['word']);
10 }
11
12 $session = $client->getSession("Administrator", "Administrator");
13
14 ?>
15

```

La classe `NuxeoPhpAutomationClient` permet de récupérer une instance de la session (`NuxeoSession`) avec `getSession`.

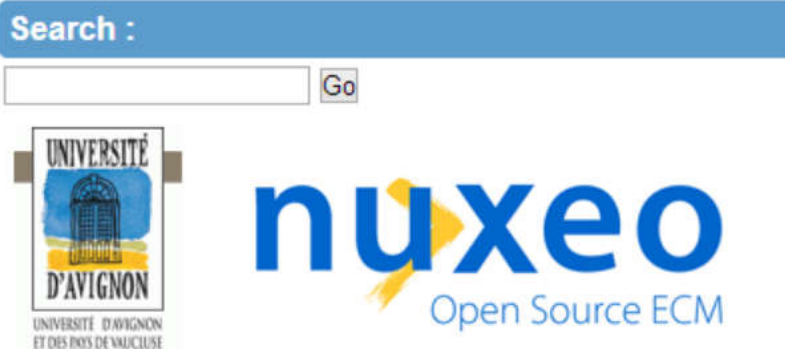
```

47 <?php
48 $search = $_REQUEST['search'];
49 $answer = $session->newRequest("Document.Query")->set('params', 'query', "SELECT * FROM
50 Document WHERE ecm:name = '". $search. "'")->sendRequest();
51
52 $documentsArray = $answer->getDocumentList();

```

A partir de la session, il est possible de créer une requête. La méthode `set` sert à configurer la requête. Une fois qu'elle est configurée, on soumet la requête avec la méthode `sendRequest`.

Avec quelques images, quelques balises HTML et un peu de CSS nous obtenons un petit module de recherche de documents dans le dépôt Nuxeo. Nous l'avons imaginé de manière à ce qu'il puisse bien s'intégrer en tant que « **widget** » sur la plate-forme de l'Université, par exemple sur l'ENT.



Ce petit client PHP n'a donc pas d'autre but que de montrer qu'il est possible d'accéder aux ressources Nuxeo facilement et proprement.

Nous effectuons une recherche pour le document « Administrator » (tout est document) sur le client PHP.

Search :

Administrator

Go



Result :

Path	Type	State	Title
/default-domain/UserWorkspaces/Administrator Workspace	project	Administrator	

Nous voyons s'afficher les caractéristiques du Document « Administrator ».

D/ Gestion de l'authentification :

Pour pouvoir envoyer des requêtes à Nuxeo au travers de l'API, il est nécessaire d'être authentifié, plus exactement, la requête. Nuxeo dispose d'un protocole apparemment « fait maison » basé sur des Tokens. Ce protocole permet entre autre à l'application cliente d'authentifier les requêtes sans devoir retenir les informations telles que le login et le mot de passe, ce qui pourrait constituer une faille de sécurité. Un Token relie un nom d'utilisateur, le nom d'une application et l'identifiant d'un appareil ou d'un dispositif. L'API REST permet également d'utiliser le protocole d'authentification OAuth2 qui a un principe de fonctionnement basé sur plusieurs Tokens. Il est également possible d'utiliser une authentification basique avec tous les problèmes de sécurité que cela peut entraîner.

E/ Retour d'expérience

L'API REST fournit un bon moyen de réaliser des applications externes à la plate-forme. De plus le « REST style » est particulièrement apprécié par les développeurs. Une fois que l'on a assimilé le fonctionnement de l'API, il devient assez simple de réaliser de petites applications dédiées à une tâche particulière ou plusieurs. Grâce à l'API, nous pouvons nous passer (non pas sans efforts de compréhension) de la solution Studio et ainsi étendre la plate-forme Nuxeo.