BEE-SMART

Documenttion

Alfresco

DOC-BEE-014

GESTION DES EVOLUTIONS

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Nature** | **Auteur** |
| 1.0 | 15/05/2023 | Création | D.RAMDANI |

**Table des matières**

[**1. ALFRESCO 3**](#_Toc135142216)

[**2. INTRODUCTION 4**](#_Toc135142217)

[**3. LES FONCTIONNALITÉS 5**](#_Toc135142218)

[**4. LES AVANTAGES ET INCONVÉNIENTS 6**](#_Toc135142219)

[** LES AVANTAGES: 6**](#_Toc135142220)

[** LES INCONVÉNIENTS: 6**](#_Toc135142221)

[**5. LES CAS D’UTILISATION 6**](#_Toc135142222)

[**6. INSTALLATION 7**](#_Toc135142223)

[**7. UTILISATION DE L’API 8**](#_Toc135142224)

# **ALFRESCO**

Une GED (Gestion Électronique de Documents) est une solution informatique qui permet de stocker, organiser, partager et gérer l'ensemble des documents d'une entreprise de manière électronique. Elle permet ainsi de remplacer la gestion papier souvent fastidieuse et coûteuse, par une gestion numérique plus efficace et économique. Les documents peuvent être facilement accessibles et partagés entre les différents services de l'entreprise, tout en garantissant leur sécurité et leur confidentialité. C'est dans ce contexte que s'inscrit Alfresco, une plateforme de gestion de contenu d'entreprise open source, permettant de répondre aux besoins de la GED.

# **INTRODUCTION**

Alfresco est une plateforme de gestion de contenu d'entreprise (ECM) open source qui permet aux entreprises de stocker, de gérer et de partager efficacement leurs informations. Elle offre une large gamme de fonctionnalités pour gérer les documents, les dossiers, les workflows et les tâches, ainsi que pour collaborer et partager des contenus en toute sécurité. Avec Alfresco, les entreprises peuvent améliorer leur productivité, leur efficacité et leur conformité réglementaire en facilitant la gestion et la diffusion de leurs contenus numériques. Dans ce travail, nous explorerons les fonctionnalités d'Alfresco, son utilisation et ses avantages pour les entreprises.

# **LES FONCTIONNALITÉS**

Alfresco est une plateforme de gestion de contenu d'entreprise (ECM) open source qui offre une variété de fonctionnalités pour aider les entreprises à gérer efficacement leurs contenus numériques. Voici quelques-unes des fonctionnalités principales proposées par Alfresco :

* Gestion de documents : Alfresco permet de stocker et d'organiser tous les types de documents numériques, tels que des fichiers texte, des images, des vidéos, des présentations et des feuilles de calcul.
* Gestion de processus métier : Alfresco offre des outils pour créer, exécuter et surveiller des processus métier automatisés, ce qui permet aux entreprises de gérer leurs workflows plus efficacement.
* Collaboration : Les fonctionnalités de collaboration d'Alfresco permettent aux utilisateurs de travailler ensemble sur des documents en temps réel, de partager des commentaires et des annotations, et de collaborer de manière transparente avec des partenaires externes.
* Recherche et récupération : Alfresco offre des outils de recherche avancée qui permettent aux utilisateurs de trouver rapidement et facilement les documents qu'ils recherchent.
* Sécurité : La plateforme Alfresco offre des fonctionnalités de sécurité robustes, notamment la gestion des droits d'accès, la sécurité du contenu et la protection contre les menaces de sécurité.

En somme, Alfresco est une plateforme ECM complète qui offre une variété de fonctionnalités pour aider les entreprises à gérer efficacement leur contenu numérique.

# **LES AVANTAGES ET INCONVÉNIENTS**

## **LES AVANTAGES:**

* Alfresco est open-source, ce qui signifie qu'il est gratuit à utiliser et peut être personnalisé pour répondre aux besoins spécifiques de chaque entreprise.
* Il offre des fonctionnalités de gestion de contenu robustes telles que la gestion des versions, la recherche avancée et la collaboration en temps réel.
* Alfresco est facile à utiliser grâce à une interface utilisateur intuitive.
* Il est compatible avec de nombreux systèmes d'exploitation et de bases de données, ce qui le rend facile à intégrer dans un environnement existant.

## **LES INCONVÉNIENTS:**

* La mise en place d'Alfresco peut être complexe, surtout pour les entreprises qui n'ont pas de personnel technique expérimenté.
* La documentation d'Alfresco peut être difficile à comprendre pour les utilisateurs inexpérimentés.
* Les mises à jour régulières peuvent être fastidieuses à installer et peuvent entraîner des problèmes de compatibilité avec les versions précédentes.
* Alfresco peut être gourmand en ressources système, ce qui peut entraîner des ralentissements et des temps d'arrêt.

# **LES CAS D’UTILISATION**

* Gestion documentaire: Alfresco permet de stocker, organiser et gérer les documents de manière centralisée. Les utilisateurs peuvent accéder aux documents en ligne, les partager avec d'autres utilisateurs, les rechercher et les récupérer en cas de besoin.
* Collaboration d'équipe: Alfresco offre des fonctionnalités de collaboration telles que des espaces de travail d'équipe, des forums de discussion, des blogs et des wikis. Les utilisateurs peuvent collaborer sur des projets en temps réel et partager des informations en toute sécurité.
* Gestion de contenu Web: Alfresco peut être utilisé pour créer et gérer des sites Web. Il permet de créer des pages Web, des formulaires en ligne, des blogs et des forums de discussion. Il offre également des fonctionnalités de publication de contenu pour les sites Web.
* Gestion de dossiers de santé électroniques: Alfresco peut être utilisé pour gérer les dossiers de santé électroniques. Il permet de stocker et de gérer les dossiers de santé électroniques en toute sécurité, tout en permettant un accès facile et rapide aux informations de santé.
* Gestion de projet: Alfresco peut être utilisé pour gérer les projets. Il permet de stocker les fichiers de projet, de suivre les tâches et les calendriers, de gérer les budgets et de collaborer en temps réel avec les membres de l'équipe.

En résumé, Alfresco est un outil de GED complet et polyvalent qui peut être utilisé dans de nombreux cas d'utilisation différents. Que ce soit pour gérer des documents, collaborer en équipe ou gérer des projets, Alfresco offre des fonctionnalités robustes pour aider les organisations à atteindre leurs objectifs.

# **INSTALLATION**

Lors de mes recherches pour installer et utiliser Alfresco, j'ai rencontré de nombreux défis et difficultés. J'ai découvert qu'il était fortement recommandé d'utiliser Docker pour déployer Alfresco. Docker est une plateforme open-source qui permet de développer, déployer et exécuter des applications dans des conteneurs. Les conteneurs sont des environnements d'exécution légers et portables qui contiennent tout le nécessaire pour exécuter une application, y compris le code, les bibliothèques et les dépendances.

Bien que l'utilisation de Docker présente plusieurs avantages pour l'exécution d'Alfresco, notamment en termes de portabilité et de cohérence des versions, j'ai décidé de ne pas utiliser cette méthode en raison des difficultés rencontrées. Au lieu de cela, j'ai choisi d'utiliser la version gratuite en ligne de 14 jours d'Alfresco, qui m'a permis d'accéder aux fonctionnalités d'Alfresco sans avoir à me soucier de l'installation et de la configuration complexes liées à Docker.

Cette approche a simplifié le processus et m'a permis de me concentrer sur l'utilisation de l'API Alfresco pour implémenter les fonctionnalités souhaitées dans mon projet, sans les tracas liés à la gestion des conteneurs Docker et à la résolution des problèmes de compatibilité.

# **UTILISATION DE L’API**

**Une image contenant cercle, Graphique, symbole, conception

Description générée automatiquement**PRÉSENTATION

L'API Alfresco est une interface de programmation d'application (API) qui permet aux développeurs d'intégrer les fonctionnalités d'Alfresco dans leurs propres applications et systèmes. Alfresco est un système de gestion de contenu d'entreprise (ECM) open-source qui aide les organisations à gérer leurs documents, dossiers et autres contenus numériques. Il offre des fonctionnalités telles que la gestion des versions, la collaboration, la recherche, la gestion des workflows et bien d'autres.

L'API Alfresco est basée sur l'architecture REST (Representational State Transfer). Les API REST sont des interfaces de programmation qui utilisent les protocoles et les conventions du Web, tels que HTTP et JSON, pour permettre la communication entre des applications et des services sur Internet. L'utilisation d'API REST présente plusieurs avantages, tels que la simplicité, la flexibilité, la facilité d'intégration et la scalabilité.

Avec l'API REST Alfresco, les développeurs peuvent effectuer des opérations telles que la création, la récupération, la mise à jour et la suppression de documents, dossiers, utilisateurs et groupes, ainsi que la gestion des workflows et des permissions. L'API permet également d'interagir avec les fonctionnalités spécifiques d'Alfresco, telles que les aspects et les métadonnées. L'API REST Alfresco est bien documentée et dispose d'une communauté active de développeurs qui contribuent à son amélioration et à son support.

Une image contenant Graphique, cercle, symbole, conception

Description générée automatiquementLE BUT DU PROJET

L'API Alfresco a été intégrée au projet pour faciliter la communication entre l'interface utilisateur et le serveur Alfresco. En utilisant l'API RESTful, l'application peut effectuer des opérations telles que la création de fichiers, la mise à jour des métadonnées et la suppression de fichiers sur le serveur Alfresco. L'interface utilisateur met à jour les informations affichées en temps réel en fonction des actions effectuées par l'utilisateur.

Lorsqu'un utilisateur effectue une action sur un fichier, comme le téléchargement, le renommage ou la suppression, l'application envoie une requête à l'API Alfresco avec les informations appropriées. Ensuite, l'API Alfresco effectue l'action demandée et renvoie une réponse à l'application. Si l'action a réussi, l'interface utilisateur est mise à jour pour refléter les modifications. Les modifications sont également reflétées sur le serveur Alfresco, assurant ainsi une synchronisation entre l'interface utilisateur et le serveur.

Une image contenant cercle, Graphique, symbole, Police

Description générée automatiquementLES DÉFIS RENCONTRÉS

Lors de la réalisation de ce projet, plusieurs défis ont été rencontrés, notamment en ce qui concerne l'intégration de l'API Alfresco et la compréhension de son fonctionnement. Voici quelques-uns de ces défis :

* Intégration de l'API Alfresco : La documentation de l'API Alfresco est vaste et couvre de nombreux aspects. Il a été difficile de comprendre comment intégrer l'API et de déterminer les méthodes API spécifiques à utiliser pour répondre aux besoins du projet.
* Compréhension du fonctionnement de l'API : L'API Alfresco utilise plusieurs concepts clés, tels que les nœuds et les enfants, qui sont essentiels pour interagir avec les documents et les dossiers. Il a été difficile de comprendre ces concepts et de savoir comment les appliquer dans le contexte du projet.
* Notion de nœuds et d'enfants : Les nœuds sont des éléments de base dans Alfresco, représentant des documents, des dossiers ou d'autres types de contenu. Les enfants sont des nœuds qui sont contenus dans un autre nœud, généralement un dossier. Il a été difficile de comprendre la relation entre les nœuds et les enfants et de savoir comment utiliser les méthodes API appropriées pour gérer ces éléments dans le système Alfresco.

Malgré ces défis, une solution a été trouvée pour intégrer l'API Alfresco et créer une interface fonctionnelle permettant de gérer les fichiers. Tout au long du projet, une meilleure compréhension de l'API et de ses concepts clés a été acquise, ce qui a permis de résoudre les problèmes et de progresser dans la réalisation du projet.

1. MANUEL D’UTILISATION

Dans cette section, nous allons explorer en profondeur l'interface de notre application. Cette exploration vous permettra de comprendre comment chaque élément interagit avec les autres, comment naviguer efficacement à travers les diverses fonctionnalités et comment utiliser chaque outil à votre disposition. Que vous soyez un débutant ou un utilisateur expérimenté, cette section vous offrira des éclairages précieux pour maximiser votre expérience utilisateur et tirer le meilleur parti de notre application. Suivez-nous dans cette aventure pour devenir un véritable expert de notre interface !

**Une image contenant Graphique, créativité, astronomie, art

Description générée automatiquementUne image contenant Graphique, créativité, astronomie, art

Description générée automatiquementUne image contenant Graphique, créativité, astronomie, art

Description générée automatiquement**Une image contenant texte, capture d’écran, logiciel, Icône d’ordinateur

Description générée automatiquement

Dans ce manuel, nous allons explorer une interface puissante et intuitive conçue pour faciliter la gestion de vos fichiers. L'interface fournit une variété de fonctions de base pour organiser, gérer et interagir avec les documents de manière efficace et transparente.

Tout d'abord, vous pouvez facilement télécharger des fichiers à l'aide d'un formulaire simple et clair. En un seul clic, vous pouvez sélectionner les fichiers à télécharger et les envoyer à notre système.

Mais cette interface ne se limite pas à l'envoi de fichiers. Après avoir téléchargé vos fichiers, vous pouvez afficher une liste ordonnée de tous vos fichiers. Chaque entrée de cette liste offre une gamme d'options d'interaction : vous pouvez télécharger le fichier sur votre appareil, le renommer pour mieux l'organiser ou le supprimer entièrement lorsque vous n'en avez plus besoin.

Afin de mieux pouvoir le code, nous allons analyser chaque fichier tout en expliquant ce qu’il fait.

* Index.php :
* <?php  
  // Importe le fichier 'functions.php' qui contient les fonctions pour interagir avec l'API Alfresco  
  require\_once 'functions.php';  
  ?>  
    
  <!DOCTYPE html>  
  <html lang="en">  
  <head>  
   <meta charset="UTF-8">  
   <meta name="viewport" content="width=device-width, initial-scale=1.0">  
   <title>Alfresco File Upload</title>  
   <link rel="stylesheet" href="styles.css"> <!-- Lien vers la feuille de style CSS -->  
  </head>  
  <body>  
  <div class="container">  
   <h1>Upload a File to Alfresco</h1>  
   <!-- Formulaire pour sélectionner et télécharger un fichier -->  
   <form onsubmit="***event***.preventDefault(); uploadFile();">  
   <label for="file">Choisir le fich:</label>  
   <input type="file" name="file" id="file" required>  
   <br><br>  
   <input type="submit" value="Upload">  
   </form>  
    
   <h2>Files in Alfresco</h2>  
   <!-- Zone d'affichage de la liste des fichiers -->  
   <div id="file-list">  
   <?php  
   // Récupère la liste des fichiers à l'aide de la fonction 'getFiles()'  
   $files = getFiles();  
   if (isset($files['error'])) {  
   echo "<span class='error'>Error: {$files['error']}</span>";  
   } else {  
   echo '<ul>';  
   // Affiche chaque fichier et les boutons d'actions associés  
   foreach ($files as $entry) {  
   $file = $entry['entry'];  
   echo "<li>";  
   echo "<span class='filename'>{$file['name']}</span>";  
   // Bouton pour télécharger le fichier  
   echo '<button onclick="downloadFile(\'' . htmlspecialchars($file['id']) . '\', \'' . htmlspecialchars($file['name']) . '\')">Télécharger</button>';  
   // Bouton pour renommer le fichier  
   echo "<button onclick=\"renameFile('{$file['id']}', '{$file['name']}')\">Rename</button>";  
   // Bouton pour supprimer le fichier  
   echo "<button onclick=\"deleteFile('{$file['id']}')\">Delete</button>";  
   echo "</li>";  
   }  
   echo '</ul>';  
   }  
   ?>  
   </div>  
  </div>  
  <!-- Importe le fichier 'scripts.js' contenant les fonctions JavaScript utilisées -->  
  <script src="scripts.js"></script>  
  </body>  
  </html>

Le fichier index.php est le point d'entrée principal de cette application web. Il fournit une interface utilisateur pour interagir avec le système de gestion des fichiers Alfresco. Voici une vue d'ensemble de ses fonctionnalités :

* Téléchargement de fichiers : L'interface offre une fonctionnalité de téléchargement de fichiers. Les utilisateurs peuvent choisir un fichier depuis leur système local et le télécharger sur le serveur Alfresco.
* Liste des fichiers : Le fichier index.php affiche également la liste des fichiers actuellement stockés sur le serveur Alfresco. Pour chaque fichier, il affiche son nom.
* Gestion des fichiers : Pour chaque fichier listé, l'utilisateur a plusieurs options. Il peut télécharger le fichier sur son système local, renommer le fichier sur le serveur Alfresco, ou supprimer le fichier du serveur.
* Interactions avec l'API Alfresco : Toutes ces actions (téléchargement de fichiers, renommage, suppression) sont effectuées en interagissant avec l'API Alfresco, grâce à des fonctions définies dans le fichier functions.php.
* Le fichier download.php :

<?php  
require\_once 'config.php';  
  
header("Content-Type: application/octet-stream");  
  
$nodeId = $\_GET['nodeId'] ?? null;  
$fileName = $\_GET['fileName'] ?? null;  
  
if ($nodeId && $fileName) {  
 $url = "{$apiUrl}/nodes/{$nodeId}/content?attachment=true";  
  
 $ch = curl\_init();  
 curl\_setopt($ch, *CURLOPT\_URL*, $url);  
 curl\_setopt($ch, *CURLOPT\_USERPWD*, "$username:$password");  
 curl\_setopt($ch, *CURLOPT\_SSL\_VERIFYPEER*, false);  
 curl\_setopt($ch, *CURLOPT\_RETURNTRANSFER*, true);  
 curl\_setopt($ch, *CURLOPT\_HEADER*, false);  
  
 $response = curl\_exec($ch);  
 $httpCode = curl\_getinfo($ch, *CURLINFO\_HTTP\_CODE*);  
  
 if ($httpCode == 200) {  
 header("Content-Disposition: attachment; filename=\"$fileName\"");  
 echo $response;  
 } else {  
 echo "Error: Failed to download the file.";  
 }  
  
 curl\_close($ch);  
} else {  
 echo "Erreur : nodeId ou fileName ne sont pas définies ";  
}  
?>

Le fichier download.php est utilisé pour télécharger les fichiers du serveur Alfresco. Il le fait en utilisant l'API Alfresco pour récupérer le contenu du fichier, puis en envoyant ce contenu au client. Voici une vue d'ensemble de ce qu'il fait :

* Importation de la configuration : require\_once 'config.php'; Ce code importe le fichier de configuration config.php, qui contient des informations cruciales telles que les identifiants de l'utilisateur et l'URL de l'API.
* Récupération des paramètres : $nodeId = $\_GET['nodeId'] ?? null; et $fileName = $\_GET['fileName'] ?? null; Ces deux lignes récupèrent les paramètres nodeId et fileName de la requête GET. Si ces paramètres ne sont pas présents, ils seront définis sur null.
* Vérification des paramètres : if ($nodeId && $fileName) { ... } Cette condition vérifie si les deux paramètres sont présents. Si c'est le cas, le script tente de télécharger le fichier. Sinon, il affiche un message d'erreur.
* Demande à l'API Alfresco : Le script crée une nouvelle requête cURL vers l'API Alfresco pour récupérer le contenu du fichier. Il utilise le nodeId pour identifier le fichier et les identifiants de l'utilisateur pour s'authentifier.
* Gestion de la réponse de l'API : Le script vérifie le code de statut HTTP de la réponse. Si le code de statut est 200 (ce qui signifie que la requête a réussi), le script envoie le contenu du fichier au client avec un en-tête Content-Disposition pour indiquer que le contenu doit être téléchargé comme un fichier. Sinon, il affiche un message d'erreur.
* Le fichier rename\_file.php :

<?php  
require\_once 'config.php';  
  
$nodeId = $\_POST['nodeId'] ?? null;  
$newName = $\_POST['newName'] ?? null;  
  
if ($nodeId && $newName) {  
 $ch = curl\_init();  
  
 curl\_setopt($ch, *CURLOPT\_URL*, "{$apiUrl}/nodes/{$nodeId}");  
 curl\_setopt($ch, *CURLOPT\_USERPWD*, "$username:$password");  
 curl\_setopt($ch, *CURLOPT\_CUSTOMREQUEST*, "PUT");  
 curl\_setopt($ch, *CURLOPT\_POSTFIELDS*, json\_encode(['name' => $newName]));  
 curl\_setopt($ch, *CURLOPT\_SSL\_VERIFYPEER*, false);  
 curl\_setopt($ch, *CURLOPT\_RETURNTRANSFER*, true);  
 curl\_setopt($ch, *CURLOPT\_HEADER*, false);  
 curl\_setopt($ch, *CURLOPT\_HTTPHEADER*, [  
 "Content-Type: application/json"  
 ]);  
  
 $response = curl\_exec($ch);  
 $httpCode = curl\_getinfo($ch, *CURLINFO\_HTTP\_CODE*);  
  
 if ($httpCode == 200) {  
 echo json\_encode(['message' => 'File renamed successfully']);  
 } else {  
 echo json\_encode(['error' => 'Failed to rename the file']);  
 }  
  
 curl\_close($ch);  
} else {  
 echo json\_encode(['error' => 'Missing nodeId or newName']);  
}  
?>

Le fichier rename\_file.php est utilisé pour renommer les fichiers sur le serveur Alfresco. Il le fait en utilisant l'API Alfresco pour modifier le nom du fichier. Voici un aperçu de son fonctionnement :

* Importation de la configuration : require\_once 'config.php'; Ce code importe le fichier de configuration config.php, qui contient des informations importantes telles que les identifiants de l'utilisateur et l'URL de l'API.
* Récupération des paramètres : $nodeId = $\_POST['nodeId'] ?? null; et $newName = $\_POST['newName'] ?? null; Ces deux lignes récupèrent les paramètres nodeId et newName de la requête POST. Si ces paramètres ne sont pas présents, ils seront définis sur null.
* Vérification des paramètres : if ($nodeId && $newName) { ... } Cette condition vérifie si les deux paramètres sont présents. Si c'est le cas, le script tente de renommer le fichier. Sinon, il renvoie un message d'erreur en format JSON.
* Demande à l'API Alfresco : Le script crée une nouvelle requête cURL vers l'API Alfresco pour modifier le nom du fichier. Il utilise le nodeId pour identifier le fichier, les identifiants de l'utilisateur pour s'authentifier, et le newName comme le nouveau nom du fichier. La requête est une requête PUT, ce qui signifie qu'elle est utilisée pour mettre à jour une ressource existante.
* Gestion de la réponse de l'API : Le script vérifie le code de statut HTTP de la réponse. Si le code de statut est 200 (ce qui signifie que la requête a réussi), le script renvoie un message de succès en format JSON. Sinon, il renvoie un message d'erreur en format JSON.
* Le fichier « functions.php »
* <?php  
  // Définition de la fonction 'getFiles()' pour récupérer la liste des fichiers stockés dans Alfresco  
  function getFiles() {  
   // URL de l'API Alfresco pour récupérer les fichiers  
   $url = "https://a3mop2z6.trials.alfresco.com/alfresco/api/-default-/public/alfresco/versions/1/nodes/-my-/children";  
   // Identifiants pour accéder à l'API Alfresco  
   $username = "ramdjibril24@gmail.com";  
   $password = "J7cQPVhXBIho";  
    
   // Initialisation de cURL pour effectuer une requête HTTP  
   $curl = curl\_init();  
   // Configuration de l'URL pour la requête  
   curl\_setopt($curl, *CURLOPT\_URL*, $url);  
   // Retourne la réponse HTTP en tant que chaîne de caractères  
   curl\_setopt($curl, *CURLOPT\_RETURNTRANSFER*, true);  
   // Ajoute les identifiants à la requête pour l'authentification  
   curl\_setopt($curl, *CURLOPT\_USERPWD*, "$username:$password");  
   // Désactive la vérification SSL  
   curl\_setopt($curl, *CURLOPT\_SSL\_VERIFYPEER*, false);  
    
   // Exécute la requête et récupère la réponse HTTP  
   $response = curl\_exec($curl);  
    
   // Vérifie s'il y a des erreurs lors de l'exécution de la requête  
   if(curl\_errno($curl)) {  
   // Retourne un tableau avec une clé "error" contenant le message d'erreur  
   return array("error" => curl\_error($curl));  
   } else {  
   // Décode la réponse JSON et retourne les entrées (fichiers)  
   $data = json\_decode($response, true);  
   return $data['list']['entries'];  
   }  
    
   // Ferme la connexion cURL  
   curl\_close($curl);  
  }  
  ?>

Le fichier functions.php contient une fonction appelée getFiles() qui est utilisée pour récupérer la liste des fichiers stockés dans Alfresco. Voici comment cette fonction fonctionne :

* Définition des paramètres : L'URL de l'API Alfresco et les identifiants de l'utilisateur sont définis au début de la fonction.
* Initialisation de cURL : cURL est une bibliothèque PHP qui permet de faire des requêtes HTTP. Ici, une nouvelle instance de cURL est créée pour faire une requête à l'API Alfresco.
* Configuration de la requête : L'URL de la requête est définie sur l'URL de l'API Alfresco pour récupérer les fichiers. Les identifiants de l'utilisateur sont ajoutés à la requête pour l'authentification. La vérification SSL est désactivée et l'option pour renvoyer la réponse HTTP en tant que chaîne de caractères est activée.
* Exécution de la requête : La requête est exécutée avec la fonction curl\_exec(), et la réponse HTTP est stockée dans la variable $response.
* Gestion des erreurs : Si une erreur se produit lors de l'exécution de la requête, la fonction retourne un tableau avec une clé "error" contenant le message d'erreur. Sinon, elle décode la réponse JSON et retourne les fichiers.
* Fermeture de la connexion cURL : À la fin de la fonction, la connexion cURL est fermée avec la fonction curl\_close().
* Le fichier script.js :

function renameFile(nodeId, oldName) {  
 var newName = prompt("Entrez un nom pour ce fichier ", oldName);  
 if (newName === null || newName === oldName) {  
 return;  
 }  
  
 var xhr = new XMLHttpRequest();  
 xhr.open('POST', 'rename\_file.php', true);  
 xhr.setRequestHeader('Content-Type', 'application/x-www-form-urlencoded');  
 xhr.onreadystatechange = function() {  
 if (xhr.readyState == 4 && xhr.status == 200) {  
 ***location***.reload();  
 }  
 }  
 xhr.send('nodeId=' + encodeURIComponent(nodeId) + '&newName=' + encodeURIComponent(newName));  
}  
  
function deleteFile(nodeId) {  
 if (!confirm("Voulez vous vraiment supprimer ce fichier ?")) {  
 return;  
 }  
  
 var xhr = new XMLHttpRequest();  
 xhr.open('POST', 'delete\_file.php', true);  
 xhr.setRequestHeader('Content-Type', 'application/x-www-form-urlencoded');  
 xhr.onreadystatechange = function() {  
 if (xhr.readyState == 4 && xhr.status == 200) {  
 ***location***.reload();  
 }  
 }  
 xhr.send('nodeId=' + encodeURIComponent(nodeId));  
}  
  
  
function uploadFile() {  
 var fileInput = ***document***.getElementById('file');  
 var formData = new FormData();  
 formData.append('file', fileInput.files[0]);  
  
 var xhr = new XMLHttpRequest();  
 xhr.open('POST', 'upload.php', true);  
 xhr.onreadystatechange = function() {  
 if (xhr.readyState == 4 && xhr.status == 200) {  
 var response = ***JSON***.parse(xhr.responseText);  
 if (response.hasOwnProperty('error')) {  
 alert('Error: ' + response.error);  
 } else {  
 alert('File uploaded successfully');  
 ***location***.reload();  
 }  
 }  
 };  
 xhr.send(formData);  
}  
  
  
  
function downloadFile(nodeId, fileName) {  
 const downloadUrl = `download.php?nodeId=${encodeURIComponent(nodeId)}&fileName=${encodeURIComponent(fileName)}`;  
 ***window***.location.href = downloadUrl;  
}

Le fichier script.js contient plusieurs fonctions JavaScript qui interagissent avec des fichiers PHP pour effectuer des opérations sur les fichiers stockés dans Alfresco. Voici une explication de chaque fonction :

* renameFile(nodeId, oldName) : Cette fonction est utilisée pour renommer un fichier dans Alfresco. Elle utilise une boîte de dialogue pour demander à l'utilisateur de saisir un nouveau nom pour le fichier. Si un nouveau nom est fourni, elle envoie une requête POST à rename\_file.php avec l'ID du nœud et le nouveau nom du fichier. Si la requête est réussie, la page est rechargée pour afficher les modifications.
* deleteFile(nodeId) : Cette fonction est utilisée pour supprimer un fichier d'Alfresco. Elle demande d'abord confirmation à l'utilisateur. Si l'utilisateur confirme, elle envoie une requête POST à delete\_file.php avec l'ID du nœud du fichier. Si la requête est réussie, la page est rechargée pour afficher les modifications.
* uploadFile() : Cette fonction est utilisée pour télécharger un fichier sur Alfresco. Elle récupère d'abord le fichier sélectionné par l'utilisateur, puis elle envoie une requête POST à upload.php avec le fichier. Si la requête est réussie, la page est rechargée pour afficher les modifications. Si une erreur se produit, elle est affichée à l'utilisateur.
* downloadFile(nodeId, fileName) : Cette fonction est utilisée pour télécharger un fichier d'Alfresco. Elle redirige l'utilisateur vers download.php avec l'ID du nœud et le nom du fichier en tant que paramètres de l'URL.