|  |
| --- |
| Bureau Virtuel Pléiade V2  Documentation Module Lemon LDAP |

|  |  |  |
| --- | --- | --- |
| Rédacteur | Vérificateur | Approbateur |
| **Quentin Marchand** | **Romain Leclerc** | Cédric Macaud |

|  |
| --- |
| Feuille de création et de mise à jour |

|  |  |
| --- | --- |
| Nom du Document | Référence |
| Documentation Module Lemon LDAP | Documentation Modules.docx |

|  |  |  |
| --- | --- | --- |
| Historique des évolutions | | |
| Version | Date | Description des évolutions |
| V0 | 21/02/2023 | Initialisation du document. |
| V1 | 22/02/2023 | Ajout d’informations |
| V2 | 29/08/2023 | Mise à jour des icônes |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Sommaire

1. Contexte
2. Arborescence
3. Principe de fonctionnement
4. Rendu Visuel

# Contexte

Le portail, pour se connecter utilise le protocole CAS et LemonLDAP. Une fois connecté, un cookie est généré, permettant d’être authentifié sur Lemon et sur Drupal.

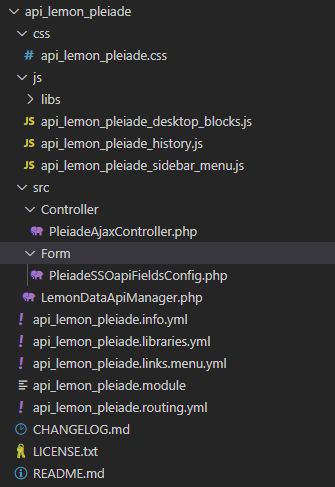
Le module permet de récupérer des informations depuis Lemon en utilisant des requêtes POST et GET.

Il permet à l’utilisateur d’accéder aux applicatifs qu’il peut utiliser en cliquant sur un lien dans un bloc. Il créé également un menu de navigation à gauche de la page.

LemonLDAP permet également d’avoir un historique de connexion. Le module permettra donc de visualiser l’historique des dernières connexions de l’utilisateur.

En dernier lieu, ce module permet de savoir à quel groupe appartient l’utilisateur. Il permettra par la suite d’afficher ou non les blocs des autres modules.

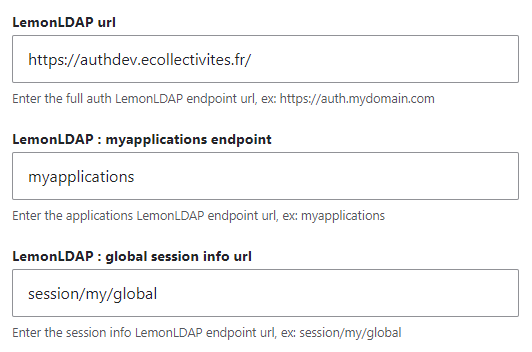
# Arborescence

Pour comprendre quel fichier sert à quoi, se référer à ce [lien](https://www.drupal.org/docs/develop/creating-modules).

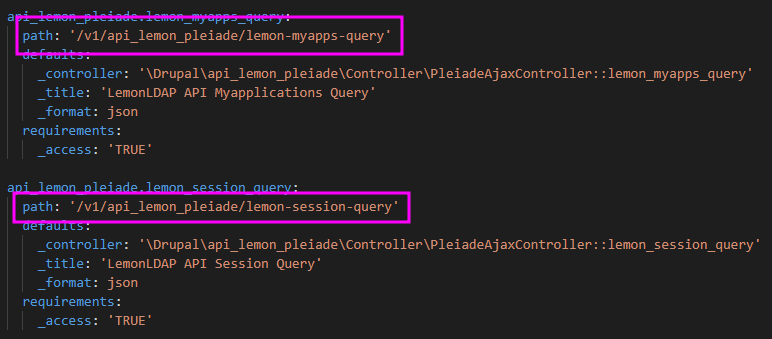
# Principe de Fonctionnement

A l’aide des différentes routes créés et du formulaire de configuration, le module va interroger Lemon LDAP à l’aide de la méthode ***executeCurl()*** contenu dans le **LemonDataApiManager.php.**

Cette fonction récupère une variable JSON en lui passant plusieurs variables : URL, Méthode, Endpoint, Paramètres. Ces derniers sont rentrés dans le formulaire de configuration du module et permettent de construire la requête de récupération des données.



Pour la récupération des applications utilisables par l’utilisateurs on utilise les paramètres ci-dessus. On stocke ensuite le JSON récupéré dans un Endpoint propre au module, un pour chaque action : récupérer les applications et l’historique.



Une fois les JSON stocké dans les URL précédente, on peut appeler cette url via une requête http XHR POST qui va chercher la variable JSON récupéré de LemonLDAP.

A partir de là, il reste plus qu’à formatter le JSON reçu pour avoir un bloc visuel et simple d’utilisation.

On commence donc par boucler sur le nombre d’entités du JSON pour créer les blocs en fonction.

On réutilise le même principe pour afficher les liens dans le menu de navigation.

Pour les blocs du tableau de bord on vérifie qu’on soit bien sur la page d’accueil pour afficher les blocs.

Pour la table de l’historique, on vérifie que la page soit bien la bonne avant d’exécuter le code.

Pour plus de lisibilité et de clarté, on a scindé les codes javascript de génération en plusieurs fichiers pour pouvoir intervenir facilement en cas de besoin. ( blocs du menu et blocs sur l’espace principal ).

La dernière version de LemonLDAP permet de récupérer les icônes associées à l’application.

On récupère donc en plus sur la route « myapps-query » les codes icônes. La bibliothèque utilisée pour les icônes est FontAwesome 6.4. et permet par un simple ajout de classe css de spécifier l’icône souhaité.

Pour les blocs principaux sur la page d’accueil, si aucun code icône est spécifié, alors on affiche l’image utilisé côté LemonLDAP.

Pour le menu de gauche, si aucun code icône est spécifié, alors on affiche une flèche.

# Rendu visuel

Une image contenant texte, capture d’écran, Police, conception

Description générée automatiquementUne image contenant texte, capture d’écran, Police, logiciel

Description générée automatiquement