|  |
| --- |
| Bureau Virtuel Pléiade V2  Documentation Module Pastell |

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

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

|  |  |
| --- | --- |
| Nom du Document | Référence |
| Documentation Module Pastell | Documentation Modules.docx |

|  |  |  |
| --- | --- | --- |
| Historique des évolutions | | |
| Version | Date | Description des évolutions |
| V0 | 22/02/2023 | Initialisation du document. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Sommaire

1. Contexte
2. Arborescence
3. Principe de fonctionnement

3.1. Entités

3.2. Documents

# 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, les applications connectées à Lemon, et sur Drupal. Une des applications qui est utilisé par les adhérents est Pastell, un logiciel de gestion de flux de documents.

Le module permet de récupérer des informations depuis Pastell en utilisant des requêtes POST et GET sous condition que l’utilisateur soit identifié.

Le module n’affiche le menu et les entitées qu’à condition que l’utilisateur appartienne au groupe « pastell » de LemonLDAP.

Les documents sont quand à eux gérés par le module Datatable.

# Arborescence

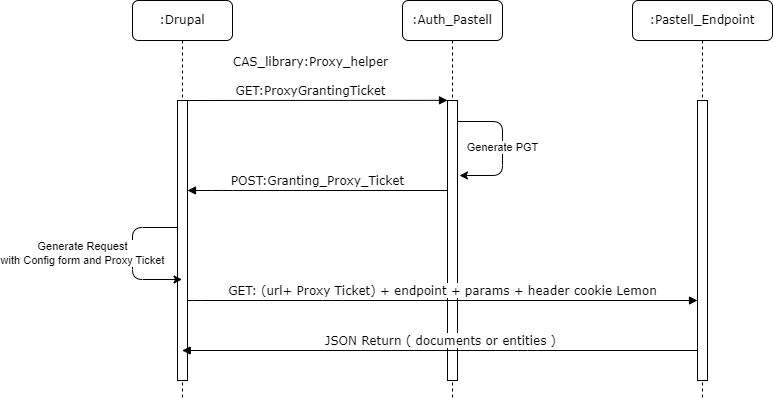
Une image contenant texte, capture d’écran, Police, logiciel

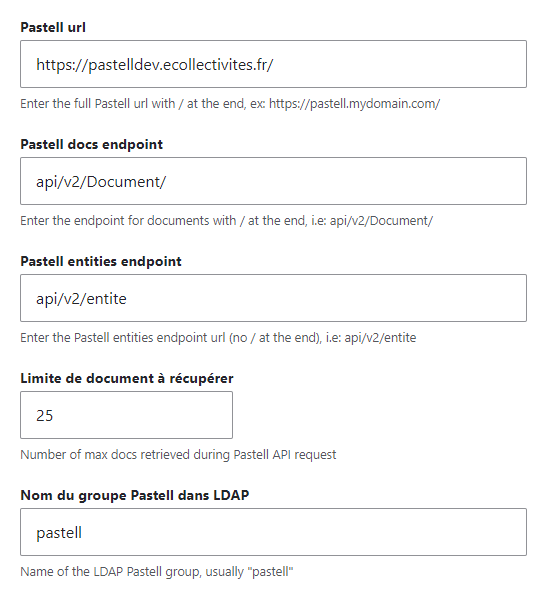
Description générée automatiquementPour comprendre quel fichier sert à quoi, se référer à ce [lien](https://www.drupal.org/docs/develop/creating-modules).

Pour plus de lisibilité le code JS a été scindé en deux parties : affichage des documents et affichage des entités.

# Principe de Fonctionnement

Pour pouvoir se connecter aux endpoints de Pastell, notre module a besoin de récupérer en premier lieu un ticket d’authentification appelé PGT ( Proxy Granting Ticket ), propre à Pastell.



A l’aide des différentes routes créés et du formulaire de configuration, le module va interroger Pastell à l’aide de la méthode ***executeCurl()*** contenu dans le **PastellDataApiManager.php,** une fois qu’il aura récupéré le PGT généré par Pastell.

Le PGT est généré en utilisant l’URL pastell dans le formulaire suivi du paramètre « **?auth=cas** ».

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.

## Entités

Pour récupérer les entités Pastell associé à l’utilisateur, on utilise le champ du formulaire « Pastell entities endpoint ».

Les entités sont récupérées sous la forme JSON suivante :

{

"id\_e": "36",

"denomination": "ACHARDS (LES)",

"siren": "200065795",

"type": "collectivite",

"centre\_de\_gestion": "31",

"entite\_mere": "0"

}

Pour chaque entité on récupère celles qui sont de type collectivité, son entité mère si elle en possède une, et se dénomination.

Au niveau de l’affichage on souhaite que chaque entités filles soit rattaché à son entité mère.

On boucle donc une première fois sur la liste des entités. Et pour chaque entité on boucle une deuxième fois pour vérifier que l’entité possède une entité fille ou non. On génère à partir de ça un formulaire html avec option.



Pour chaque choix du formulaire, on ajoute la valeur de la collectivité dans le stockage local du navigateur, afin qu’au rafraichissement de la page, l’utilisateur retrouve le choix qu’il avait avant de quitter la page.

## Documents

Pour récupérer les documents Pastell associé à l’utilisateur on utilise les champs du formulaire « Pastell docs endpoint » et « Limite de document à récupérer ».

Les documents sont récupérés sous la forme JSON suivante :

{

"id\_d": "NLS5LG7",

"id\_e": "4",

"role": "editeur",

"last\_action": "creation",

**"last\_action\_date": "2016-04-25 15:07:46",**

"last\_type": "mailsec-destinataire",

**"type": "mailsec-destinataire",**

**"titre": "",**

"creation": "2016-04-25 15:07:46",

"modification": "2016-04-25 15:07:46",

"denomination": "CDG85",

"siren": "288500028",

"date\_inscription": "2013-08-22 10:03:26",

"etat": "0",

"entite\_mere": "0",

"centre\_de\_gestion": "0",

"is\_active": "1",

"entite\_base": "CDG85",

"entite": [],

**"last\_action\_display": "creation"**

}

Pour récupérer les documents de tel ou tel entité, on spécifie dans la requête GET l’id de l’entité, précédemment stocké dans le stockage local grâce au formulaire « select ».

Pour chaque document, on récupère le titre, le type, la date de dernière modification, et la dernière action effectué.

Le module Datatable gère l’affichage des documents en fonction des entités.

## Flux

Pour récupérer les droits sur les différents types de fichier, Pastell possède une route sur son api. Le contenu de la route est stocké ensuite sur une route côté Pléiade. On affiche ensuite les actions possibles sur les types de fichier dans le menu de gauche : consulter les documents et créer un document.

Les types de document dont l’utilisateur à accès sont sous la forme JSON suivante :

"changement-email":{

"type":"Type de dossier d'administration",

"nom":"Changement d'email"

},

"actes-cdg":{

"type":"Flux CDG",

"nom":"Actes (CDG)"

},

"actes-ecollectivites-config-ged":{

"type":"Flux CDG",

"nom":"Actes (ecollectivites, configuration de la GED)"

},…

Le résultat avec les URLs utilisables directement au clic :

Une image contenant texte, capture d’écran, Police

Description générée automatiquement