

Manuel du développeur pour Mold Center

Debuigny Léo

21 août 2019

Table des matières

1	Description	3
1.1	Description du manuel du développeur	3
1.2	Utilisation des documentations	3
2	Les outils de développement	3
2.1	Description	3
2.2	Liste des aides disponibles	3
3	Gestion de l'index et des applications disponibles	4
3.1	Description	4
3.2	Ajout ou modification de l'index	4
3.3	Ajout d'une application	4
4	Système d'importation des données	5
4.1	Description	5
4.2	Étapes de fonctionnement	5
4.3	Vérification du fichier avant importation	5
4.4	Sécurité avant importation	5
5	Système d'exportation des données	5
5.1	Description	5
5.2	Implémentation de l'exportation Excel sur une autre application	5
6	Gestion du système de traduction	6
6.1	Description	6
6.2	Types de traductions	6
6.3	Ajout d'un fichier de traduction	6
6.4	Configuration du langage par défaut de l'application	6
6.5	Emplacement des sources du système de traduction	6
6.5.1	Emplacement des fonctions sources	6
7	Gestion des données	7
7.1	Description	7
7.2	Base de données JSON et SQL oracle	7
7.3	Base de données des données disponibles	7
7.4	Autre base de donnée de gestion	7
8	Gestion des utilisateurs	8
8.1	Description	8
8.2	Composition d'un utilisateur	8
8.3	Gestion des rôles et droits par défaut	8

9	Gestion des droits dynamique et des préférences des utilisateurs	9
9.1	Description	9
9.2	Le pouvoir des dictionnaires	9
9.3	HaveRight et UserHaveRight : Vérification des autorisations	9
9.4	Droits principaux sur l'ensemble des applications	9
9.4.1	Constitution du tableau des droits principaux	9
9.5	Droits sur les données pour chaque applications	10
10	Format des formulaires semi-dynamique pour l'ajout et l'édition	11
10.1	Description	11
10.2	Utilisation et mise à jour	11
10.3	Format de retour (confirmation des formulaires)	11
10.4	Format pour l'édition	11
11	Gestion des statuts, ajout ou modification	12
11.1	Description	12
11.2	Modification de la traduction d'un statut	12
11.3	Ajout d'un statut	12

1 Description

1.1 Description du manuel du développeur

Ce manuel regroupe l'ensemble technologies et fonctionnalités du centre des applications. Ce manuel permet la mise à niveau de certains éléments de base des applications et les diverses méthodes pour modifier ou ajouter des fonctionnalités dans le code source.

1.2 Utilisation des documentations

Le code source dispose de documentation pour toute les fonctionnalités réalisées. En cas de doute sur une fonctionnalité ou son fonctionnement il faut consulté les documentations correspondante. Des schéma divers sont également disponible pour tout ce qui concerne la base de données ou la gestion des données au sein de l'application.

2 Les outils de développement

2.1 Description

Divers outils de développement sont disponibles directement sur la plateforme web. pour y accéder il faut un compte administrateur et aussi changer la valeur de la variable principal dans le contrôleur "developController" pour activer le système d'aide.

2.2 Liste des aides disponibles

1. Visualisation des tables utilisées par l'application web.
2. Ajout de données en base de données.



3 Gestion de l'index et des applications disponibles

3.1 Description

L'index est un système qui permet d'afficher de manière dynamique les différentes applications disponible sur le centre des applications.

3.2 Ajout ou modification de l'index

bdd/INDEX.json Fichier JSON qui contient les informations pour chaque application, il suffit d'ajouter un élément dans ce fichier pour voir l'application apparaître.

3.3 Ajout d'une application

Pour ajouter une application il est possible de faire une copie d'une déjà existante si elle doit utiliser les même technologies, sinon il suffit de créer une pack de contrôleurs avec son modèle et sa vue, puis ajouter la route à l'index pour prendre en compte la nouvelle application.

4 Système d'importation des données

4.1 Description

L'importation des données Excel ne s'effectue normalement qu'une seule fois, pour utiliser le système d'importation il faut un fichier Excel formaté correctement suivant le modèle donnée. Ensuite il faut activer le système d'importation et utiliser un compte administrateur pour importer les données. Il est également possible de voir les ajout de données de façon précise.

4.2 Étapes de fonctionnement

1. Préparer le fichier Excel contenant les données en suivant l'exemple du fichier ./01_Analyse/Data_List/Mold_List.xlsx et Mold_Spec.xlsx
2. Accéder au contrôleur du système d'importation : inc/Controllers/importController.php
3. Changer la variable ACTIVATED à True pour activer le système d'importation
4. Changer la variable VISUAL à True si l'on souhaite voir les étapes d'ajout des données
5. Changer la variable UPLOAD à True pour valider l'upload des données.

4.3 Vérification du fichier avant importation

Il est possible de visualiser l'ajout de données sans modifier la base pour vérifier si le fichier Excel est correctement formaté. Pour cela il suffit d'activer le système d'importation, d'activer l'affichage des étapes d'ajout et de désactiver le système d'upload des données.

4.4 Sécurité avant importation

Avant d'effectuer une importation il vaut mieux créer une table pour tester le système (changer le modèle pour importer les données sur la table en question). Puis une fois la table correctement remplie avec les données importées il suffit de changer le nom de la table et la faire correspondre à la table final voulu.

5 Système d'exportation des données

5.1 Description

Le système d'exportation des données est utilisable sur toute les applications disponibles. Il offre la création simple d'un fichier Excel avec le téléchargement de celui-ci par la suite avec des changements de style et différents outils supplémentaire.

5.2 Implémentation de l'exportation Excel sur une autre application

Le fichier Excel contenant les données est généré automatiquement avec les droits sur les données de l'utilisateur. Ainsi pour toute les applications utilisant le même format il suffit de copier le template d'exportation d'une application dans une autre pour intégrer le l'importation des données en fichier Excel.

6 Gestion du système de traduction

6.1 Description

Le système de traduction permet à l'aide d'une fonction dans le frontcontroller de récupérer les traductions pour chaque éléments et aussi de vérifier l'existence de la traduction en question.

6.2 Types de traductions

Fichier normal -XX.ini Ces fichiers permettent d'effectuer la traduction d'une page web particulière.

Fichier pour les données _List-XX.ini Ces fichiers permettent de traduire des données venant par exemple de la base de données ou de l'index.

6.3 Ajout d'un fichier de traduction

Préparation Pour ajouter un fichier de traduction il suffit de le nommer en suivant l'exemple des fichiers déjà présent.

Ajout de la traduction Ensuite on ajoute par fonctionnement de clé => valeurs l'ensemble des traductions nécessaire.

Implémentation dans une page Pour finir on importe dans le contrôleur le ou les fichiers de traduction puis on les utilise par la suite dans la vue comme un tableau normal en PHP.

Traduction Pour terminer il faut copier le ou les fichiers de traduction créer pour effectuer la traduction pour les autres langages disponible.

6.4 Configuration du langage par défaut de l'application

Un fichier config.ini disponible dans `/inc/Controllers/Translate/config.ini` permet de choisir les paramètres de l'application. Pour changer la langue il suffit de choisir parmi ceux proposé et changer la valeur de "defaultTranslate". (Actuellement seul fr et en sont disponible)

6.5 Emplacement des sources du système de traduction

Fonctionnement Pour utiliser la traduction, on utilise une clé qui vient d'un fichier de traduction ini qui contient la traduction en question dans les langages disponible.

6.5.1 Emplacement des fonctions sources

Le Controller de base contient les deux fonctions qui permettent d'utiliser la traduction.

getVerifyTranslage Cette fonction permet de récupérer une traduction contenu dans un dictionnaire en vérifiant qu'elle existe.

getTranslate Cette fonction permet de récupérer le langage voulu actuellement et si il est inexistant alors la langue par défaut est utilisé et enregistré. Pour finir la fonction renvoi le fichier de traduction voulu qui permet de faire la liaison entre les données et leurs traductions correspondante.

setTranslate Cette fonction permet de changer la langue de l'application sur la session en cours. Un fois la langue sélectionner cela change la variable de langage et change donc la langue de l'application.

7 Gestion des données

7.1 Description

Les données sont gérées en partie par la base de données. Ainsi lorsque l'on souhaite ajouter des données il suffit de l'ajouter en base et dans la modèle correspondant. L'application ira ensuite récupérer les informations sur la donnée en question et l'utilisera pour l'afficher et l'utiliser.

7.2 Base de données JSON et SQL oracle

JSON Certaines bases de données sont en JSON pour décharger la charge sur la base de données et avoir des informations rapidement en local. L'index, les descriptions des droits principaux et les droits par défaut utilisateur sont en fichier JSON dans le dossier bdd à la racine.

SQL Oracle Le reste et le plus gros des données est contenu dans les bases de données, ainsi des scripts de démarrage sont disponibles pour créer l'ancienne et la nouvelle base de données pour utiliser l'ensemble du site web.

Emplacement des fichiers de base de données L'ensemble des fichiers sources relatif aux bases de données sont disponibles dans le dossier bdd à la racine du projet.

7.3 Base de données des données disponibles

Description Chaque donnée dispose d'une description et de la liste des données disponibles. Cela permet de gérer dynamiquement l'ensemble des applications et des données.

Ajout d'une nouvelle donnée Pour ajouter une nouvelle donnée il suffit de suivre l'exemple donnée par les autres données puis attribuer les droits par la suite avec le compte administrateur. Chaque nouvelle donnée a besoin d'être inscrite dans la base de données avec les outils du développeur par exemple, ensuite il faut ajouter la traduction de la donnée correspondante et vérifier qu'elle est intégrée dans le modèle.

7.4 Autre base de données de gestion

Données des statuts Chaque statut est décrit dans la base de données, cela permet également la gestion dynamique des statuts et des filtres.

8 Gestion des utilisateurs

8.1 Description

Le système pour gérer les utilisateur est CRUD normal. L'utilisateur dispose de certaines informations qui lui sont propre.

8.2 Composition d'un utilisateur

1. Identifiant de l'utilisateur : Toujours en majuscule
2. Nom et prénom de l'utilisateur : Fusionné et séparé par un espace (explode(' ', \$NOMPRE))
3. Mot de passe de l'utilisateur : Hasher par défaut en sha1
4. Droits principaux de l'utilisateur : Liste des droits par application en JSON
5. Droits sur les données : Liste des droits et préférences par application des données le tout en JSON

8.3 Gestion des rôles et droits par défaut

bdd/DEFAULT_USER_RULE Il existe un fichier JSON qui répertorie les rôles disponible lors de la création d'un utilisateur et les droits par défaut attribué aux utilisateurs. Il suffit de changer ce fichier pour ajouter de nouveaux rôles ou simplement modifier les droits par défaut ou les rôles disponibles.

9 Gestion des droits dynamique et des préférences des utilisateurs

9.1 Description

Les droits dynamiques permettent de définir des droits sur l'ensemble des applications à un utilisateur mais aussi de définir les droits et préférences pour chaque donnée d'une application. Le tout est sauvegarder pour chaque utilisateur, ce qui permet d'avoir un système flexible avec des préférences différentes pour chaque utilisateur.

9.2 Le pouvoir des dictionnaires

L'ensemble des tableau de droits sont des dictionnaire (tableau clé => valeur), cela permet d'utiliser la fonction `isset` très pratique pour déterminer par exemple si un utilisateur dispose d'un droit ou non.

9.3 HaveRight et UserHaveRight : Vérification des autorisations

HaveRight C'est une fonction du frontcontrollers qui permet de savoir si l'utilisateur connecté dispose d'un droit principaux particuliers.

Pour autoriser uniquement l'administrateur il suffit de mettre un `HaveRight('admin', 'ALL')` et pour autoriser le modérateur il suffit de mettre un `HaveRight('moderator', 'app')`. Si dans un dernier cas on souhaite savoir si l'utilisateur est modérateur d'au moins une application on peut utiliser `HaveRight('moderator', 'None')`.

Sinon pour une vérification avec un utilisateur classique il suffit d'utilise `HaveRight('right', 'app')`. La liste des applications est disponible avec l'index et la liste des droits est disponible dans le dossier bdd qui contient la description des droits principaux.

UserHaveRight C'est une copie de la fonction `HaveRight` mais où il est possible de spécifier l'utilisateur qui dispose du droit en question.

9.4 Droits principaux sur l'ensemble des applications

9.4.1 Constitution du tableau des droits principaux

Administrateur L'administrateur dispose de toute les droits sur toute les applications, il est donc pas nécessaire de lister l'ensemble de ses droits. Il dispose donc uniquement d'un "admin" à la racine du tableau.

```
1      {
2      "admin": true
3      }
```

Modérateur Le modérateur est un administrateur mais uniquement sur une ou plusieurs applications.

```
1      {
2      "app_01": {
3      "moderator": true
4      },
5      "app_02": {
6      "moderator": true
7      }
8      }
```

Utilisateur normal Un utilisateur normal dispose de droits spécifique pour chaque application.

```
1      {
2        "app_01": {
3          "view": true,
4          "add": true,
5          "edit": true,
6          "remove": true
7        },
8        "app_02": {
9          "view": true,
10         "edit": true,
11       }
12     }
```

ATTENTION droit d'ajout ! Le droit d'ajout nécessite les droits de visualisation pour toute les données de l'application, tout simplement car il est impossible de créer un objet sans toute ses informations.

9.5 Droits sur les données pour chaque applications

Description Chaque application dispose de certaines données et certains filtres. Le but est de conservé les droits et les préférences utilisateurs sur ces données en question. On utilise donc un système de droit pour chaque application pour chaque donnée. Sachant que les modérateur ou administrateur dispose de l'ensemble des données.

```
1      {
2        "Mold_App": {
3          "STATUS_MOULE": {"edit": true, "search": true, "primary": true
4          },
5          "REF_MOULE": {"edit": true, "search": true, "primary": true},
6          "PATTERNSIZE": {"edit": true, "search": true, "primary": true}
7          ,
8          "CODE_PROD_2": {"edit": true, "search": true, "primary": true}
9          ,
10         "REF_DIMENSION": {"edit": true, "search": true, "primary":
11         true},
12         "DIM_CTN2": {"edit": true, "search": true, "primary": true},
13         "FILTER": {"operation": true, "netoyage": false}
14       }
15     }
```

10 Format des formulaires semi-dynamique pour l'ajout et l'édition

10.1 Description

Le système de formulaire pour les applications est un peu particuliers, en effet les formulaires sont géré de manière semi-dynamique.

10.2 Utilisation et mise à jour

Un switch case est utilisé pour déterminer quels données appartiennent à quel genre de formulaire, ainsi il est possible d'avoir un formulaire en partie dynamique et propre.

Pour ajouter ou modifier un champ il suffit soit de le laisser comme tel afin qu'il prennent le formulaire par défaut (du texte simple), ou alors il est possible de l'ajouter à d'autre formulaire particuliers comme les dates voir de créer un tout nouveau genre de formulaire pour lui et d'autre qui suivront plus tard.

10.3 Format de retour (confirmation des formulaires)

Chaque champ du formulaire porte le nom de sa donnée en tant que nom dans le formulaire, ainsi la récupération des champ voulu est très simplifié.

10.4 Format pour l'édition

L'édition est un peu particuliers car certains utilisateur ne pourront pas modifier toute les données qu'il veulent, ainsi il devient nécessaire d'avoir plusieurs étapes avant de récupérer les données.

1. Récupération des données avant modification.
2. Remplacement des données modifier avec les nouvelles données.
3. Envoi dans le système d'historique les deux tableau de données pour une comparaison.
4. Modification des données dans la base.

11 Gestion des statuts, ajout ou modification

11.1 Description

Le système de statut est très simple à prendre en main. Il permet entre autre d'ajouter des statuts disponible et utilisable. Ils apparaîtront automatiquement en tant que filtre et devront être autorisé par un modérateur ou un administrateur dans la configuration utilisateur.

11.2 Modification de la traduction d'un statut

Pour changer la traduction d'un statut il suffit d'aller dans le fichier de traduction `/inc/Controllers/Translate/fr/Status_fr.ini` et de modifier la ligne voulu pour changer la traduction d'un statut.

11.3 Ajout d'un statut

L'ajout d'un statut est pratiquement dynamique, il faut suivre quelque étapes pour en ajouter un :

1. Ajouter la ligne pour le statut dans la base de données sur la table `MOLD_STATUS_TYPE`.
2. Ajouter la traduction du statut dans le fichier `/inc/Controllers/Translate/fr/Status_List-fr.ini` (ajouter Description pour effectuer la description du statut).
3. Ajouter le statut dans les comptes utilisateurs via la plateforme web.