protoDoc Documentation

Version 0.0.1

Marieme Doua

28 November 2016

CeRTAE ProtoExt est une application web qui propose une approche basée sur la méthode DATARUN. Cette approche consiste à réaliser les interfaces d'une application par la construction de vues à partir d'un modèle de données standard.

Cette documentation présente un guide pour les utilisateurs de cette application.

Table de matière

1.1 La structure de l'interface

1.1.1 Organisation de l'interface

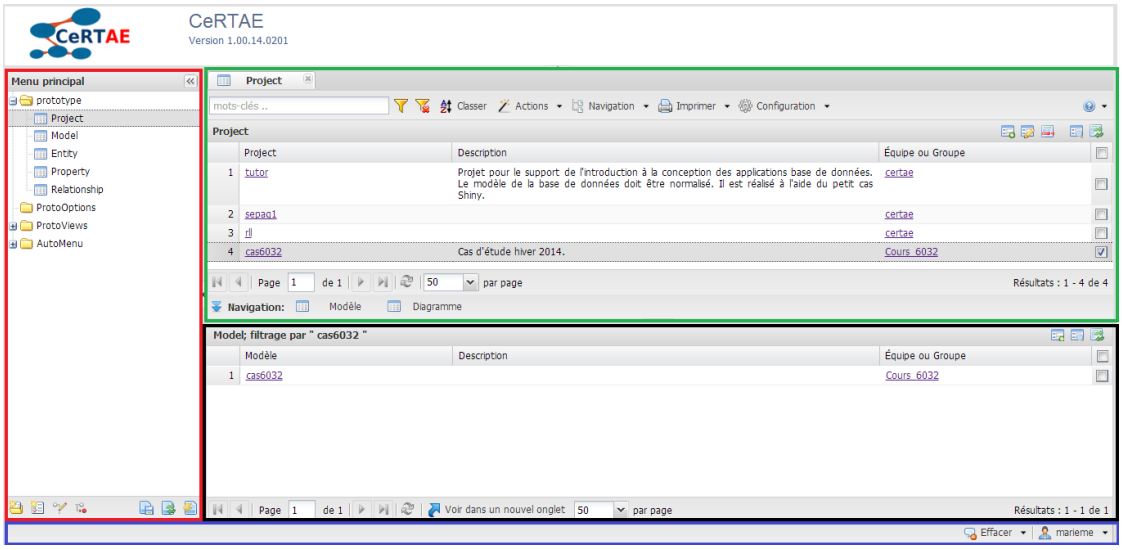


Figure 1 : Organisation de l'interface de l'application.

L'interface de l'application ProtoExt est organisée en plusieurs zones, qui sont :

Le menu principal

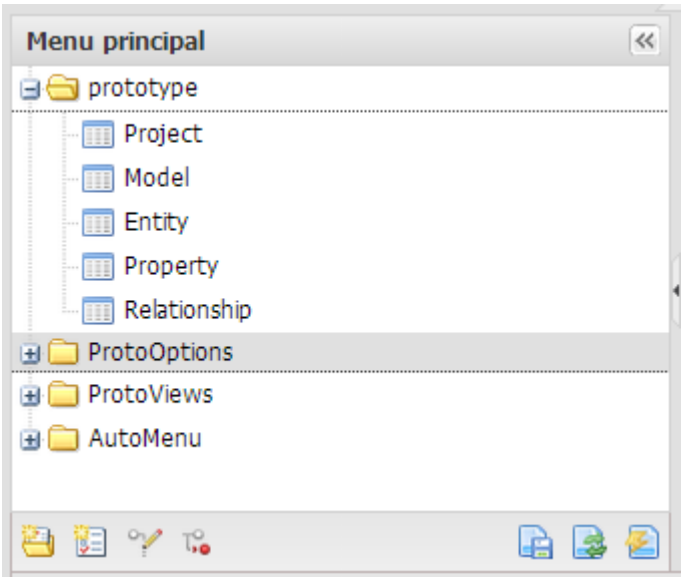


Figure 2 : Menu principal.

Le menu principal affiche le contenu de chaque objet disponible sur l'arborescence de l'application ainsi que la liste des projets et leurs vues prototypes.

Pour afficher le contenu d'un objet du menu, double-cliquez sur celui-ci. Il sera ouvert comme un nouvel onglet dans la grille principale.

Il est possible de cacher le menu principal en cliquant sur le bouton  .

Pour réafficher le menu principal, cliquez sur le bouton  .

Les fonctions du menu principal

Figure 3 : fonctions du menu principal.

1- Pour déplacer un dossier, cliquez sur celui-ci, glissez-le jusqu'au l'endroit voulu (sans relâcher la souris).

2- : option pour créer un nouveau dossier dans l'arborescence.

3-  : option pour créer une nouvelle vue sur une entité. Sélectionnez le dossier où vous voulez placer la nouvelle vue, et remplir les champs :

- text : le nom de la vue ;
- option : cherchez l'entité à partir de la liste des options ;
- iconCls : nom de l'icône ;
- qtip : pour créer un tooltip (infobulle).
- qtitle : pour le titre à afficher lorsque la vue est ouverte dans la grille principale.

Les deux premiers champs sont obligatoires, les autres sont optionnels.

4- : option pour modifier le nom de l'objet sélectionné.

5- : option pour supprimer un objet dans l'arborescence. L'objet sera supprimé du menu (pas de la BD).

6- : option pour enregistrer l'état actuel du menu principal. Toutes les modifications (personnalisations) réalisées dans l'arborescence seront enregistrées. Si vous n'enregistrez pas vos modifications, l'application chargera toujours le menu par défaut (ou le dernier menu enregistré).

7- : option pour actualiser le menu (retourne le dernier menu enregistré).

8- : option pour réinitialiser le menu (retourne le menu par défaut).

La grille principale

La grille principale, au centre de l'interface, sert à afficher le contenu des objets sélectionnés dans le menu principal.

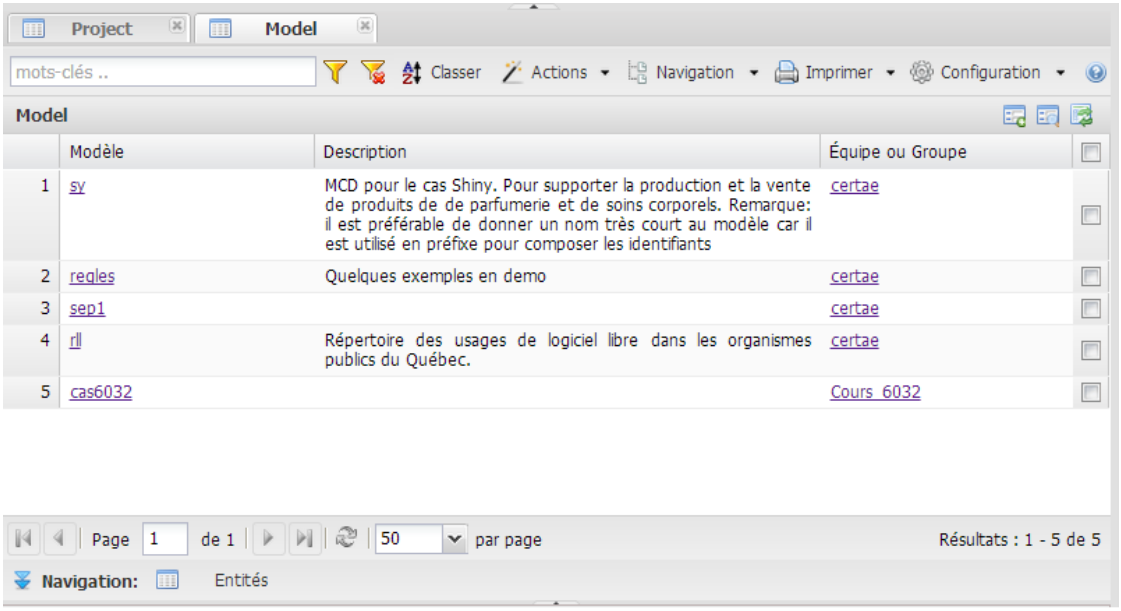


Figure 4 : grille principale de l'application.

Les onglets

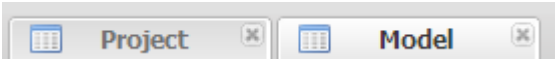


Figure 5 : exemple d'onglets.

Les onglets correspondent à la liste des objets ouverts à partir du menu principal. Chaque fois qu'un objet est ouvert, un nouvel onglet apparaîtra. Vous pouvez vous déplacer d'un contenu à l'autre en cliquant sur son onglet. La grille principale affichera le contenu de l'onglet sélectionné.

Plusieurs onglets peuvent être ouverts en même temps, mais il est déconseillé d'ouvrir plus de six à la fois, car cela pourrait avoir un effet sur la mémoire et ralentir l'application.

Le menu des fonctions

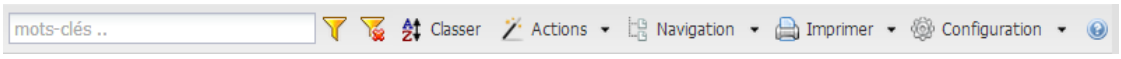


Figure 6 : menu des fonctions

Le menu des fonctions regroupe les fonctions disponibles pour la grille principale. Ces fonctions sont explorées un peu plus tard dans cette documentation.

Le menu d'édition

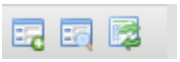


Figure 7 : menu d'édition

Le menu d'édition permet l'ajout, la modification, la suppression ou la consultation des enregistrements à partir de la grille principale ou la grille de navigation.

 Ajouter : permet d'ouvrir un formulaire pour ajouter un nouvel enregistrement dans l'objet ouvert. Remplir les champs requis dans le formulaire et cliquer sur le bouton **Enregistrer**. Le nouvel enregistrement sera ajouté en dernière position sur la grille.

 Editer : sélectionner un enregistrement pour faire apparaitre ce bouton. Cliquez sur ce dernier pour ouvrir le formulaire de modification. Modifier les champs voulus et cliquer sur le bouton **Enregistrer**.

 Consulter : sélectionner un enregistrement pour faire apparaitre ce bouton. Cliquez sur ce dernier pour ouvrir le formulaire de consultation.

 Copier : sélectionner un enregistrement et cliquer sur ce bouton pour dupliquer celui-ci. La ligne copiée apparaitra dans la première ligne de la grille principale.

 Supprimer : sélectionner un enregistrement pour faire apparaitre ce bouton. Cliquez sur ce dernier pour supprimer celui-ci, une confirmation est requise. Une fois l'objet supprimé, il n'est plus possible de le récupérer.

Note : La suppression d'un objet entraine la suppression de ses détails. Si un projet est supprimé, la suppression est propagée vers les modèles, les entités, les attributs, les associations et les vues pour ce projet. Idem pour les modèles, les entités, etc.

La grille

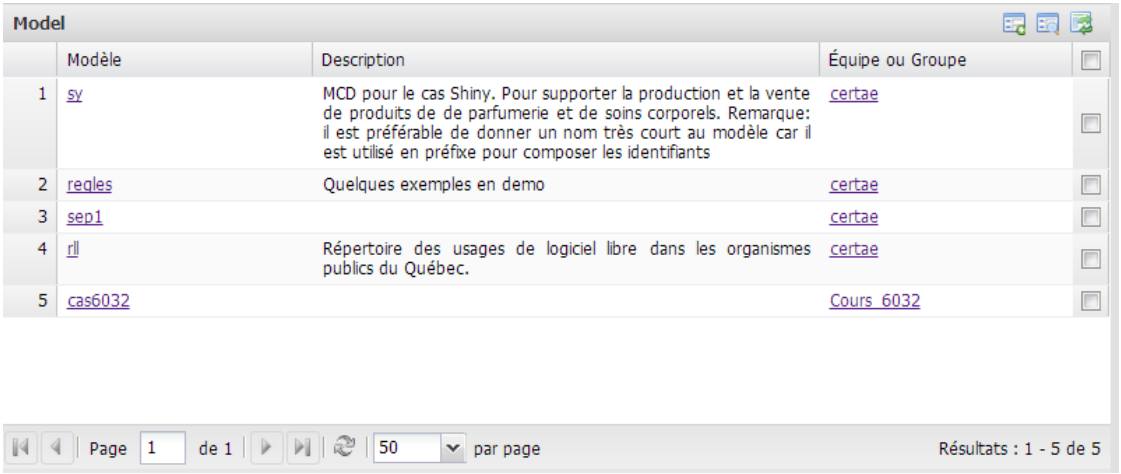


Figure 8 : contenu de la grille

Le centre de la grille principale sert à afficher le contenu de l'objet de l'onglet actif.

Vous pouvez naviguer entre les pages et personnaliser le nombre de résultats par page à partir de la barre en dessous de la grille.

La barre de navigation

Figure 9 : menu de navigation

Si l'objet sélectionné dans la grille principale a des détails (relations avec d'autres objets), ces détails s'affichent automatiquement sur cette barre.

- Pour afficher un détail, cliquez sur son nom dans la barre de navigation (exemple : Entités dans la *Figure 9*).
- Pour cacher la grille de navigation, cliquez sur l'icône Navigation de la barre de navigation.

La grille de navigation

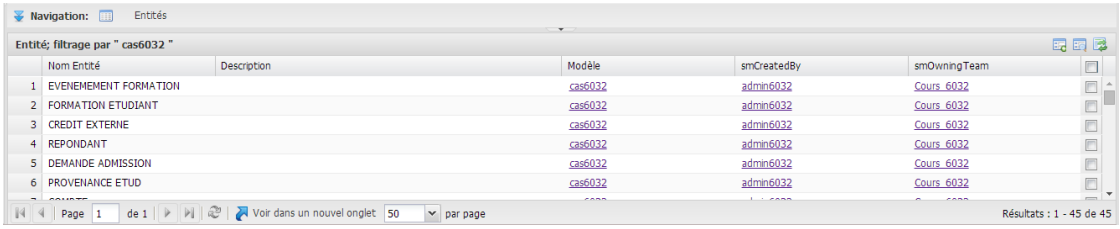


Figure 10 : Grille de navigation.

La grille de navigation permet d'afficher et de parcourir les détails de l'objet sélectionné dans la grille principale :

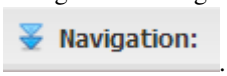
- à partir d'un projet, vous pouvez naviguer vers ses modèles ;
- à partir d'un modèle, vous pouvez naviguer vers ses entités ;
- à partir d'une entité, la navigation se fait vers ses propriétés, ses relations et ses vues.

Deux façons pour afficher la grille de navigation :

- par la **fonction Navigation** du menu des fonctions : cliquez sur le bouton **Navigation** ;
- par la **barre de navigation** de la grille principale : cliquez sur une des options situées sur la barre de navigation.

Pour voir les détails d'un autre objet, vous n'avez qu'à cliquer sur celui-ci dans la grille principale et le contenu de la grille de navigation sera actualisé automatiquement.

Pour visualiser l'information d'un modèle de la grille de navigation, sélectionnez le modèle sur la grille, ensuite, cliquez sur l'icône  pour ouvrir le formulaire de visualisation en mode de lecture.

Pour cacher la grille de navigation, cliquez sur le bouton  situé en haut au milieu de cette même grille ou sur le bouton 

.

Il est possible de choisir le nombre de résultats affichés par page, rafraichir les résultats affichés sur la grille et ouvrir les résultats dans un nouvel onglet à partir de la barre en dessous de la grille.

La barre d'états

Figure 11 : Barre d'états.

La barre d'états est divisée en deux parties :

la barre de message :

L'application du prototypeur communique avec l'utilisateur par l'entremise de la barre de messages pour confirmer une opération ou envoyer un message d'erreur. Les messages d'erreur sont affichés en utilisant une police en couleur rouge.

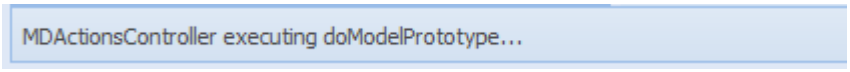


Figure 12 : exemple de message

Dans la figure ci-dessus, le message montre l'exécution de l'opération « doModelPrototype ».

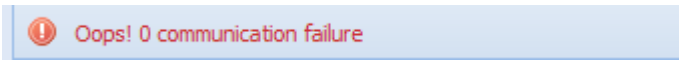


Figure 13 : exemple de message d'erreur.

Exemple de message d'erreur : problème de communication avec le serveur.

Les boutons :

Figure 14 : Boutons de la barre d'états.

— Le premier bouton permet de fermer tous les onglets ouverts et vider la mémoire cache.
— Le deuxième bouton affiche le nom de l'utilisateur, vous pouvez cliquer sur la flèche noire de ce bouton pour accéder au bouton de fermeture de session.

1.1.2 Menu des fonctions :

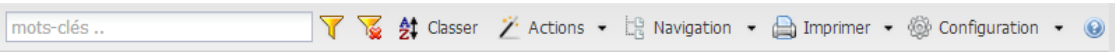


Figure 15 : menu des fonctions

Ce menu regroupe les fonctions pour manipuler l'objet affiché dans la grille principale.

Si vous cliquez sur le bouton, le sous-menu correspondant apparait au-dessous du menu de fonctions.

Si vous cliquez sur la flèche noire (à côté du bouton), une liste de visualisation du sous-menu apparait.

Dans les deux cas, vous pouvez accéder aux options pour la fonction choisie.

Les fonctions de la grille dépendent de l'objet ouvert. Tous les objets n'ont pas les mêmes fonctions.

Les fonctions de l'application sont :

Rechercher

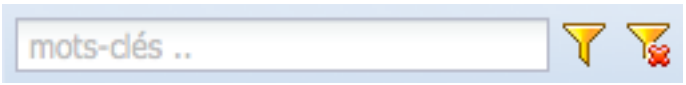


Figure 16 : fonction Rechercher

Cette fonction permet de filtrer les résultats de la grille principale à partir du critère recherché (sensible aux accents).

Pour rechercher, cliquez sur le bouton  ou sur la touche *entrer*.

Pour effacer le critère, cliquez sur le bouton  .

Le filtrage s'applique sur plusieurs critères, par exemple, la colonne « nom » et la colonne « description » de l'onglet actif.

Les colonnes qui peuvent être filtrées peuvent être personnalisées à partir de la fonction configuration.

Classer

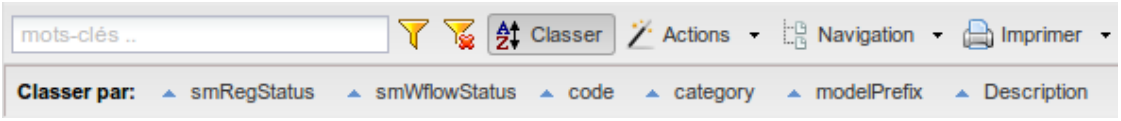


Figure 17 : fonction Classer

Cette fonction permet de trier la grille principale selon la colonne choisie. En cliquant sur ce bouton, la liste d'option disponible pour le tri s'affiche.

Cliquez sur la colonne choisie. Cliquez une deuxième fois, sur la même colonne pour inverser l'ordre de tri.

Les colonnes qui peuvent être filtrées peuvent être personnalisées à partir de la fonction configuration.

Actions

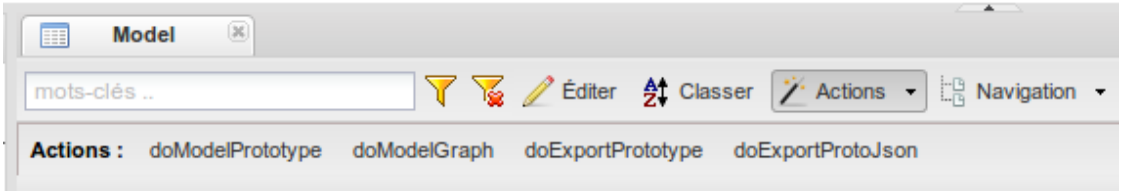


Figure 18 : fonction Actions

Selon l'objet actif, différentes actions sont disponibles. Ces actions sont entre autres :

— **doImportSchema :**

L'action doImportSchema permet d'importer un schéma de base de données (exemple un schéma .sql) dans l'application.

— **doImportOMS :**

L'action doImportOMS permet d'importer un modèle OMS (open modèle sphère) dans l'application à fin de générer le prototype.
Pour le moment, le fichier doit être nommé OMS.exp et placé dans le dossier media.

— **doModelPrototype :**

L'action doModelPrototype génère le prototype pour un modèle sélectionné.

— **doModelGraph :**

L'action doModelGraph génère le modèle conceptuel graphique pour un modèle sélectionné.

— **doExportPrototype :**

L'action doExportPrototype permet l'exportation du prototype sous format d'un modèle django (models.py).

— **doExportProtoJson :**

L'action doExportProtoJson permet l'exportation du prototype sous format d'un fichier json.

— **doEntityPrototype :**

L'action doEntityPrototype génère le prototype pour une entité sélectionnée.

— **doWFlowResume :**

L'action doWFlowResume affiche la liste des modifications en attente de validation (voir workflow).

— **Accepter et Refuser :**

Ces actions permettent d'accepter ou de refuser un (workflow) en attente.

Navigation

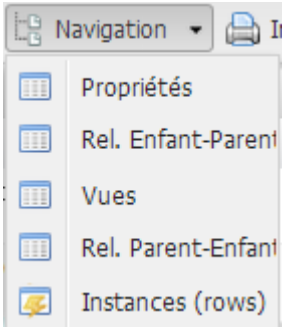


Figure 19 : Fonction navigation et son sous-menu pour l'entité.

La fonction Navigation permet de naviguer vers les détails d'un objet sélectionné.

En cliquant sur ce bouton, la grille de navigation s'affiche.

Imprimer

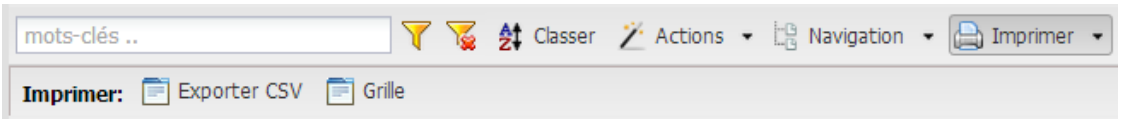


Figure 20 : Fonction imprimer et son sous-menu.

La fonction **imprimer** permet d'imprimer le contenu de la grille principale. Le sous-menu d'imprimer est composé de deux options : **Exporter CSV** et **Grille**. Notez que vous pouvez aussi accéder aux options du sous-menu en cliquant sur le petit triangle noir situé à droit du bouton imprimer.

Quand vous cliquez sur l'option **Grille**, un nouvel onglet sera ouvert automatiquement dans votre fureteur, la fenêtre du dialogue d'impression sera affichée à l'écran. Selon le type d'imprimante installée sur votre ordinateur, vous pourriez enregistrer le résultat de l'impression en format PDF. Le contenu de la grille est transformé dans un format de sortie en HTML (le résultat est montré dans le nouvel onglet ouvert). Vous pouvez enregistrer le contenu de cet onglet comme une page web (HTML).

Quand vous cliquez sur l'option **Exporter CSV**, une fenêtre sera affichée à l'écran. À partir de cette fenêtre, il est possible d'ouvrir le fichier avec un logiciel de feuille de calcul comme Microsoft Excel ou Libre Office Calc ou de l'enregistrer sur votre ordinateur.

Note : Notez que le contenu de la grille principale est imprimé ou exporté au complet. Par exemple, si vous avez 3000 résultats qui sont distribués dans 10 pages, quand vous utilisez une des fonctions Grille ou Exporter CSV, les 3000 résultats seront imprimés ou exportés. Il n'est pas nécessaire de se déplacer à chacune de pages pour faire l'impression/exportation.

Filtres

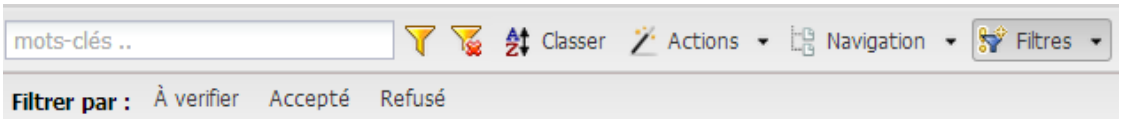


Figure 21 : Fonction filtrer et son sous-menu.

Cette fonction s'affiche avec les objets qui ont un workflow. Les options sont :

- **À vérifier :** affiche la liste des objets modifiés par l'utilisateur et en attente d'acceptation par l'administrateur.
- **Accepté :** affiche la liste des objets modifiés par l'utilisateur et acceptés par l'administrateur.
- **Refusé :** affiche la liste des objets modifiés par l'utilisateur et refusés par l'administrateur.

Onglets

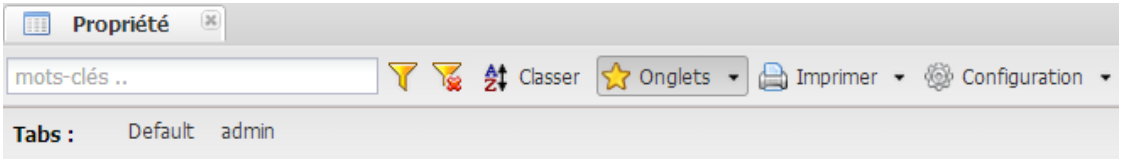


Figure 22 : Fonction onglets et son sous-menu.

Pour l'objet Proprieté, il existe une fonction Onglets qui permet d'afficher les propriétés de deux façons :

- **Default :** pour l'affichage par défaut nom de propriété ;
- **admin :** permet d'afficher les propriétés préfixées par le nom de la table.

Configuration

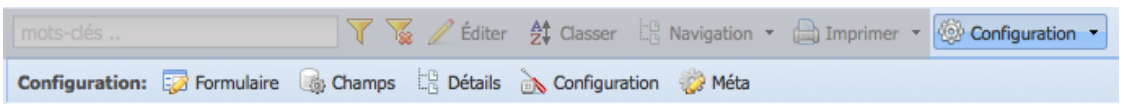


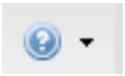
Figure 23 : Fonction configuration et son sous-menu.

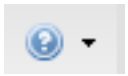
Cette fonction permet la personnalisation de l'application.

Note : L'accès à toutes les options de la fonction Configuration est réservé aux utilisateurs avec les droits. Si votre compte n'a pas les droits suffisants, vous verrez uniquement les options pour la configuration de base.

Quand vous cliquez sur le bouton Configuration (ou sur le petit triangle noir à côté), un sous-menu apparait. Ce sous-menu est composé des options suivantes :

1. Formulaire ;
2. Champs ;
3. Détails ;
4. Configuration ;
5. Méta.

Aide 

Le dernier bouton de menu des fonctions  est le bouton d'aide en ligne de l'application.

1.2 La configuration de l'application

1.2.1 Formulaire

À partir de cette option, vous pouvez personnaliser le formulaire qui s'affiche pour le menu d'édition.

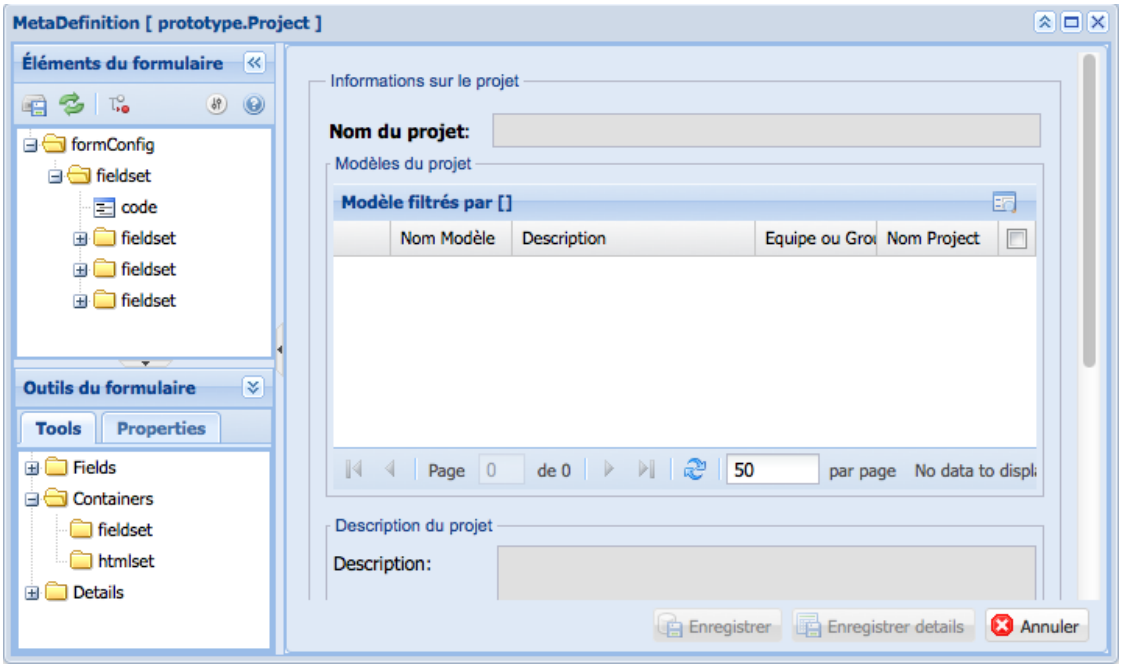


Figure 24 : Option formulaire du menu configuration.

La fenêtre du formulaire est organisée en trois grandes sections :

1. **Les éléments du formulaire :**

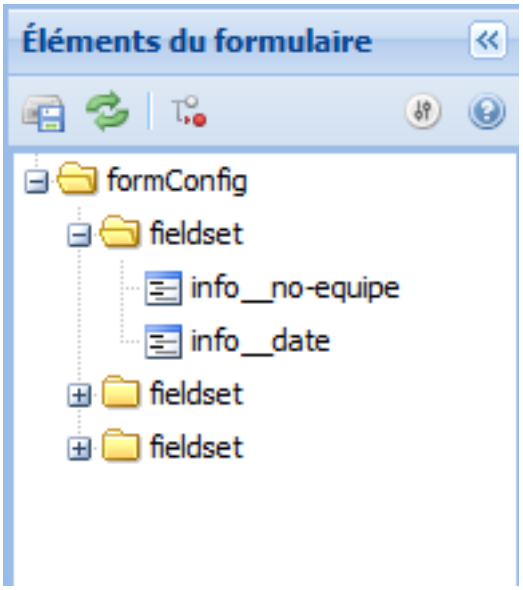


Figure 25 : la fenêtre des éléments du formulaire.

Les éléments du formulaire (côté haut-gauche de la fenêtre) affichent dans une arborescence les champs actifs sur les formulaires d'édition.

À partir de cette fenêtre, il est possible de personnaliser l'affichage et l'ordre d'apparition de champs sur le formulaire. L'ordre d'apparition de champs est défini selon la position de l'objet dans l'arborescence. Il est possible aussi de regrouper les champs à l'intérieur des contenants de type « Fieldset » ou « HTMLset ».

les icônes de cette fenêtre sont :

—  : pour enregistrer les modifications du formulaire.
—  : pour rafraîchir l'aperçu du formulaire (côté droit de la fenêtre)
—  : pour supprimer un objet dans l'arborescence.
— : pour cacher la fenêtre des éléments du formulaire de l'interface.
—  : icône attribuée aux champs.
—  : icône attribuée aux contenants de type fieldset et HTMLset.

2. **Les outils du formulaire :**

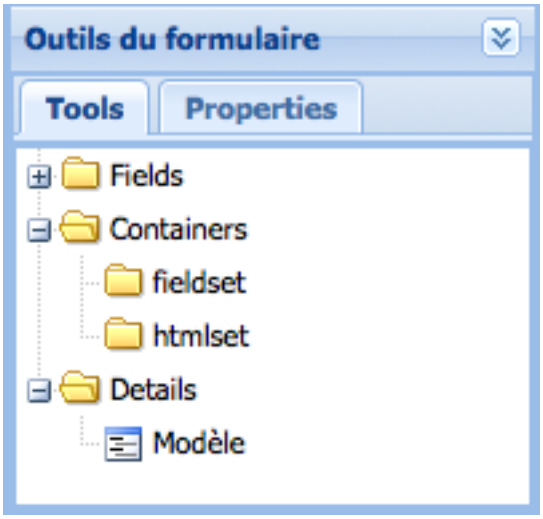


Figure 26 : les outils du formulaire.

La fenêtre des outils (Tools) du formulaire (côté bas-gauche de la fenêtre, donne accès aux outils permettant de créer et de personnaliser un formulaire. Les outils sont organisés dans une arborescence où ils se retrouvent les champs, les contenants et les détails pour une vue.

L'onglet des propriétés (Properties) donne accès aux propriétés de personnalisation de l'objet sélectionné dans la fenêtre des éléments.

Pour l'onglet Outils :

— **Fields (Champs) :**

Le dossier Fields contient la liste complète des champs pour l'objet sélectionné (un projet, un modèle, une entité, une propriété, une relation ou une vue). Dans le cas d'une vue, les champs correspondent aux propriétés ou attributs des entités crées par l'utilisateur. On y retrouve aussi des champs crées par l'application du prototypeur comme la date de la dernière modification, le nombre de l'équipe ou groupe à qui appartient le projet, etc.

— **Containers (Contenants) :**

Le dossier Containers est composé de deux types de contenants qui servent à regrouper plusieurs champs dans le formulaire.

1. Un contenant de type Fieldset crée un contour autour des champs.
2. Un contenant de type HTMLset c'est un éditeur de texte HTML. Le contenant HTMLset s'utilise pour les champs de type de base « texte » que stockent des chaînes de 65, 535 caractères maximum.

— **Details (Détails) :**

Le dossier Détails contient les détails configurés pour les projets, les modèles, les entités, les propriétés et les relations. Quand un détail est ajouté au formulaire, ce détail s'affiche dans la forme d'une grille.

Note : voir annexe pour la signification des champs de l'onglet Propriétés.

---

3. **L'aperçu du formulaire :**

L'espace réservé à l'aperçu du formulaire (côté droit de la fenêtre) permet de prévisualiser les modifications appliquées au formulaire. Les modifications sont affichées en temps réel, cela veut dire que si vous changez l'ordre des champs ou que vous regroupez plusieurs champs dans un fieldset, vous verrez dans l'espace de l'aperçu le résultat final quasi instantanément.

Personnaliser les formulaires

— **Personnaliser un contenant de type Fieldset :**

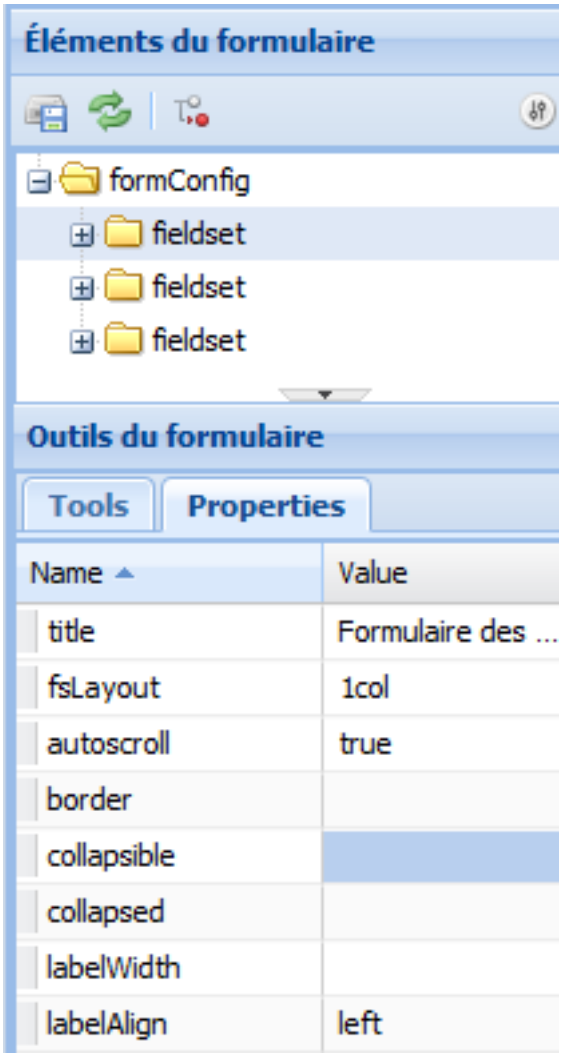


Figure 27 : Personnaliser un contenant de type Fieldset.

1. Sélectionnez le fieldset à personnaliser de l'arborescence.
2. Cliquez sur l'onglet « Properties » de la fenêtre Outils du formulaire.
3. Éditez les valeurs des propriétés en double-cliquant sur les champs pour les modifier. Enregistrez les changements au formulaire en cliquant sur le bouton  Enregistrer formulaire.

— **Ajouter des éléments au formulaire :**

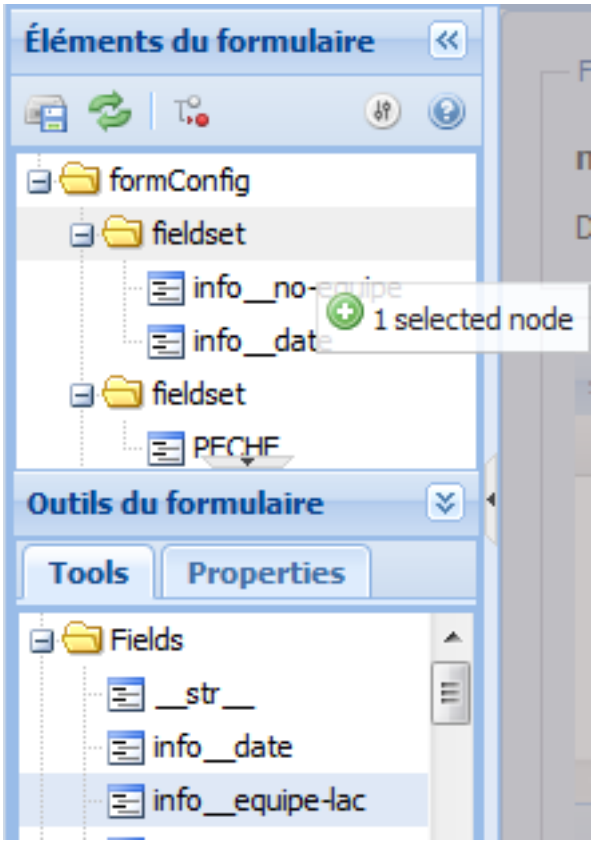


Figure 28 : Ajouter des éléments au formulaire.

1. Ajouter un champ dans un fieldset :

1. Cliquez sur l'onglet Tools de la fenêtre Outils du formulaire.
2. Cliquez sur le plus « + » situé du côté gauche du dossier « Fields » pour visualiser la liste de champs disponibles.
3. Sélectionnez de la liste le champ à ajouter au fieldset.
4. Glissez et déposez le champ à l'intérieur du fieldset.
5. Enregistrez les changements au formulaire en cliquant sur le bouton  .

2. Ajouter un contenant fieldset ou HTMLset :

1. Cliquez sur l'onglet Tools de la fenêtre Outils du formulaire.
2. Cliquez sur le plus « + » situé du côté gauche du dossier « Containers » pour visualiser la liste de contenants disponibles.
3. Sélectionnez de la liste le type de contenant à ajouter à l'arborescence.
4. Glissez et déposez le contenant à l'endroit désiré de l'arborescence. Notez qu'il est possible d'insérer un contenant à l'intérieur d'un autre contenant.
5. Enregistrez les changements au formulaire en cliquant sur le bouton  .

3. Ajouter un détail à l'arborescence :

Les détails prennent automatiquement la forme d'une grille (comme la grille principale de l'application). La grille n'est pas personnalisable à partir de la fenêtre formulaire.

1. Cliquez sur l'onglet Tools de la fenêtre Outils du formulaire.
2. Cliquez sur le plus « + » situé du côté gauche du dossier « Details » pour visualiser la liste de détails disponibles.
3. Sélectionnez de la liste le détail à ajouter à l'arborescence.
4. Glissez et déposez le contenant à l'endroit désiré de l'arborescence.
5. Enregistrez les changements au formulaire en cliquant sur le bouton  .

— **Personnaliser les champs du formulaire :**

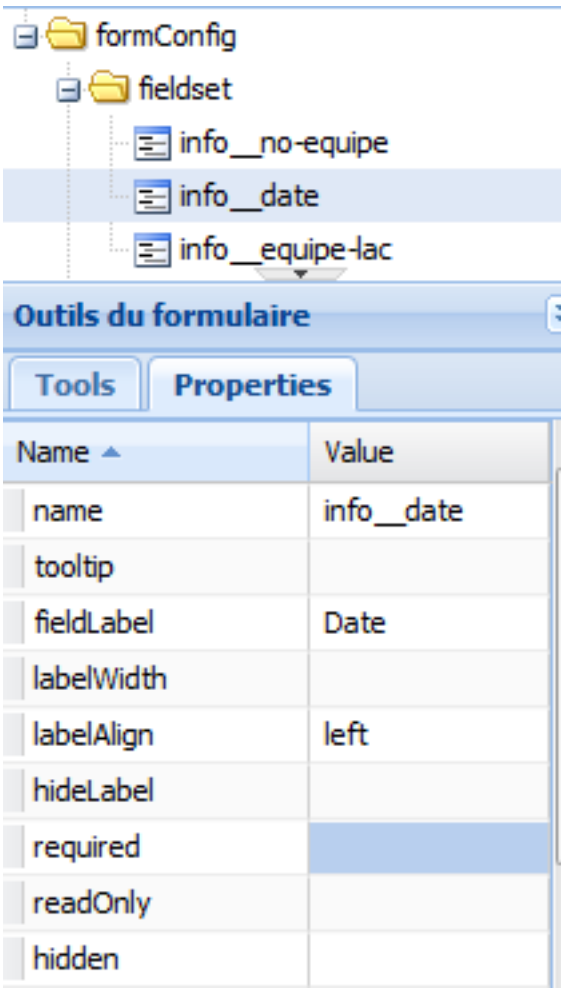


Figure 29 : personnaliser les champs.

1. Sélectionnez le champ à personnaliser.
2. Cliquez sur l'onglet Properties de la fenêtre Outils du formulaire.
3. Éditez les valeurs des propriétés et enregistrez les changements au formulaire en cliquant sur le bouton  .

Note : voir annexe pour la signification des champs de l'onglet Propriétés.

— **Changer l'ordre d'apparition des champs du formulaire :**

1. sélectionnez le champ ;
2. glissez-le jusqu'au l'emplacement désiré dans le même contenant ;
3. vous pouvez déplacer un champ d'un contenant à l'autre de la même façon.

1.2.2 Champs

L'option **champs** permet d'activer/désactiver les champs de la définition de la Méta. Toutes les vues ont un sous-ensemble de champs activés par défaut, mais il est possible de les activer ou les désactiver dans la définition de la Méta. Les absorptions des champs qui se retrouvent dans les entités qui ont une relation avec l'entité principale (qui prend le rôle de parent) se font dans cette même fenêtre. Il est possible aussi d'activer des champs de type gestion comme la date de la dernière modification, la date de création ou modifié par.

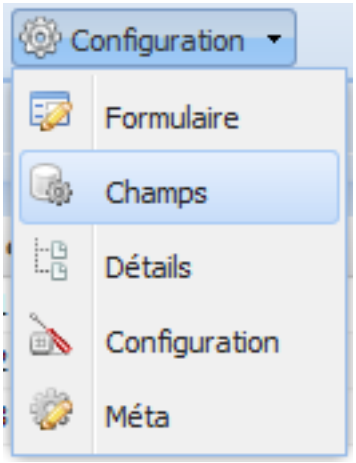


Figure 30 : option champs du menu de configuration.

Activer ou désactiver les champs de la Méta

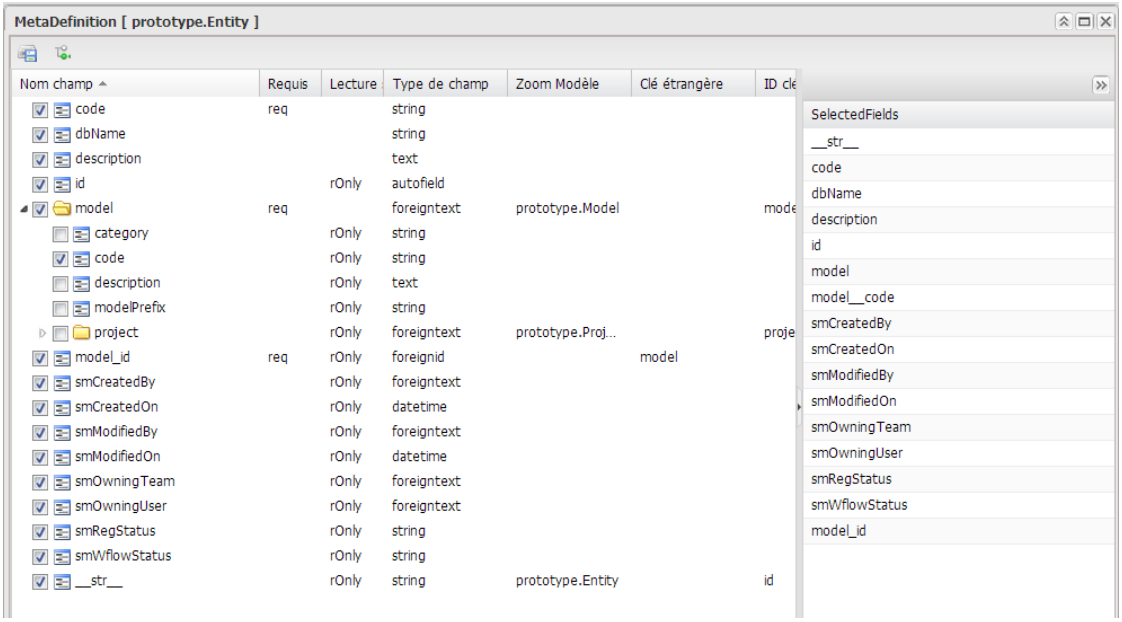


Figure 31 : liste de champs.

— Pour activer ou désactiver les champs, cochez ou décochez la case située à gauche du champ.

— L'icône du dossier représente un autre objet en relation avec l'objet courant et signifie que d'autres champs se trouvent à l'intérieur de ce dossier.
— Pour montrer les champs d'un dossier, cliquez sur le triangle blanc situé à gauche de l'icône dossier.
— Dans la *Figue 31*, la meta pour l'entité contient le nom (code) du modèle à qui elle appartient.
— Enregistrez les changements en cliquant sur le bouton  avant de quitter la fenêtre.
— Vous pouvez ajouter un nouveau champ, en cliquant sur le bouton  en haut du formulaire.
— Cochez ce champ et enregistrer pour l'activer dans la meta.

1.2.3 Détails

L'option **Détails** permet d'activer/désactiver les détails de la définition de la meta. Un détail est un sous-ensemble d'attributs d'une entité enfant.

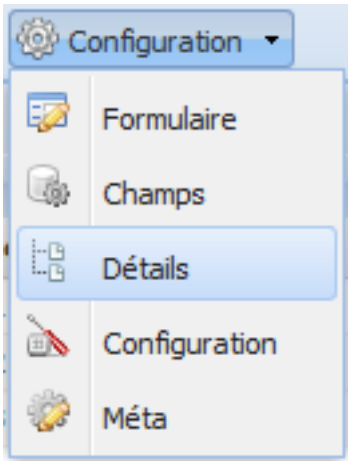


Figure 32 : option détails du menu.

La navigation se fait à partir d'une entité enfant en cliquant sur un des détails disponibles dans l'entité parente.

Dans toutes le vues, tous les détails sont activés par défaut, mais il est possible de les activer ou les désactiver dans la définition de la meta.

Les détails d'une entité sont accessibles à partir de la barre navigation (*Figure 23*).

Figure 33 : détails d'une vue affichés dans la barre de navigation.

Activer ou désactiver les détails de navigation

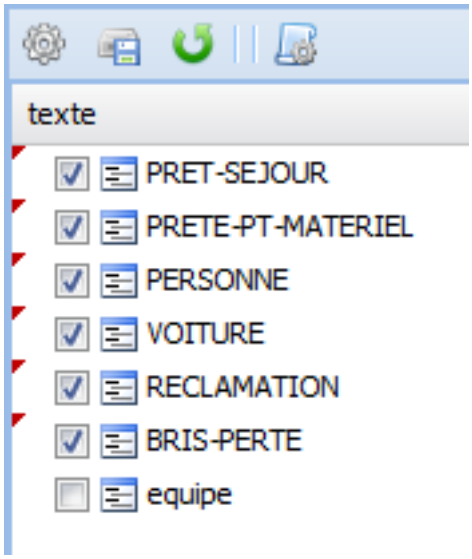


Figure 34 : liste de détails.

Pour activer ou désactiver les détails :

— cochez ou décochez la case située à gauche du détail ;

— enregistrez les changements en cliquant sur le bouton  .

1.2.4 Configuration

L'option **configuration** donne accès au sous-ensemble de propriétés de personnalisation de base. Cette option est visible pour les utilisateurs qui ont des droits d'accès limités à l'application.

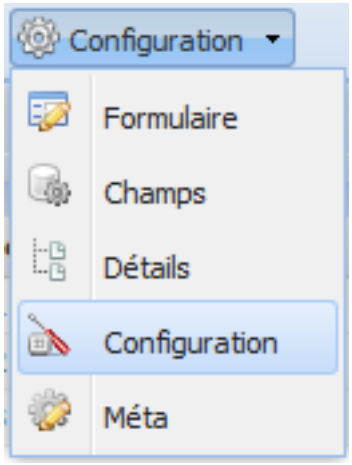


Figure 35 : option configuration du menu.

Afficher ou cacher des colonnes de la grille

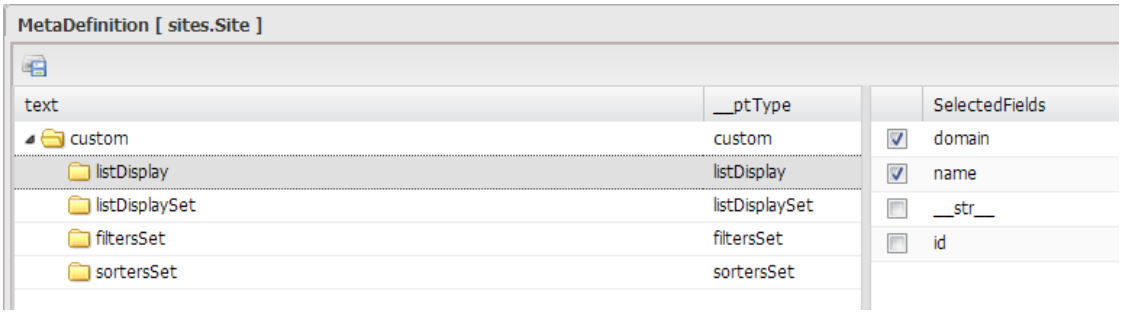


Figure 36 : afficher/cacher les champs

1. Cliquez sur la propriété « listDisplay » de la liste.
2. Pour afficher ou cacher des champs de la grille, cochez ou décochez la case située à gauche du nom du champ.
3. Enregistrez les changements en cliquant sur le bouton  .

Changer l'ordre d'apparition des colonnes sur la grille

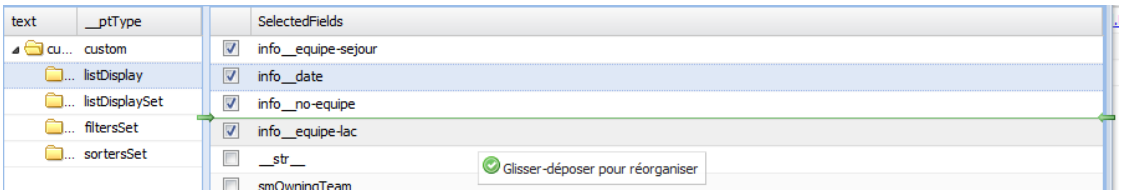


Figure 37 : changer l'ordre d'apparition des champs

1. Cliquez sur la propriété « listDisplay » de la liste.
2. Glissez et déposez le champ vers le haut ou vers le bas.
3. Enregistrez les changements en cliquant sur le bouton  .

Définir différentes configurations de colonnes

ListDisplaySet : à vérifier le fonctionnement.

Définir les critères de recherche pour la fonction rechercher

1. Cliquez sur la propriété « filtersSet » de la liste.
2. Cliquez sur le bouton  pour ajouter un nouveau filtre.
3. Entrez le nom de la colonne.
4. Enregistrez les changements en cliquant sur le bouton  .
5. Pour supprimer un tri sélectionnez-le et cliquez sur le bouton .

Définir les critères de tri pour la fonction classer

1. Cliquez sur la propriété « SortersSet » de la liste.
2. Cliquez sur le bouton  pour ajouter un nouveau tri.
3. Entrez le nom de la colonne.
4. Enregistrez les changements en cliquant sur le bouton  .
5. Pour supprimer un tri sélectionnez-le et cliquez sur le bouton .

1.2.5 Méta

Cette option permet la configuration de la Méta.

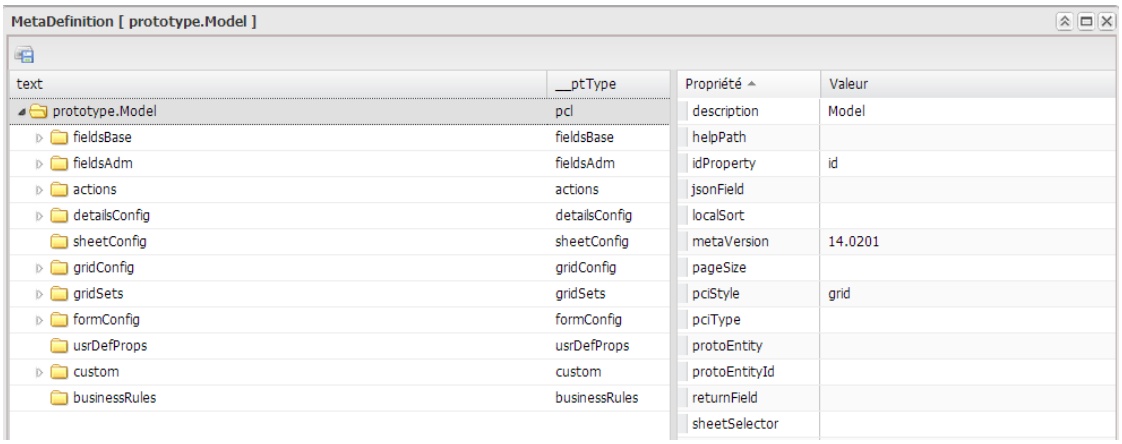


Figure 38 : option Méta du menu.

Si vous cliquez sur une composante de la Méta (côté gauche de la Figure 38), les propriétés s'affichent sur le côté droit.

Configuration de la Méta

Pour changer, la configuration de la Méta :

1. Sélectionnez la Méta (le dossier racine).
2. La liste des propriétés s'affiche à droite.
3. Changer les propriétés voulues, certaines sont non modifiables.
4. Cliquez sur  pour enregistrer.

Note : voir annexe pour la signification des propriétés.

Configuration des champs

Pour changer la configuration d'un champ :

1. Double-cliquez sur **fieldsBase** ou **fieldsAdmin** pour afficher leurs contenus :

— **fieldsBase :** contient les attributs de base de l'entité.
— **fieldsAdm :** contient les clés étrangères de l'entité.

2. Sélectionnez le champ à modifier.
3. La liste des propriétés s'affiche à droite.
4. Changer les propriétés voulues, certaines sont non modifiables.
5. Cliquez sur  pour enregistrer.

Note : voir annexe pour la signification des propriétés.

Configuration des actions

actions : contient les actions définies sur l'objet courant (qui sont les actions de la fonction actions du menu des fonctions).

En cliquant sur cette composante, le bouton d'ajout  apparait, mais pour le moment les actions doivent être définies dans le modèle django.

Double-cliquez sur ce dossier pour afficher la liste des actions de l'objet.

Note : voir annexe pour la signification des propriétés.

Configuration des détails

detailsConfig : liste des objets auxquels on peut naviguer à partir de l'objet courant (voir détails).

Note : voir annexe pour la signification des champs.

Configuration des templates HTML

sheetConfig : permet de définir des templates HTML.

Pour définir un nouveau template :

1. cliquez sur sheetConfig ;
2. cliquez sur  puis entrer le nom du nouveau template ;
3. ou double-cliquez sur sheetConfig pour accéder aux templates définis ;
4. sélectionnez le template ;
5. la liste des propriétés s'affiche à droite ;
6. modifiez le template en modifiant les propriétés ;
7. cliquez sur  pour enregistrer ;
8. pour supprimer un template, Sélectionnez-le et cliquez sur ;

9. cliquez sur  pour enregistrer.

Note : voir annexe pour la signification des propriétés.

Configuration de la grille

gridConfig : à partir de cette option vous pouvez configurer la grille principale.

Cliquez sur **gridConfig** pour accéder à ses propriétés ;

1. la liste des propriétés s'affiche à droite ;
2. changer les propriétés voulues, certaines sont non modifiables ;
3. cliquez sur  pour enregistrer.

Note : voir annexe pour la signification des propriétés.

Double-cliquez sur **gridConfig** pour à accéder aux sous-options :

— **listDisplay :** sert à personnaliser l'affichage et l'ordre d'apparition de champs sur la grille.
— **baseFilter :** permet de définir des filtres de bases qui peuvent être utilisés pour la fonction Rechercher.
— **initialFilter :** permet de définir le filtre initial qui est utilisé pour la grille par défaut.
— **initialSort :** permet de configurer l'ordre d'apparition (ascendante ou descendante) pour un champ de la grille. Quand la grille est chargée, elle affiche son contenu ordonné par le champ configuré avec initialSort. Les valeurs possibles sont : ASC ou DESC.
— **searchFields :** sert à choisir les champs qui seront pris en compte au moment de faire une recherche sur le contenu de la grille. Cette propriété est utilisée avec la propriété « searchable » d'un champ.
— **sortFields :** sert à choisir les champs qui seront pris en compte au moment de faire le classement du contenu de la grille. Cette propriété est utilisée avec la propriété « sortable » d'un champ.
— **hiddenFields :** sélectionnez les champs pour les cacher de la grille. Notez que cette fonctionnalité cache un champ à l'affichage, elle ne l'efface pas de l'application.
— **readOnlyFields :** sélectionnez les champs pour les rendre non modifiables par l'utilisateur.

Pour **listDisplay**, **searchFields**, **sortFields**, **hiddenFields** et **readOnlyFields** :

1. sélectionnez l'une de ces options ;
2. activer ou desactiver les champs ;
3. cliquez sur  pour enregistrer.

Pour **baseFilter**, **initialFilter** et **initialSort** :

1. sélectionnez l'une de ces options ;
2. le bouton  s'affiche en haut à côté du  ;
3. cliquez sur ce bouton et entrez le nom de la colonne à ajouter aux filtres ou aux tris ;
4. cliquez sur  pour enregistrer ;
5. pour supprimer un filtre ou un tri, sélectionner-le et cliquez sur ;
6. cliquez sur  pour enregistrer.

Pour **gridSets :** même rôle que l'option configuration.

Configuration des formulaires

formConfig : personnalise la présentation du formulaire. Le formulaire peut être personnalisé à partir de deux endroits : formConfig dans l'option Méta ou dans l'option formulaire du menu de configuration. On vous recommande de vous servir de l'option formulaire parce qu'elle est plus conviviale pour la personnalisation.

Configuration des propriétés des utilisateurs

usrDefProps : propriétés définies par l'utilisateur.

> **Note :** voir annexe pour la signification des propriétés.

Configuration par les simples utilisateurs

custom : voir l'option configuration.

Configuration des règles d'affaires

businessRules : option non fonctionnelle pour le moment. Elle sera utilisée pour définir les règles d'affaires à partir de l'application.

1.3 Prototypage

1.3.1 Démarche et pratiques du prototypage

La conception du MCD (modèle conceptuel de données) est une phase préliminaire au processus de prototypage. Une fois réalisée, il suffit de suivre les étapes suivantes, pour créer votre prototype :

Enregistrement du modèle

Pour enregistrer un nouveau modèle dans l'application, vous devez le créer ou l'importer.

Importation d'un modèle existant

1. À partir de l'onglet Projet, créer ou sélectionner un projet existant ;
2. Cliquer sur Actions du menu de fonctions ;
3. Choisr une des deux options (doImportSchema ou doImportOMS) selon le modèle à importer.

Création d'un nouveau modèle

Les pages suivantes expliquent étape par étape la procédure de création d'un nouveau modèle :

Créer un projet La première étape pour construire un prototype consiste à créer un nouveau projet qui contiendra le modèle conceptuel de données. Le modèle de données peut aussi être créé à l'intérieur d'un projet existant. Un projet peut contenir un ou plusieurs modèles.

Si vous utilisez un projet existant, vous pouvez passer à l'étape suivante : créer un modèle .

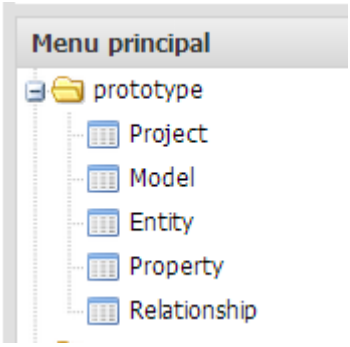


Figure 39 : créer un projet.

1. À partir du menu principal, cliquez deux fois sur Projet dans le dossier Pototype pour ouvrir l'onglet Projet.
2. Cliquez sur le bouton Ajouter.
3. Dans le formulaire ouvert, remplissez :

— le nom du projet (information obligatoire) ;
— la description du projet (information optionnelle).

4. Cliquez sur le bouton Enregistrer de la fenêtre.

Quand vous cliquez sur le bouton Enregistrer, la grille « Modèle filtré par " " » située après le nom du projet affiche les contrôles d'édition (voir figure 40). Il est possible d'ajouter un modèle à partir de cette grille en cliquant sur le bouton Ajouter du menu d'édition.

C'est une de deux façons d'ajouter un modèle dans un projet. Passez à l'étape « créer un modèle » pour continuer avec la démarche.

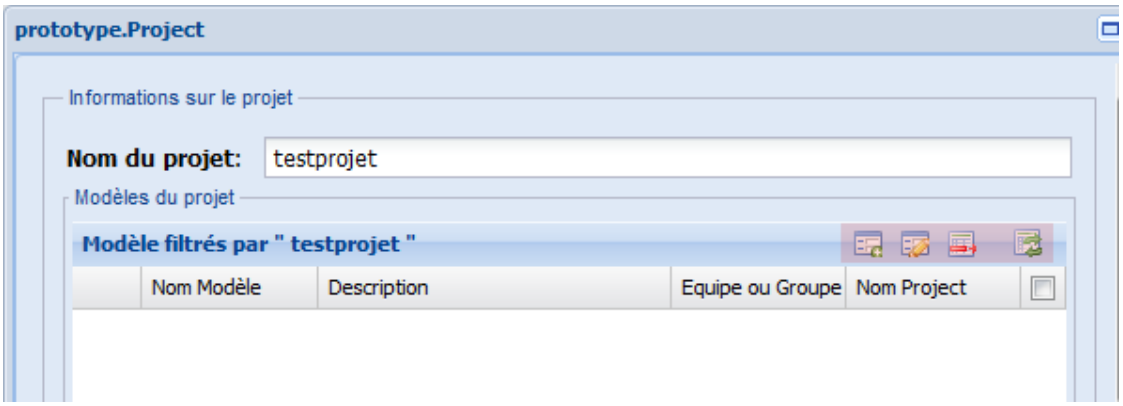


Figure 40 : ajout du modèle à partir du projet.

Créer un modèle La deuxième étape pour construire notre prototype est la création du modèle. Vous pouvez créer un modèle dans un nouveau projet ou le créer à l'intérieur d'un projet existant. Un modèle de données peut être rattaché à un seul projet, mais un projet peut contenir plusieurs modèles.

Il y a deux façons de créer un nouveau modèle que nous allons explorer ci-bas :

Créer un nouveau modèle à partir du menu principal

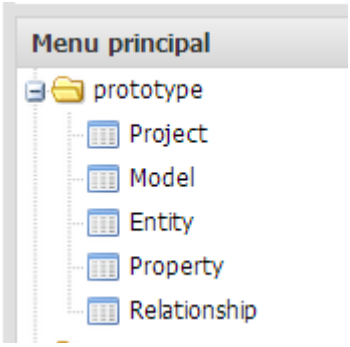


Figure 41 : créer un modèle

1. À partir du menu principal, cliquez deux fois sur Modèle du dossier Prototype pour ouvrir l'onglet Modèle.
2. Cliquez sur le bouton Ajouter, un formulaire sera affiché sur l'écran.
3. Dans le formulaire, remplissez :
   — le nom du modèle (information obligatoire).
   — le nom du projet (information obligatoire). Cliquez sur le bouton  pour sélectionner le projet de la liste.
   — la description du modèle (information optionnelle).
4. Cliquez sur le bouton Enregistrer de la fenêtre.

Créer un nouveau modèle à partir du formulaire du projet

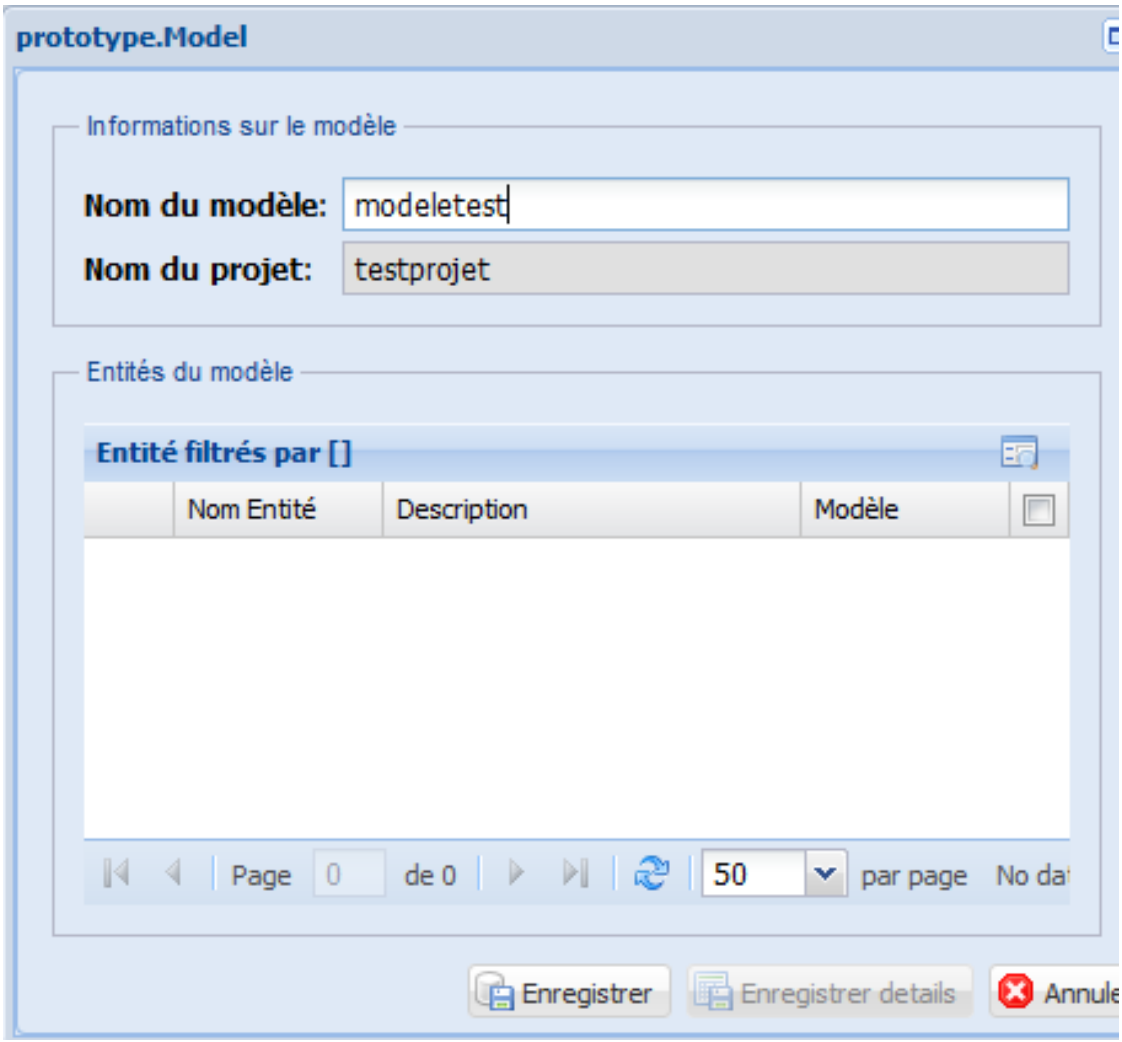


Figure 42 : créer un modèle détails.

1. À partir du formulaire du projet, cliquez sur le bouton Ajouter de la grille « Modèles du projet ».
2. Un nouveau formulaire (comme celui de la figure 42) s'affichera dans l'interface.
3. Dans le formulaire, remplissez :
   — le nom du modèle (information obligatoire)
   — la description du modèle (information optionnelle).
4. Cliquez sur le bouton **Enregistrer** de la fenêtre.

Notez dans les deux façons présentées ici, quand vous cliquez sur le bouton Enregistrer, la grille « Entité filtrés par ” ” » située après le nom du projet affiche les contrôles d'édition (voir figure 43). À partir de cette grille, il est possible d'ajouter des entités en cliquant sur le bouton Ajouter du menu d'édition.

Ceci est une de deux façons d'ajouter des entités dans un projet. Passez à l'étape créer une entité pour continuer avec la démarche.

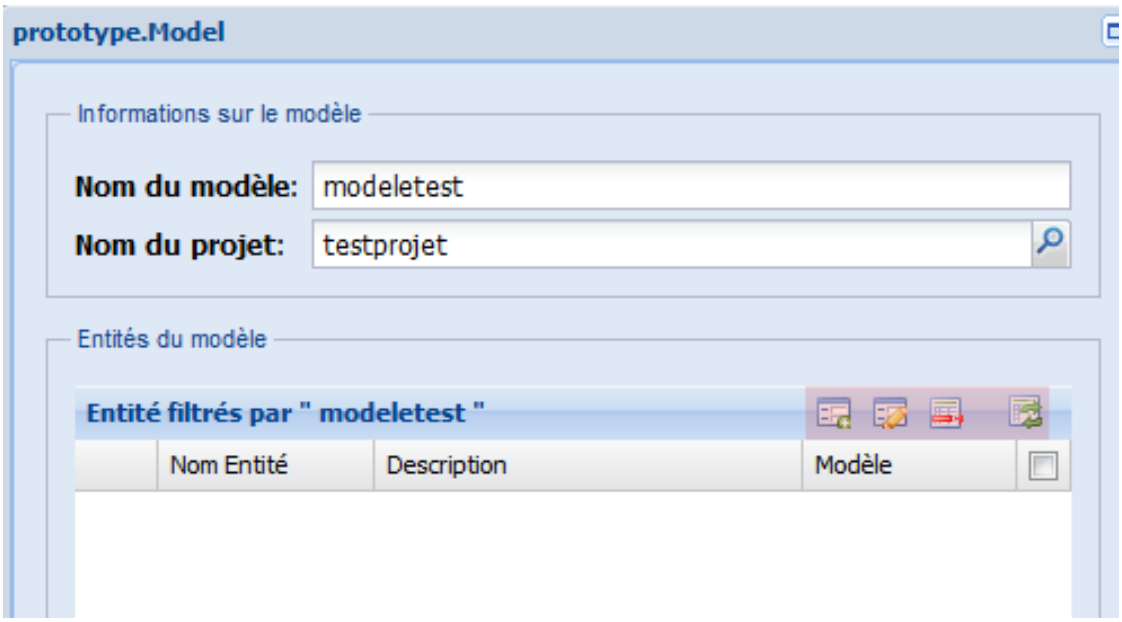


Figure 43 : ajout d'une entité à partir du modèle.

Créer une entité La troisième étape de la construction de notre prototype est la création des entités.

Il y a deux façons de créer les entités d'un modèle que nous allons explorer ci-bas :

Créer une nouvelle entité à partir du menu principal

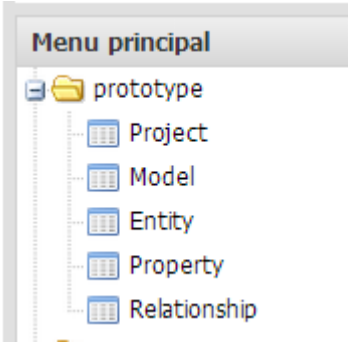


Figure 44 : ajouter une entité

1. À partir du menu principal, cliquez deux fois sur Entité du dossier Prototype, pour ouvrir l'onglet Entité.
2. Cliquez sur le bouton Ajouter, un formulaire sera affiché sur l'écran.
3. Dans le formulaire, remplissez :
   — le nom de l'entité (information obligatoire).
   — le nom du modèle (information obligatoire). Cliquez sur le bouton  pour sélectionner le modèle de la liste.
   — la description de l'entité (information optionnelle).
4. Cliquez sur le bouton Enregistrer de la fenêtre.

Créer une nouvelle entité à partir du formulaire du modèle

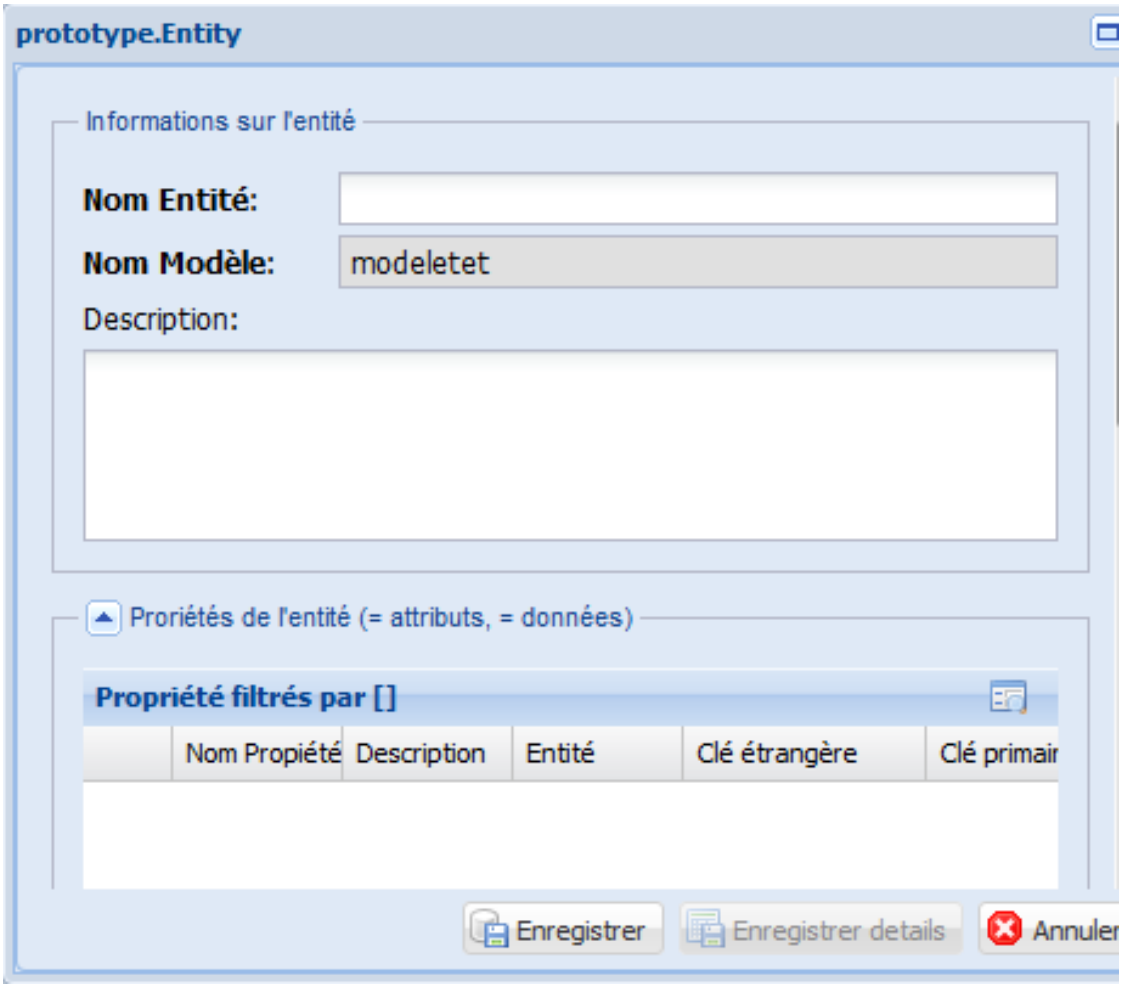


Figure 45 : ajouter une entité détails

1. À partir du formulaire du modèle, cliquez sur le bouton Ajouter de la grille « entités du modèle ».
2. Un nouveau formulaire (comme celui de la figure 45) s'affichera dans l'interface.
3. Dans le formulaire, remplissez :
   — le nom de l'entité (information obligatoire).
   — la description de l'entité (information optionnelle).
4. Cliquez sur le bouton Enregistrer de la fenêtre.

Notez dans les deux façons présentées dans ce texte que quand vous cliquez sur le bouton Enregistrer, la grille « Propriétés filtrés par " " » située après le nom du modèle affiche les contrôles d'édition (voir figure 46). À partir de cette grille, il est possible d'ajouter les propriétés d'une entité en cliquant sur le bouton Ajouter du menu d'édition.

Ceci est une de deux façons d'ajouter les propriétés d'une entité. Passez à l'étape créer une propriété. : pour continuer avec la démarche.

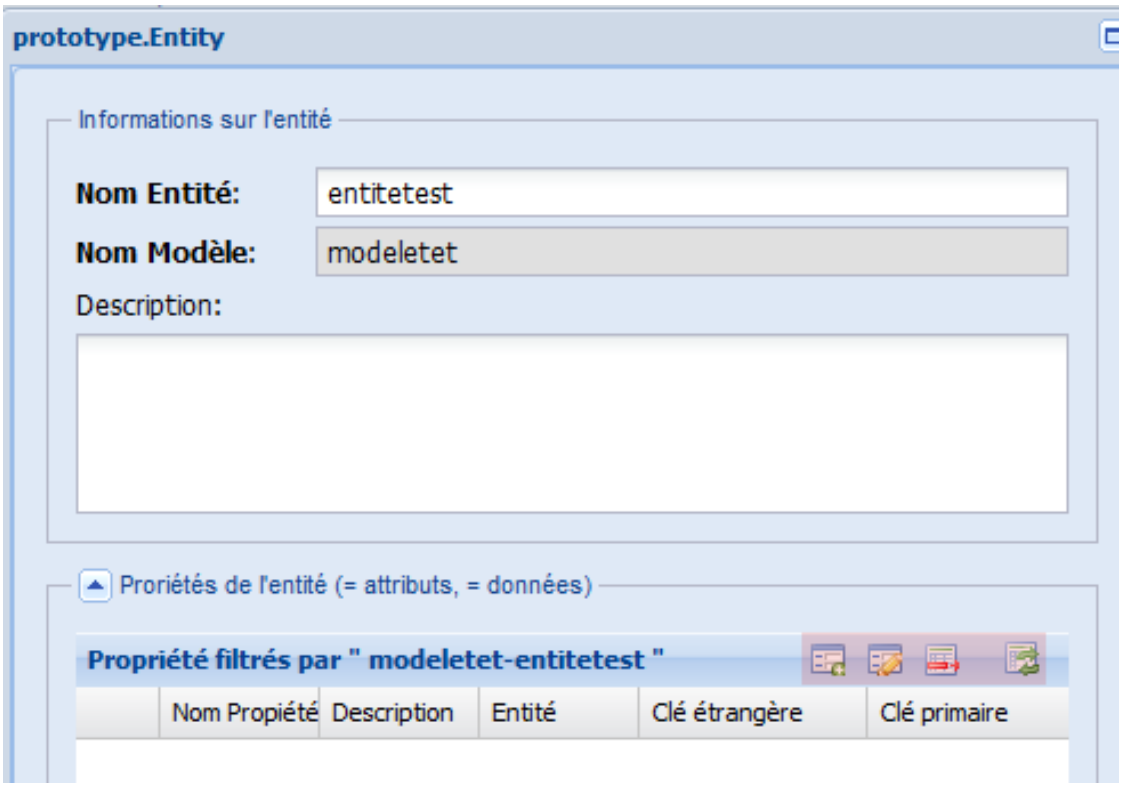


Figure 46 : ajout d'une propriété à partir de l'entité.

Créer une propriété La quatrième étape pour construire notre prototype est la création des propriétés des entités. Il y a deux façons de créer les propriétés d'une entité que nous allons explorer ci-bas :

Créer une nouvelle propriété à partir du menu principal

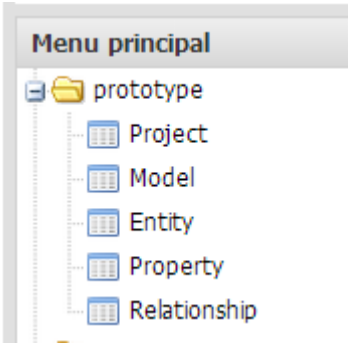


Figure 47 : créer une propriété.

1. À partir du menu principal, cliquez deux fois sur Propriété du dossier Prototype pour ouvrir l'onglet Propriété.
2. Cliquez sur le bouton Ajouter, un formulaire sera affiché sur l'écran.
3. Dans le formulaire, remplissez (voir figure plus bas) :

— le nom de la propriété (information obligatoire) ;

— le nom de l'entité (information obligatoire) ; cliquez sur le bouton  pour sélectionner l'entité de la liste ;
— la description de l'entité (information optionnelle) ;

- cochez la case de la clé primaire (isPrimary) si la propriété est la clé de l'entité. Quand vous cochez cette case, la case Réquis (isRequired) est cochée automatiquement.
  Si la propriété n'est pas une clé primaire, mais que la propriété est obligatoire, cochez la case Réquis (isRequired).
  La case Clé étrangère est cochée automatiquement par l'application et elle est disponible en lecture seulement ;
- sélectionnez le Type de base (baseType, information optionnelle). Le type de base par défaut est string. Si le choix de type de base est décimal, remplissez le champ Décimales.
  Si le choix de type de base est combo, remplissez le champ Valeurs de la liste déroulante et le champ Valeur par défaut ;
- la longueur (prpLength, information optionnelle).
- la valeur par défaut (prpDefault, information optionnelle). Ce champ est utilisé en combinaison avec le type de base combo. Inscrivez dans ce champ la valeur par défaut de votre liste de valeurs. Par exemple, si vous avez une liste de provinces et que votre projet est pour les gens de Québec, inscrivez QC dans la valeur par défaut. Cette valeur sera le premier choix montré à l'usager ;
- les Décimales (prpScale, information optionnelle). Ce champ est utilisé en combinaison avec le type de base décimal. Inscrivez dans ce champ le nombre de positions après la virgule pour un chiffre ;
- les Valeurs liste déroulante (prpChoices, information optionnelle). Ce champ est utilisé en combinaison avec le type de base combo. Inscrivez la liste de valeurs pour une propriété. Les mots doivent être séparés par des virgules sans espaces. Exemple : QC,ON,NS,NB,BC...
- voir les pages Prototype et annexe pour plus d'options sur les propriétés.

4. Cliquez sur le bouton Enregistrer de la fenêtre.

Créer une nouvelle propriété à partir du formulaire de l'entité

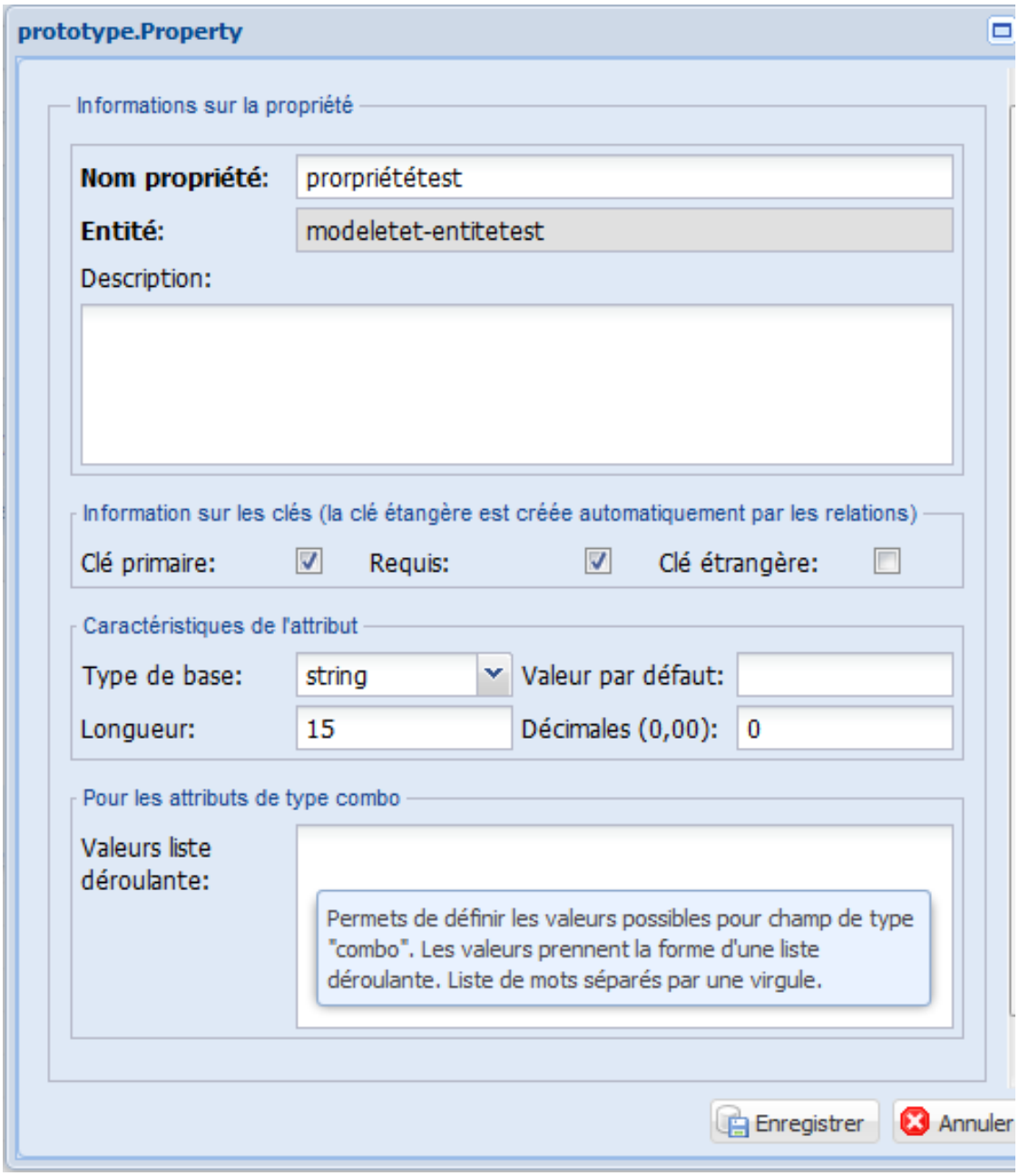


Figure 48 : créer une propriété détails.

1. À partir du formulaire de l'entité, cliquez sur le bouton Ajouter de la grille « propriétés filtrées par " " » ;
2. Un nouveau formulaire (comme celui de la figure à gauche) s'affichera dans l'interface ;
3. Dans le formulaire, remplissez les parties comme dans la partie précédente.
4. Cliquez sur le bouton Enregistrer de la fenêtre.

Notez dans la deuxième façon présentée dans ce texte que quand vous cliquez sur le bouton Enregistrer, la grille « Relation filtrées par " "» située en bas de la grille « Propriétés filtrées par " "» affiche les contrôles d'édition (voir figure 49). À partir de cette grille, il est possible de créer les relations entre les entités en cliquant sur le bouton Ajouter du menu d'édition.

Ceci est une de deux façons de créer les relations entre les entités. Passez à l'étape créer les relations entre les entités pour continuer avec la démarche.

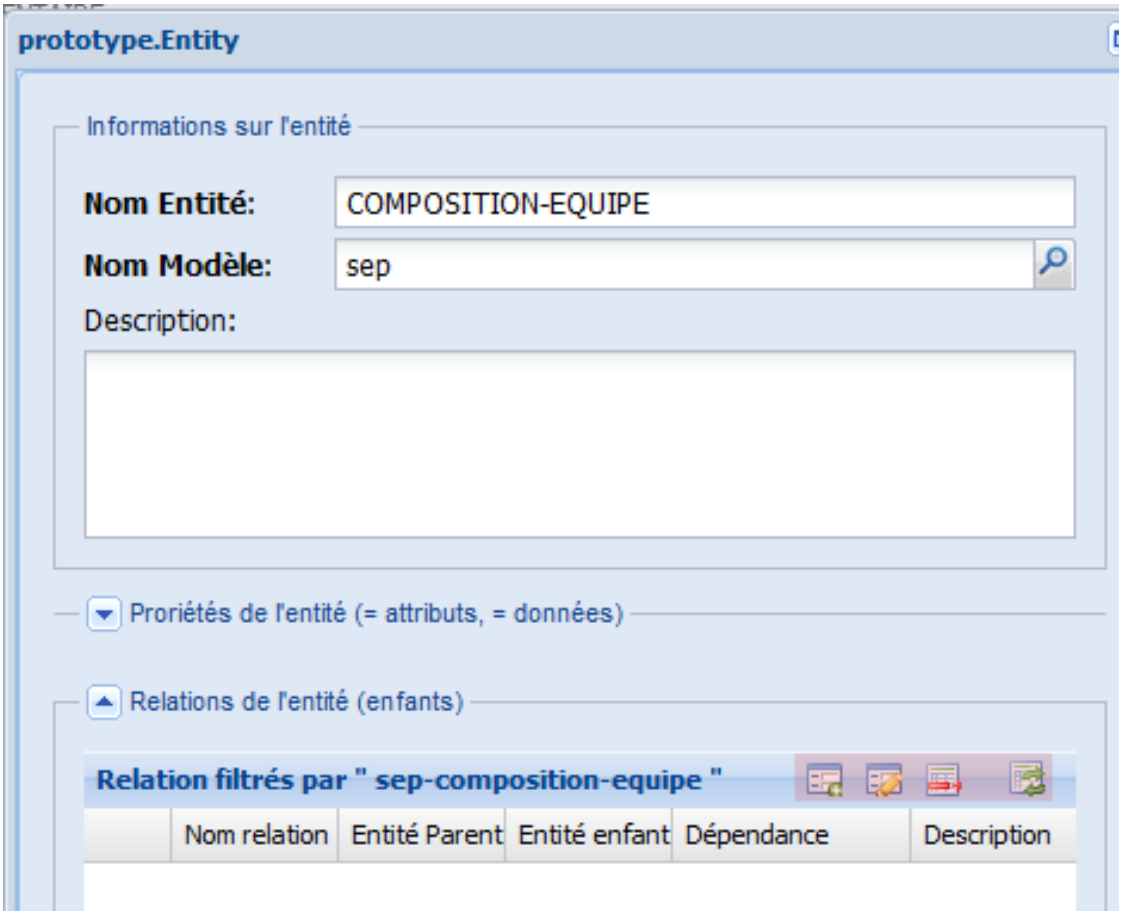


Figure 49 : ajout d'une relation à partir de la propriété.

Créer une relation La cinquième étape pour construire un prototype consiste à créer les relations entre les entités du modèle. Les relations peuvent être créées à partir de l'entité enfant vers l'entité parente ou vice-versa. Les clés étrangères sont générées automatiquement par l'application du prototypeur au moment de créer la relation entre deux entités.

De la même manière que dans les étapes précédentes, il y a deux façons de créer les relations entre les entités.

Créer une nouvelle relation à partir du menu principal

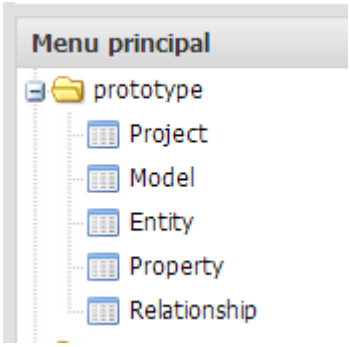


Figure 50 : créer une relation.

1. À partir du menu principal, cliquez deux fois sur Relation du dossier Prototype pour ouvrir l'onglet Relation.

2. Cliquez sur le bouton Ajouter, un formulaire sera affiché sur l'écran.
3. Dans le formulaire, remplissez (voir figure 51) :

— le nom de la relation (information obligatoire). Comme bonne pratique, composez les noms de vos relations en commençant par le nom de l'entité enfant suivi du nom de l'entité parent. Les deux noms séparés par un trait d'union, exemple séjour-chalet.
— le nom de l'entité parent (information obligatoire). Cliquez sur le bouton  pour sélectionner l'entité de la liste.
— le nom de l'entité enfant (information obligatoire). Cliquez sur le bouton  pour sélectionner l'entité de la liste.
— cochez la case de la dépendance de clé si la connectivité du côté enfant est dépendante : 1,1. Si votre connectivité est 1,1 ne cochez pas cette caisse.
— la description de la relation (information optionnelle).

4. Cliquez sur le bouton **Enregistrer** de la fenêtre.

Créer une nouvelle relation à partir du formulaire de l'entité

Figure 51 : créer une relation.

1. À partir du formulaire de l'entité, cliquez sur le bouton Ajouter de la grille « relations filtrées par " " ».
2. Un nouveau formulaire (comme celui de la figure à gauche) s'affichera dans l'interface.
3. Dans le formulaire, remplissez les informations comme dans la partie précédente.
4. Cliquez sur le bouton Enregistrer de la fenêtre.

Maintenant que votre modèle est complet, nous allons voir comment générer le prototype et alimenter le modèle.

Génération du modèle conceptuel graphique

L'application vous permet de visualiser le modèle entré sous forme graphique :

1. À partir de l'onglet **Modèle**, sélectionnez le modèle à générer.
2. Cliquez sur le bouton **Actions** du menu des fonctions.
3. Cliquez sur l'option **doModelGraph** . Un message de confirmation de l'opération sera affiché sur la barre de message.

4. Le modèle graphique généré est ouvert dans un nouvel onglet de votre fureteur sous forme d'un fichier PDF.

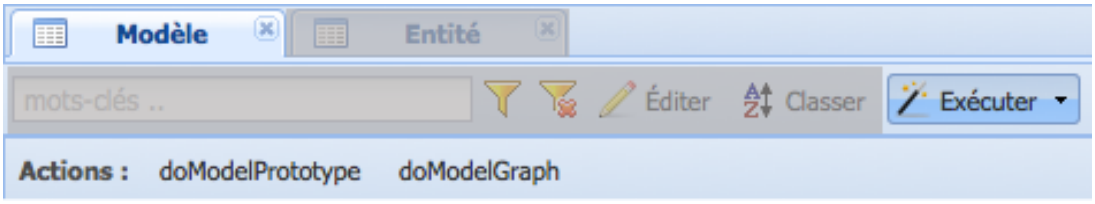


Figure 52 : fonction Exécuter et son sous-menu.

L'action doModelGraph génère le modèle conceptuel graphique pour un modèle sélectionné.

Formalisme du modèle graphique

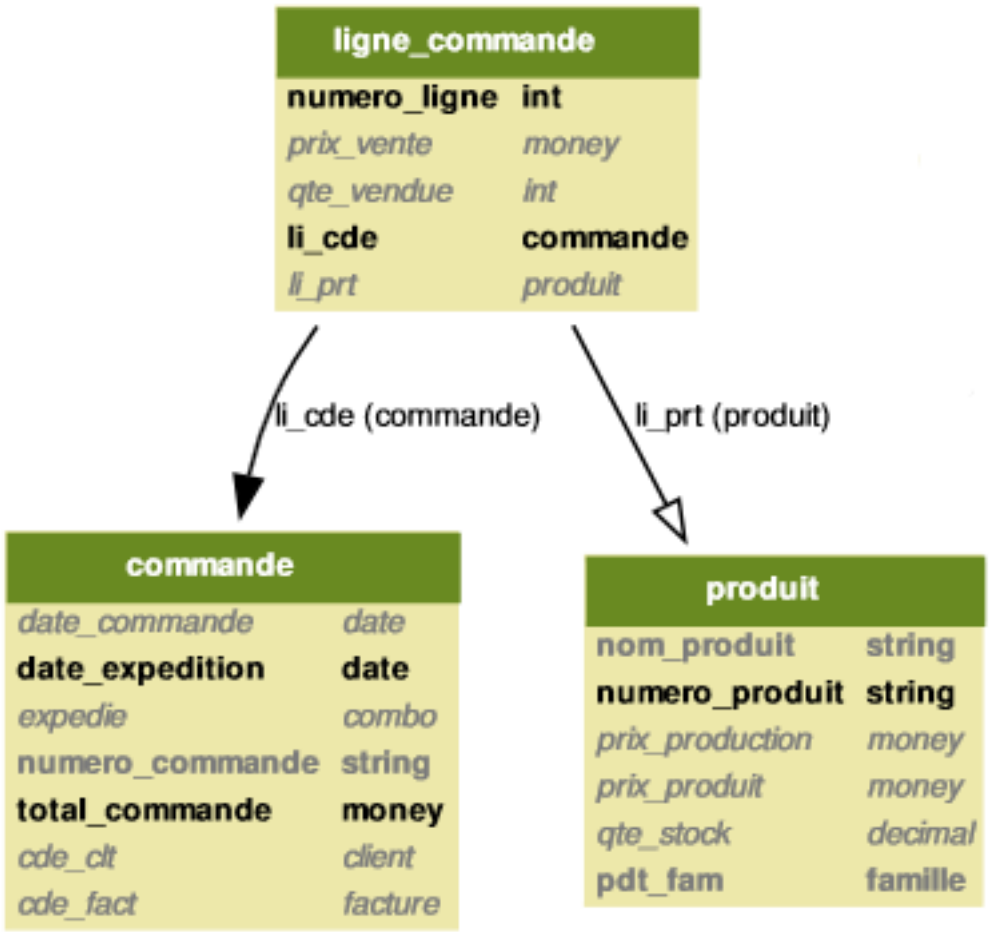


Figure 53 : formalisme du modèle graphique.

— **Clé primaire** : la police du texte est grasse et sa couleur est noire. Exemples dans la figure : numero_ligne, numero_produit et date_expedition.
— **Clé étrangère** : la police du texte est italique et sa couleur est gris pâle. Le nom de son entité enfant apparait du côté droit du nom de la clé (exemple : li_prt -> produit ou cde_fact -> facture).
— **Clé étrangère dépendante (obligatoire)** : la police du texte est grasse et sa couleur est noire. Le nom de son entité enfant apparait du côté droit du nom de la clé (exemple : li_cde -> commande).
— **Clé étrangère requise (obligatoire)** : la police du texte est grasse et sa couleur est gris pâle. Le nom de son entité enfant apparait du côté droit du nom de la clé (exemple : pdt_fam -> famille).

— **Propriété** : la police du texte est en italique et sa couleur est gris pâle (exemples dans la figure : prix_vente, qte_vendue et prix_production).
— **Propriété requise (obligatoire)** : la police du texte est grasse et sa couleur est gris pâle (exemples : nom_produit et numero_commande).
— **Relation dépendante (obligatoire)** : la flèche est pleine.
— **Entité parent** : la flèche pointe vers l'entité parent.

Générer le prototype

Avant de commencer à enregistrer des informations dans votre modèle, il faut générer le prototype correspondant.

Vous pouvez générer un prototype pour un modèle au complet ou pour certaines de ses entités.

Générer un prototype pour un modèle :

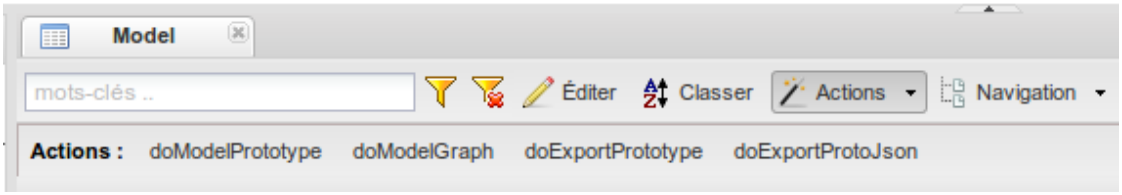


Figure 54 : fonction Actions et son sous-menu pour le modèle.

L'action doModelPrototype génère le prototype pour un modèle sélectionné.

1. À partir de l'onglet **Modèle**, sélectionnez le modèle à générer.
2. Cliquez sur le bouton **Actions** du menu des fonctions.
3. Cliquez sur l'option **doModelPrototype**. Un message de confirmation de l'opération sera affiché sur la barre de message.
4. Réinitialiser le menu principal en cliquant sur le bouton pour faire apparaître les vues récemment créées. Le nouveau prototype est placé dans AutoMenu –> ProtoOptions –> nom_du_projet –> nom_du_modèle.

Note : Les nouvelles vues sont générées automatiquement à l'intérieur de l'arborescence AutoMenu -> ProtoOptions. Notez que les vues des prototypes antérieures sont aussi affichées avec les vues récemment créées. Une vue est créée pour chaque entité dans le modèle de données. Vous aurez le même nombre de vues que des entités. Les noms des vues sont composés du nom du modèle suivi d'un tiret (-) et du nom de l'entité qu'elle représente.

---

5. Les vues dans l'arborescence AutoMenu -> ProtoOptions sont de vues générées automatiquement, si vous effectuez des modifications sur ces vues, la prochaine fois que le menu principal sera rafraîchi, les vues retourneront à leur valeur par défaut. Pour garder les modifications des vues, il est nécessaire de créer un nouveau dossier sur l'arborescence du menu principal. Si vous avez déjà créé un dossier pour le projet, glissez la nouvelle vue à l'intérieur de ce dossier. Dans le cas contraire, passez aux prochaines étapes.
6. Cliquez sur le bouton Nouveau dossier du menu principal. Entrez le nom du nouveau dossier, par exemple le nom du projet. Le nouveau dossier apparaîtra à la fin de la liste des vues générées dans AutoMenu -> ProtoOptions.
7. Procédez à glisser-déposer à l'intérieur de ce dossier chacune de vues appartenant à ce projet. Sélectionnez une vue à la fois.

8. Quand toutes les vues seront à l'intérieur du dossier, glissez et déposez le dossier vers l'arborescence du menu principal. Par exemple, après la composante relation. Assurez-vous que le dossier se retrouve à l'extérieur de l'arborescence AutoMenu.
9. Sélectionnez **AutoMenu** et cliquez sur le bouton Supprimer noeud du menu principal. Ceci effacera l'arborescence d'AutoMenu (la suppression n'est pas définitive, pour récupérer cette arborescence cliquez sur le bouton réinitialiser du menu principal).
10. Pour terminer, cliquez sur le bouton Enregistrer du menu principal.

Avertissement : Si vous n'enregistrez pas les modifications du menu, elles seront perdues à la fermeture de l'application du prototypeur.

Créer ou copier une vue pour une entité (concept) :

Il existe trois façons pour créer une nouvelle vue :

1. **avec l'action doEntityPrototype :**

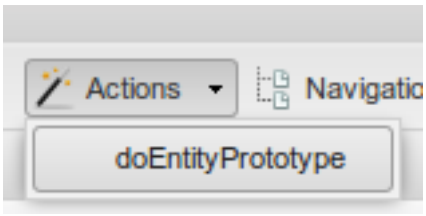


Figure 55 : fonction Actions et son sous-menu pour l'entité.

L'action doEntityPrototype génère le prototype pour une entité sélectionnée.

1. À partir de l'onglet **Entité**, sélectionnez l'entité à générer.
2. Cliquez sur le bouton **Actions** du menu des fonctions.
3. Entrer un nom pour cette vue du prototype ;
4. Cliquez sur l'option **doEntityPrototype**. Un message de confirmation de l'opération sera affiché sur l'interface la barre de message.
5. Réinitialiser le menu pour faire apparaître la vue récemment créée.

Note : Les nouvelles vues sont générées automatiquement à l'intérieur de l'arborescence AutoMenu -> ProtoOptions. Notez que les vues des prototypes antérieures sont aussi affichées avec les vues récemment créées. Une vue est créée pour chaque entité dans le modèle de données. Vous aurez le même nombre de vues que des entités. Les noms des vues sont composés du nom du modèle suivi d'un tiret (-) et du nom de l'entité qu'elle représente.

6. Les vues dans l'arborescence AutoMenu -> ProtoOptions sont de vues générées automatiquement, si vous effectuez des modifications sur ces vues, la prochaine fois que le menu principal sera rafraîchi, les vues retourneront à leur valeur par défaut. Pour garder les modifications des vues, il est nécessaire de créer un nouveau répertoire sur l'arborescence du menu principal. Si vous avez déjà créé un dossier pour le projet, glissez la nouvelle vue à l'intérieur de ce dossier. Dans le cas contraire, passez aux prochaines étapes.
7. Cliquez sur le bouton Nouveau dossier du menu principal. Entrez le nom du nouveau dossier, par exemple le nom du projet. Le nouveau dossier apparaîtra à la fin de la liste des vues générées.
8. Procédez à glisser-déposer à l'intérieur du dossier chacune de vues appartenant à ce projet. Sélectionnez une vue à la fois.

9. Quand toutes les vues seront à l'intérieur du dossier, glissez et déposez le dossier vers l'arborescence du menu principal. Par exemple, après la composante relation. Assurez-vous que le dossier se retrouve à l'extérieur de l'arborescence AutoMenu.
10. Sélectionnez **AutoMenu** et cliquez sur le bouton Supprimer noeud du menu principal. Ceci effacera l'arborescence d'AutoMenu (la suppression n'est pas définitive, pour récupérer cette arborescence cliquez sur le bouton réinitialiser du menu principal).
11. Pour terminer, cliquez sur le bouton Enregistrer du menu principal.

> **Avertissement :** Si vous n'enregistrez pas les modifications du menu, elles seront perdues à la fermeture de l'application du prototypeur.

2. **à partir du Menu principal :**

À partir du menu principal, sélectionnez le dossier où vous voulez placer la nouvelle vue, cliquez sur  pour créer une nouvelle vue.
Remplir les champs avec :
— text : le nom de la vue ;
— option : cherchez l'entité à partir de la liste des options ;
— iconCls : nom de l'icône ;
— qtip : pour créer un tooltip (infobulle).
— qtitle : pour le titre à afficher lorsque la vue est ouverte dans la grille principale.
Les deux premiers champs sont obligatoires, les autres sont optionnels.

3. **par la copie d'une vue existante :**

Vous pouvez copier une vue existante avec ses configurations, à partir du menu des configurations :
— double-cliquez sur la vue à copier pour l'ouvrir dans un onglet ;
— sélectionner Configuration => Méta ;
— cliquez sur le nom de la vue en haut de l'arborescence de la méta pour afficher les options à droite ;
— dans l'option viewCode, entrez le nouveau nom de la vue ;
— cliquez sur le bouton enregistrer ;
— le nom complet sera affiché en haut de la fenêtre de configuration de la Méta (exemple ProtoLib.userFiles.nomDeLaVue) ;
— réinitialiser le menu pour faire apparaître la vue récemment créée ;
— la nouvelle vue sera placée dans un dossier donné par le nom complet dans AutoMenu => ProtoViews. Pour notre exemple, la vue sera placée dans AutoMenu => ProtoViews => ProtoLib => userFiles => ViewCode ;
— déplacer la vue hos dossier AutoMenu avant de quitter et enregistrer pour ne pas perdre vos modifications, car le dossier AutoMenu est généré automatiquement.

Alimenter la base de données :

— Sélectionnez une vue ;
— cliquez sur ajouter ;
— remplir les champs requis et enregistrer.

Le Workflow

De façon pratique, le workflow sert à décrire le circuit de validation, les tâches à répartir entre les différents acteurs d'un processus, les délais, les modes de validation, et à fournir à chacun des acteurs les informations nécessaires à

l'exécution de sa tâche. [1]

Dans notre application, un workflow permet aux administrateurs la validation des ajouts et modifications faits par les utilisateurs.

Activer le Workflow

Pour activer le Workflow sur une entité, il faut que la classe dans le modèle ait un attribut _workflow (non configurable à partir de l'application) ;

Vérifier les Workflows en attente

Deux façons pour vérifier et accepter les workflows en attente de validation :

- La fonction **Filtres** : disponible à partir du menu des fonctions pour les vues qui ont un attribut workflow.
  Vous pouvez accepter ou refuser la modification par le biais des boutons **Accepter** ou **Refuser** de la fonction **Actions** sur le même menu.
- L'action **doWFlowResume** : disponible à partir du menu des fonctions pour les vues Wflow Admin Resume de l'application protoLib.
  Pour cette dernière, vous devez avoir les droits pour l'application protoLib et ajouter un nouvel enregistrement dans la table ParametersBase, avec les paramètres suivants :
  - parameterKey : wflow ;
  - parameterTag : I ;
  - parameterValue : nom_du_modèle.nom_de_l_entité.

L'action doWflowResume permet de voir tous les workflows sur toutes les entités/vues tandis que la fonction Filtres permet de voir seulement les workflow sur la vue ouverte.

1.4 Applications

1.4.1 ProtoLib

Cette application définit les concepts additionnels pour compléter la structure de Django pour les besoins génériques de l'application, dont, la sécurité. Elle permet de gérer les propriétés dynamiques du prototypage et offre les services de soutien à la gestion générique des vues.

TeamHierarchy

La classe TeamHierachy définit la hiérarchie des groupes des utilisateurs.

Les attributs de cette classe sont :

- code : nom du groupe (clé primaire) ;
- description : une description du groupe (optionnelle) ;
- parentNode : clé étrangère vers le groupe parent du nouveau groupe (non obligatoire) ;
- site : site du groupe (optionnel).

UserProfile

La classe UserProfile définit le profil des utilisateurs.

Les attributs de cette classe sont :

1. wikipedia

— user : clé étrangère obligatoire vers un utilisateur (user de django)
— userTeam : clé étrangère vers un groupe de TeamHierarchy (optionnelle) ;
— langage : par défaut la langue de l'application est le français. Les langues disponibles sont :français : fr, anglais : en, espagnol :es.
— userTree : les équipes auxquelles appartient l'utilisateur.

UserShare

La classe UserShare permet de partager les droits d'un utilisateur.

Les attributs de cette classe sont :
— user : clé étrangère obligatoire vers un utilisateur ;
— userTeam : clé étrangère vers un groupe de TeamHierarchy.

ProtoModel

Cette classe est une classe abstraite, elle implantée par plusieurs autres classes. Ses attributs seront automatiquement mis à jour par l'application.

Les attributs de cette classe sont :
— smOwningUser : utilisateur propriétaire de l'objet ;
— smOwningTeam : le groupe auquel appartient l'utilisateur ;
— smCreatedBy : utilisateur qui a créé l'objet ;
— smModifiedBy : le dernier utilisateur qui a modifié l'objet ;
— smRegStatus :
— smWflowStatus : statut de workflow (I : à verifier, OK : accepté, R : refusé) ;
— smCreatedOn : date de création de l'objet ;
— smModifiedOn : date de la dernière modification de l'objet ;
— _protoObj :

ProtoDefinition

Cette table stocke la définition des PCIs et des menus pour chaque vue :

Les attributs de cette classe sont :
— code : nom de l'objet (obligatoire) ;
— description : description (optionnel) ;
— metaDefinition : contient la définition de la Méta ;
— active : booléen qui indique si l'objet est actif ou non ;
— overWrite : booléen qui était utilisé pour des fins de tests (non utile actuellement).

CustomDefinition

Cette classe hérite de la classe ProtoModel. Elle contient la définition du menu pour chaque groupe.

Les attributs de cette classe sont :
— code : nom de l'objet (obligatoire) ;
— description : description (optionnel) ;
— metaDefinition : contient la définition de la Méta ;
— active : booléen qui indique si l'objet est actif ou non ;
— overWrite : booléen qui était utilisé pour des fins de tests (non utile actuellement).

UserFiles

La classe UserFile sert à définir les documents des utilisateurs (fonctionnement pas encore testé).

Les attributs de cette classe sont :
- docfile : champs de type Field qui définit le fichier.

DiscreteValue

La classe DiscretValue permet de définir des valeurs discrètes où la définition d'une table n'est pas vraiment utile.

Les attributs de cette classe sont :
- code : la valeur de l'objet (obligatoire) ;
- value : la valeur de l'objet (obligatoire) ;
- description : description (optionnelle)
- title : clé étrangère vers un autre objet DiscreteValue.

ParametersBase

Cette classe hérite de la classe ProtoModel, elle sert à configurer des paramètres de Bases. Pour le moment, elle permet seulement l'activation de workflow pour les entités qui ont cet attribut.

Pour activer le workflow, initialiser les paramètres avec :
- parameterKey : wflow
- parameterTag : I
- parameterValue : nom_du_modèle.nom_de_l_entité

PtFunction

Cette classe sera utilisée pour créer des fonctions à partir de l'application. Ce qui permettra de définir les règles d'affaires à partir de l'application (pas encore fonctionnelle).

Les attributs de cette classe sont :
- code : nom de la fonction obligatoire ;
- modelName : nom du modèle obligatoire ;
- arguments : liste des paramètres séparés par des virgules ;
- functionBody : corps de la fonction ;
- tag : le tag de la fonction ;
- description : description de la fonction (optionnelle).

WflowAdminResume

Cette classe hérite de la classe ProtoModel. Elle contient le dernier résumé des nouvelles qui requièrent une action de l'administrateur.

Les attributs de cette classe sont :
- viewEntity : nom de l'entité sur laquelle le workflow est activé ;
- activityCount : nombre de workflows en attente sur l'entité (rempli automatiquement).

WflowUserReponse

Cette classe hérite de la classe ProtoModel. Elle contient les résultats des actions de l'administrateur :

Les attributs de cette classe sont :
- viewEntity : nom de l'entité sur laquelle le workflow est activé ;
- wfAction : action (accepter ou refuser) ;
- strKey : la raison de refus ;
- adminMsg : message de l'administrateur qui sera envoyé à l'utilisateur.

1.4.2 Prototype

Cette application définit les concepts pour créer un prototype.

Project

La classe qui représente un projet, elle hérite les propriétés de la classe ProtoModel dans le modèle ProtoLib. Ses attributs sont :
- code : nom du projet obligatoire ;
- description : optionnelle.

Les attributs ci-dessous sont optionnels et définit les paramètres pour la base de données du projet :
- dbEngine : type de la base de données ;
- dbName : nom de la base de données ;
- dbUser : nom d'utilisateur de la BD ;
- dbPassword : mot de passe pour la BD ;
- dbHost : nom de l'hôte ;
- dbPort : numéro de port.

Model

La classe qui représente un modèle, elle hérite les propriétés de la classe ProtoModel dans le modèle ProtoLib. Ses attributs sont :
- project : nom du projet auquel appartient le modèle ;
- code : nom du modèle (obligatoire) ;
- category : catégorie du modèle (optionnelle) ;
- modelPrefix : préfixe à ajouter au nom du modèle (optionnel) ;
- description : optionnelle.

Entity

La classe qui représente une entité (concept), elle hérite les propriétés de la classe ProtoModel dans le modèle ProtoLib. Ses attributs sont :
- model : nom du modèle auquel appartient l'entité
- code : nom de l'entité ;
- dbName : nom de l'entité dans la BD ;
- description : optionnelle.

Property

La classe qui représente les propriétés d'une entité, elle hérite les propriétés de la classe ProtoModel dans le modèle ProtoLib. Ses attributs sont :

- entity : nom du modèle à laquelle appartient l'entité ;
- code : nom de la propriété ;
- baseType : type de base de la propriété. Par défaut c'est string ;
- prpLength : pour définir une longueur pour le champ ;
- prpScale : nombre de positions après la virgule pour un chiffre ;
- vType : type de validation ;
- prpDefault : valeur par défaut (peut varier selon les cas) ;
- prpChoices : liste de choix pour les propriétés dont le baseType est combo ;
- isSensitive : booléen qui indique si oui ou non la propriété requiert un niveau de sécurité plus élevé ;
- description : description optionnelle ;
- notes :
- isPrimary : booléen qui indique si oui ou non la propriété est une clé primaire ;
- isLookUpResult : booléen qui indique si oui ou non la propriété est visible dans les résultats de la fonction rechercher ;
- isNullable : booléen qui indique que la propriété peut être nulle ou non ;
- isRequired : booléen qui indique que la propriété est requise ou non ;
- isReadOnly : booléen qui indique que la propriété est en lecture seule (non modifiable à partir de l'application) ;
- isEssential : booléen qui indique si la propriété doit sortir dans la vue ;
- isForeign : booléen qui indique si oui ou non la propriété est une clé étrangère ;
- crudType : liste de choix pour type d'édition (editable, screenOnly, storeOnly, etc...) ;
- dbName : nom de la propriété dans la BD.

Relationship

Cette classe hérite les propriétés de la classe Property. Elle définit les relations entre les concepts (entités). Ses attributs sont :

- refEntity : nom de l'entité enfant ;
- relatedName : nom de la relation ;
- baseMin : cardinalité (baseMin >= refMin) ;
- baseMax : cardinalité (baseMax <= refMax) ;
- refMin : référence minimale pour la relation (0 : pour une relation 0 N) ;
- refMax : référence maximale pour la relation (N : pour une relation 0 N) ;
- onRefDelete :
- typeRelation :

PropertyEquivalence

Cette classe indique l'équivalence entre deux propriétés. Elle hérite les propriétés de la classe ProtoModel. Ses attributs sont :

- sourceProperty : la première propriété (obligatoire).
- targetProperty : la deuxième propriété (obligatoire).
- description : description (optionnelle).

Prototype

Cette classe hérite les propriétés de la classe ProtoModel dans le modèle ProtoLib. Ses attributs sont :

- entity : nom de l'entité ;
- code : nom du prototype ;

— description : description optionnelle ;
— notes :
— metaDefinition : la Méta.

ProtoTable

Cette classe hérite les propriétés de la classe ProtoModel dans le modèle ProtoLib. Ses attributs sont :
— entity : nom de l'entité ;
— info :
— objects :

Diagram

Cette classe définit les diagrammes d'un projet, elle hérite les propriétés de la classe ProtoModel dans le modèle ProtoLib. Ses attributs sont :
— project : clé étrangère du projet représenté par le diagramme ;
— code : nom du diagramme (obligatoire) ;
— description : description (optionnelle) ;
— notes : notes (optionnelle) ;
— title : titre du diagramme (optionnelle) ;
— prefix :
— graphLevel : entier qui indique le niveau de représentation ('all', 'essential', 'required' , 'primary', 'title', etc.)
— grphMode : entier qui indique le mode de représentation (record, htmlTable, graph) ;
— graphForm : entier qui indique le formalisme de représentation (ObjetRealational, ER, DataRun) ;
— showPrpType : si vrai, affiche le type de la propriété ;
— showBorder : si vrai, affiche le borders ;
— showFKey : affiche les clés étrangères.
— info :
— objects :

DiagramEntity

Cette classe définit les diagrammes d'une entité, elle hérite les propriétés de la classe ProtoModel dans le modèle ProtoLib. Ses attributs sont :
— diagram : nom du diagramme auquel appartient le diagramme DigramEntity.
— entity : entité représentée par le diagramme ;
— info :
— objects :

1.5 Annexes

1.5.1 Index des champs

Champs :	Signification :
autoscroll :	si vrai, défilement automatique du formulaire.
afterCellUpdate :	règle d'affaires à exécuter après la mise à jour d'une cellule.
afterRowDelete :	règle d'affaires à exécuter après la suppression d'une ligne.

Tableau 1.1

Champs :	Signification :
afterSave :	règle d'affaires à exécuter après une sauvegarde.
border :	bordure du formulaire en pixels.
beforeCellEdit :	règle d'affaires à exécuter avant la suppression d'une cellule.
beforeCellUpdate :	règle d'affaires à exécuter avant la mise à jour d'une cellule.
beforeOpSave :	règle d'affaires avant l'enregistrement de l'opération.
beforeRowDelelte :	règle d'affaires à exécuter avant la suppression d'une ligne.
beforeRowInsert :	règle d'affaires à exécuter avant l'ajout d'une ligne.
beforeValidate :	règle d'affaires avant la validation de l'opération.
collapsible :	transforme le contenant en un contenant pliable/dépliable. Un bouton avec un triangle noir apparaitra dans la p
collapsed :	spécifie si à l'ouverture du formulaire le contenant sera par défaut plié ou déplié.
cellLink :	la cellule est un lien (True, False).
choices :	liste de choix possible pour la propriété.
conceptDetail :	le nom de l'entité détail.
cpFromField :	
cpFromZoom :	
crudType :	liste de choix pour la propriété (crud : create, read, update, delete).
dblClick :	règle d'affaires pour le double clique.
description :	courte description.
detailField :	
detailName :	nom de l'attribut dans le modèle.
detailTitleField :	
detailTitleLabel :	
detailTitleLabel :	booléen qui sert à activer ou désactiver la fonctionnalité d'exportation en format CSV.
fieldLabel :	étiquette du champ dans le formulaire ou la grille.
fkField :	
fkId :	
flex :	valeur numérique (pourcentage %) que définit la largeur des colonnes de la grille principale. Flex fait un calcu
Format :	
fsLayout :	structure du formulaire (fluid, 1, 2 ou 3 colonnes).
getLinkFilter :	règle d'affaires pour récupérer une requête.
header :	l'entête de cette propriété dans la grille principale.
height :	hauteur en pixels.
hidden :	propriété cachée.
hideLabel :	si vrai, les étiquettes seront cachées du formulaire.
hideRowNumbers :	booléen qui sert à afficher ou cacher les numéros de lignes sur la grille.
helpPath :	un lien (par exemple vers un wiki sur la propriété).
idProperty :	propriété qui est la clé primaire (souvent c'est le id).
issRowLoad :	règle d'affaires lors d'un chargement d'une instance.
jsonField :	definit un champ au format JSON.
localSort :	
labelAlign :	alignement des étiquettes dans le formulaire.
labelWidth :	largeur des étiquettes dans le formulaire.
metaversion :	la version de la Méta (non modifiable à partir de l'application).
masterField :	
masterTitleField :	
menuText :	le titre dans l'entête de la grille.
maxWidth :	largeur maximale en pixels.
maxHeight :	hauteur maximale en pixels.
minWidth :	largeur minimale en pixels.

Tableau 1.1

Champs :	Signification :
minHeight :	hauteur minimale en pixels.
name :	nom de la propriété dans le modèle (non modifiable).
pageSize :	nombre de résultats par page par défaut c'est 50.
pciStyle :	le style de la PCI (grid pour grille, form pour formulaire et tree pour le menu principal)
pciType :	
protoEntity :	le nom de l'entité dans la table protoTable.
protoEntityId :	l'id de l'entité dans la table protoTable.
physicalName :	
primary :	la propriété est une clé primaire.
prpDefault :	valeur par défaut.
prpLength :	longueur du champ.
prpScale :	nombre de positions après la virgule pour un chiffre.
returnField :	champ non utilisé à effacer de l'interface ;
readOnly :	si vrai, la propriété non modifiable par l'utilisateur.
required :	si vrai, la propriété est obligatoire.
reposition :	règle d'affaires pour le déplacement des instances dans la grille principale.
sheetSelector :	le sheet (template) à utiliser.
shortTitle :	titre à afficher sur la grille.
updateTime :	date de la dernière mise à jour.
userVersion :	si l'utilisateur veut définir des versions pour ses prototypes sur application.
viewCode :	nom de la vue. Si vide vue principale.
viewEntity :	nom de la classe dans le modèle (non modifiable à partir de l'application).
viewIcon :	nom de l'icône utilisé.
searchable :	si oui, les valeurs de cette propriété peut être recherchée (à partir de la fonction rechercher).
sortable :	si oui, la grille principale peut être triée par ce champ (à partir de la fonction classer).
title :	ajoute un titre au contour du fieldset.
tooltip :	infobulle : chaîne de caractères qui crée un message dans la forme d'une infobulle quand la souris passe sur le
type :	type du champ.
validationComplete :	règle d'affaires pour la validation complète des modifications.
vType :	type de validation pour le type du champ.
wordWrap :	valeur booléenne qui permet de visualiser le contenu du champ en plusieurs lignes. Propriété utilisée avec les
xtype :	
zoomFilter :	
zoomModel :	
zoomMultiple :	
zoomConfigure :	
zoomreturn :	
__ptType :	type de l'objet (PCI, fieldsBase, fieldAdm, etc...)