Étude Technique

Enregistrement des requêtes OGAM

|  |  |
| --- | --- |
| Résumé | Analyse technique des besoins d'enregistrement des requêtes OGAM |
| Mots-clés | OGAM, Métamodèle, OGAM Bureau |
| Confidentialité | Confidentiel IGN - MNHN |

|  |  |  |
| --- | --- | --- |
| Documents de référence | | |
| Étude technique de l'évolution majeure du produit OGAM | Philippe Dumoulin | SIDT/2015.025 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Nom - Fonction | Service | Date | Signature |
| Rédacteur | Sylvain GALOPIN  Analyste OGAM | D2SI/SIDT | 24/03/2017 |  |
| Rédacteur | Anna Mouget  Développeur Ginco | D2SI/SAI | 16/03/2017 |  |
| Vérificateur | Séverine Candelier  Chef de projet Ginco | D2SI/SAI | 21/03/2017 |  |
| Approbateur | Philippe DUMOULIN  Chef du produit OGAM | D2SI/SIDT | 24/03/2017 |  |

***Versions du document***

|  |  |  |
| --- | --- | --- |
| Révision | Date | Commentaire |
| 1 | 15/03/2017 | Création du document par Sylvain Galopin |
| 2 | 16/03/2017 | Modification du document par Anna Mouget |
| 3 | 17/03/2017 | Modification du document par Sylvain Galopin |
| 4 | 21/03/2017 | Modification du document par Séverine Candelier |
| 5 | 24/03/2017 | Modification du document par Philippe Dumoulin et Sylvain Galopin |

***Comité de lecture***

|  |  |  |  |
| --- | --- | --- | --- |
| Nom | Unité | Nom | Unité |
| Philippe Dumoulin | D2SI/SIDT/IDF | Anna Mouget | D2SI/SAI/ISD |
| Benoit Pesty | D2SI/SIDT/IDF | Séverine Candelier | D2SI/SAI/ISD |
| Florent Bourcier | D2SI/SIDT/IDF | Gautam Pastakia | D2SI/SAI/ISD |
| Judith Panijet | MNHN |  |  |

***Liste de diffusion***

|  |  |  |  |
| --- | --- | --- | --- |
| Nom | Unité | Nom | Unité |
| Philippe Dumoulin | D2SI/SIDT/IDF | Anna Mouget | D2SI/SAI/ISD |
| Benoit Pesty | D2SI/SIDT/IDF | Séverine Candelier | D2SI/SAI/ISD |
| Florent Bourcier | D2SI/SIDT/IDF | Gautam Pastakia | D2SI/SAI/ISD |
| Cécile Borges-Lemarié | D2SI/SIDT/IDA | Laurent Quene | D2SI/SIDT/IDA |
| Judith Panijet | MNHN | Laurent Delgado | D2SI/SIDT/ISD |

Table des matières

[1 Introduction 6](#_Toc478372784)

[1.1 Cadre de l'étude 6](#_Toc478372785)

[1.2 Contexte 6](#_Toc478372786)

[1.3 Objectif du document 6](#_Toc478372787)

[2 Analyse de l'existant 7](#_Toc478372788)

[2.1 Côté client et serveur 7](#_Toc478372789)

[2.2 SGBD et architecture des données 7](#_Toc478372790)

[3 Analyse technique de la fonction d'enregistrement des requêtes 8](#_Toc478372791)

[3.1 Interface homme-machine 8](#_Toc478372792)

[3.1.1 Enregistrement d'une requête 8](#_Toc478372793)

[3.1.2 Gestion des requêtes enregistrées 10](#_Toc478372794)

[3.1.3 Gestion des groupes 12](#_Toc478372795)

[3.1.4 Affichage des requêtes enregistrées 14](#_Toc478372796)

[3.2 Interface client/serveur 15](#_Toc478372797)

[3.2.1 Les nouvelles requêtes 15](#_Toc478372798)

[3.2.2 Mise à jour des requêtes existantes 16](#_Toc478372799)

[3.3 Côté Serveur 17](#_Toc478372800)

[3.4 Interface serveur/SGBD 17](#_Toc478372801)

[3.5 Données 17](#_Toc478372802)

[3.6 Droits 18](#_Toc478372803)

[4 Mise en œuvre des développements 19](#_Toc478372804)

[4.1 Phase 1 – Enregistrement des requêtes 19](#_Toc478372805)

[4.2 Phase 2 – Modification de l'affichage des requêtes enregistrées 19](#_Toc478372806)

[4.3 Phase 3 – Gestion des groupes 19](#_Toc478372807)

[4.4 Recettes des développements 19](#_Toc478372808)

[4.4.1 Jeux de données de tests 19](#_Toc478372809)

[4.4.2 Recettes des phases par les développeurs 20](#_Toc478372810)

[4.4.3 Recettes des phases par le chef de produit 20](#_Toc478372811)

**Table des illustrations**

[Figure 1 : Modèle de données actuel des requêtes prédéfinies 7](#_Toc478372812)

[Figure 2 : Formulaire de création d'une requête enregistrée 8](file:///C:\Users\sgalopin\Desktop\Etude%20Technique%20-%20Enregistrement%20des%20requêtes%20OGAM_AM_SG_SC_PDU.docx#_Toc478372813)

[Figure 3 : Formulaire de modification d'une requête enregistrée 9](#_Toc478372814)

[Figure 4 : Tableau des requêtes enregistrées classées par groupes 10](#_Toc478372815)

[Figure 5 : Tableau des requêtes enregistrées classées par type de données 11](#_Toc478372816)

[Figure 6 : Fenêtre de gestion des groupes 12](#_Toc478372817)

[Figure 7 : Formulaire de création d'un groupe 13](#_Toc478372818)

[Figure 8 : Formulaire de mise à jour d'un groupe 13](#_Toc478372819)

[Figure 9 : Affichage d'une requête enregistrée 14](#_Toc478372820)

[Figure 10 : Diagramme de classes des requêtes prédéfinies amendé 17](file:///C:\Users\sgalopin\Desktop\Etude%20Technique%20-%20Enregistrement%20des%20requêtes%20OGAM_AM_SG_SC_PDU.docx#_Toc478372821)

**Terminologie**

**Glossaire**

|  |  |
| --- | --- |
| **Configurateur** | Progiciel IGN pour la configuration des métadonnées au sens OGAM |
| **Framework** | Ensemble cohérent de composants logiciels structurels qui sert à créer les fondations ainsi que les grandes lignes de tout ou partie d'un logiciel |
| **Javascript** | Langage de programmation de scripts principalement employé dans les pages web |
| **Métadonnées** | Ensemble de données permettant de décrire les données métier ainsi qu'une partie du système d'information (distinctes des métadonnées de type INSPIRE) |
| **Dataset** | Groupe de données ou processus de saisie |

**Acronymes**

|  |  |
| --- | --- |
| **DAO** | Objet d'accès aux données (Data Access Object) |
|  |  |
|  |  |
| **HTML5** | Format de données conçu pour représenter les pages web |
| **IGN** | Institut national de l'information géographique et forestière |
|  |  |
| **METAOGAM** | Métamodèle du produit OGAM |
| **OGAM** | Open-source Generic Application based on Metadata |
| **OGAM Bureau** | Application de saisie et de consultation bureautique générique basée sur un métamodèle |
| **PHP** | Langage de programmation libre principalement utilisé pour produire des pages Web dynamiques via un serveur |
| **REST** | Style d'architecture adaptée au web (REpresentational State Transfer) |
| **SGBD** | Système de gestion de base de données |
| **SINP** | Système d'information nature et paysage |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# Introduction

## Cadre de l'étude

Ce document s'inscrit dans le cadre de la migration du framework PHP du produit OGAM vers Symfony2. En effet, l’équipe Ginco a implémenté les fonctionnalités de sauvegarde, d’édition et de suppression d’une requête sous le Framework Zend, avec comme objectif que ces développements soient reversés dans le produit OGAM. Maintenant qu’OGAM utilise le framework Symfony2, il s’agit de migrer ces fonctionnalités de gestion des requêtes et d’y ajouter la gestion des groupes de requêtes.

## Contexte

Le produit OGAM (Open-source Generic Application based on Metadata) est issu du projet E-Forest mené par l’ex-IFN. Il est composé d'une interface de visualisation et de saisie de données naturalistes s’appuyant fortement sur le framework javascript ExtJS, qui vient d'être mis à jour. Il est générique car il est configuré par un ensemble de métadonnées définissant une partie de son architecture et les données métiers à manipuler.

Ce produit est utilisé à la fois pour des projets externes, comme récemment dans le cadre du SINP (Système d'Information sur la Nature et les Paysages) au sein du projet GINCO, et des projets IGN (refonte de la chaîne de traitement notamment). Il commence à être connu et utilisé au travers de nombreux établissements ou institutions comme l'ONF, l'IGN, l'ONEMA, l'INRAP, l'INRA, le MEDDE. Plusieurs applications s'appuient sur son métamodèle comme par exemple les applications nomades SONGE (Solution pour un Outil Nomade de Gestion de l'Eau) et RMQS2 (Réseau de mesure de la qualité des sols).

L'utilisation du métamodèle d’OGAM dans le projet SONGE pour l’ONEMA (Office National de l'Eau et des Milieux Aquatiques) a exigé son évolution pour intégrer de nouvelles fonctionnalités et augmenter sa généricité mais ces modifications n'ont pas encore été reversées. Son utilisation dans les projets SINP, RMQS2, SOIF (Saisie des observations de l'inventaire forestier) nécessite une nouvelle évolution qui devra être prise en compte par OGAM.

## Objectif du document

Ce document fait suite à un atelier avec le développeur OGAM chargé de la migration des requêtes enregistrées, une partie des développeurs et la coordinatrice du projet Ginco. Il a pour objectif de présenter les décisions prises ensemble quant aux fonctionnalités de gestion des requêtes sauvegardées, et de gestion des groupes de requêtes sauvegardées. De plus ce document présentera les évolutions du métamodèle nécessaires pour répondre aux nouveaux besoins.

# Analyse de l'existant

## Côté client et serveur

L’équipe Ginco a développé, pour les besoins de son projet, les fonctionnalités d’enregistrement, d’édition et de suppression d’une requête avancée. Ce travail a été réalisé sous le framework Zend, et ajouté au dépôt de code OGAM. Les fonctionnalités implémentées sont décrites dans la documentation utilisateur Ginco (cf <https://ginco.ign.fr/doc/recherche-visu/page-recherche-enregistrees.html>).

## SGBD et architecture des données

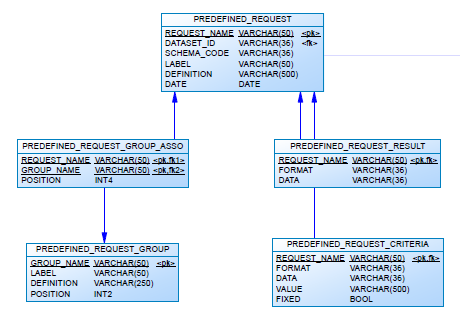


Figure 1 : Modèle de données actuel des requêtes prédéfinies

La base de données d’Ogam contient des tables permettant de stocker des requêtes. Elles ont notamment été utilisées pour créer des requêtes prédéfinies, disponibles sur l’IHM. Les requêtes contiennent des critères et des colonnes et peuvent être classées via des groupes.

# Analyse technique de la fonction d'enregistrement des requêtes

## Interface homme-machine

### Enregistrement d'une requête

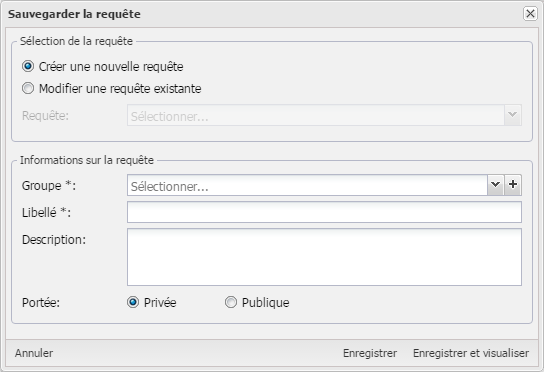


Figure 2 : Formulaire de création d'une requête enregistrée

Afin d’enregistrer une requête, un bouton « Enregistrer la requête » va être ajouté en bas du requêteur avancé. Il n’est visible que par les utilisateurs ayant le droit d’enregistrer une requête. Lors de la demande de sauvegarde (En cliquant sur le bouton "Enregistrer la requête") l'application récupérera toutes les colonnes et les critères avec leurs valeurs et ouvrira une fenêtre flottante contenant le formulaire illustré ci-dessus. La fenêtre flottante permet de ne pas surcharger l'écran de consultation (déjà bien chargé) par l'ajout d'un autre onglet. De plus, cela permet d'avoir une bonne longueur de champ pour tous les champs, ce qui améliore la lisibilité.

Un premier groupe de champs nommé "Sélection de la requête" contient deux champs de type radio permettant de choisir de créer une nouvelle requête (par défaut) ou de sélectionner une requête déjà existante afin de la modifier.

Un deuxième groupe de champs nommé "Informations sur la requête" contient de trois à quatre champs décrivant la requête à enregistrer. Un champ "Groupe" permet de préciser le groupe dans lequel la requête doit être placée, avec la possibilité de créer un nouveau groupe si besoin pour les utilisateurs ayant le droit d'administrer les groupes. Un champ "Libellé" permet de renseigner le nom de la requête et un champ "Description" permet de décrire la requête. Les utilisateurs ayant le droit d'administrer les requêtes publiques pourront rendre leur requête publique via le champ "Portée". Par défaut les requêtes créées via l'interface seront toutes privées. Dans ce cas, elles ne seront visibles que par l’utilisateur les ayant créées. Les champs Groupe, "Libellé" et "Portée" seront obligatoires (une étoile à côté du libellé du champ indiquera le caractère obligatoire du champ). "Description" sera facultative.



Figure 3 : Formulaire de modification d'une requête enregistrée

Le formulaire ci-dessus illustre le cas d'une modification de requête. L'utilisateur peut choisir de modifier une requête déjà existante au lieu d'en créer une nouvelle. Une fois la sélection du bouton radio "Modifier une requête existante", le champ "Requête" se dégrise pour permettre la sélection d'une requête via une liste déroulante. Un utilisateur n’ayant pas les droits de gestion des requêtes publiques ne verra que ses propres requêtes privées dans la liste déroulante. Un utilisateur ayant les droits de gestion des requêtes publiques verra ses propres requêtes privées ainsi que les requêtes publiques. Une fois une requête sélectionnée, les champs du groupe de champs "Informations sur la requête" sont automatiquement remplis avec les valeurs enregistrées de la requête. Une demande de modification d’une requête enregistrée via la page des requêtes enregistrées présélectionnera une requête. Le formulaire ne doit pas permettre l’enregistrement d’une requête si le triplet de champs (libellé, utilisateur, type de données, portée) existe déjà, et le signaler à l’utilisateur via un message « Le nom de la recherche existe déjà. Veuillez en indiquer un autre ».

Il est primordial pour Ginco que des requêtes de même nom puissent exister s’il s’agit de deux utilisateurs différents. Il est également primordial que deux requêtes de même nom puissent exister avec le même utilisateur s’il s’agit de types de données différents ou de portées différentes

### Gestion des requêtes enregistrées

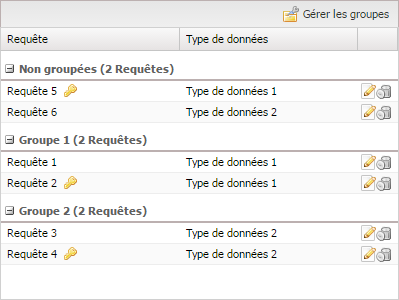


Figure 4 : Tableau des requêtes enregistrées classées par groupes

Le tableau des requêtes enregistrées existe déjà actuellement (sous le nom de requêtes prédéfinies). Plusieurs éléments vont y être rajoutés:

- Une icône à côté du libellé des requêtes privées permettra de rapidement identifier les requêtes de ce type,

- Une colonne "Type de données" afin de permettre de classer les requêtes par type de données,

- Une colonne "Groupe" afin de permettre de (re)classer les requêtes par groupe,

- Un bouton "Gérer les groupes" afin de pouvoir assurer la gestion des groupes (uniquement disponible pour les utilisateurs ayant les droits d'administration sur les groupes),

- Des boutons "Editer" au niveau de chaque requête afin de permettre de recharger les requêtes dans le requêteur avancé et de les modifier (dans ce cas le formulaire d'enregistrement de la requête sera en mode modification avec la requête sélectionnée),

- Des boutons "Supprimer" au niveau de chaque requête afin de permettre la suppression des requêtes. Lorsque l’utilisateur cliquera sur le bouton de suppression d’une requête, une popup s’affichera avec le message « Voulez-vous vraiment supprimer la requête ? ». Les choix proposés seront « Oui » et « Non ».

Les boutons "Editer" et "Supprimer" ne seront disponibles que si:

- L'utilisateur a les droits d'administration pour ses requêtes privées dans le cas des requêtes privées,

- L'utilisateur a les droits d'administrations pour ses requêtes publiques dans le cas des requêtes publiques,

- L’utilisateur est propriétaire de la requête.

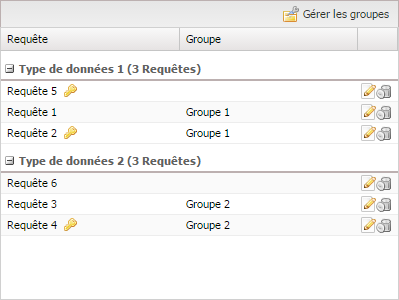


Figure 5 : Tableau des requêtes enregistrées classées par type de données

**Notes :**

Il ne sera plus obligatoire pour les requêtes d'être classées dans un groupe pour être visibles. Ainsi un groupe "Non groupée(s)" (créé que pour l’affichage) regroupant toutes les requêtes non groupées permettra de les visualiser.

En particulier, la suppression d'un groupe n’entraînera pas la suppression des requêtes y étant attachées. L’utilisateur pourra attacher à nouveau ces requêtes à un autre groupe en les éditant et en leur attribuant un nouveau groupe.

La gestion de l'ordre d'affichage des requêtes via l'interface est volontairement laissée de côté pour le moment (pas de besoin réel), si ce n’est un classement des groupes ou des types de données dans l’ordre alphabétique selon l’organisation appliquée, et un classement des requêtes par ordre alphabétique au sein de l’organisation précédente. En fonction des possibilités du framework ExtJS, pour les groupes, un classement pourra être fait d’abord sur le champ position, puis par ordre alphabétique. Sachant que l’équipe Ginco mettra les requêtes créées par le configurateur en position « 1 », et celles créées dans l’interface en position « 2 ». Idéalement les requêtes sans-groupe seront positionnées en « 3 ».

### Gestion des groupes

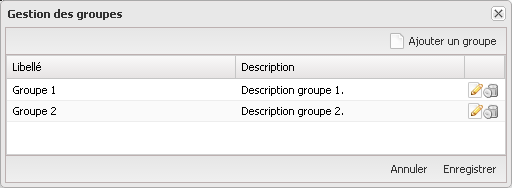


Figure 6 : Fenêtre de gestion des groupes

Pour les utilisateurs ayant les droits d'administration des groupes, il sera possible de créer, d'éditer et de supprimer des groupes. La fenêtre de gestion des groupes ci-dessus pourra être ouverte via le bouton "Gérer les groupes" présent dans la barre d'outils supérieure du tableau des requêtes enregistrées. Les groupes sont toujours publics, il n'est donc pas possible d'enregistrer des groupes privés. L'utilisateur pourra créer un nouveau groupe via le bouton "+" positionné à côté du champ "Groupe" du formulaire de sauvegarde des requêtes ou via le bouton "Ajouter un groupe" de la barre d'outils de la fenêtre de gestion des groupes. Il pourra également modifier ou supprimer les groupes via les boutons disponibles pour chaque groupe.

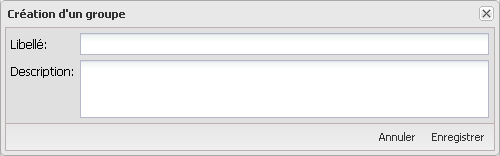


Figure 7 : Formulaire de création d'un groupe

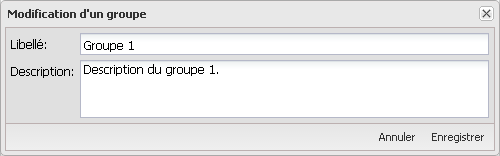


Figure 8 : Formulaire de mise à jour d'un groupe

**Notes :**

La gestion de l'ordre d'affichage des groupes via l'interface est volontairement laissée de côté pour le moment (pas de besoin réel). Par défaut, les groupes s’afficheront dans l’ordre alphabétique. En fonction des possibilités du framework ExtJS, pour les groupes, un classement pourra être fait d’abord sur le champ position, puis par ordre alphabétique. Sachant que l’équipe Ginco mettra les requêtes créées par le configurateur en position « 1 », et celles créées dans l’interface en position « 2 ». Idéalement les requêtes sans-groupe seront positionnées en « 3 ».

L’enregistrement d’un groupe dont le libellé est déjà utilisé par un autre groupe ne doit pas être permis.

### Affichage des requêtes enregistrées

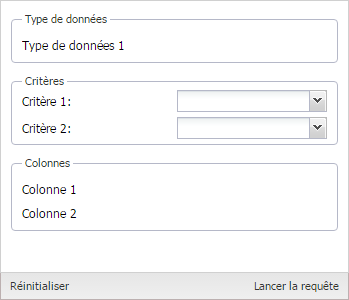


Figure 9 : Affichage d'une requête enregistrée

L'affichage des requêtes prédéfinies existe déjà actuellement. Plusieurs éléments vont y être ajoutés:

- Un groupe de champs "Type de données" contenant le libellé du type de données de la requête sélectionnée,

- Un groupe de champs "Colonnes" contenant les colonnes de la requête sélectionnée,

- Un groupe de champs "Critères" contenant tous les critères de la requête sélectionnée (la notion de critère fixé disparaît).

- Le libellé du bouton "Annuler" est remplacé par le libellé "Réinitialiser" pour homogénéiser avec le même bouton disponible pour les requêtes avancées.

## Interface client/serveur

L'utilisation de l’architecture REST (REpresentational State Transfer), qui repose sur le protocole http, est préconisée pour des transactions typiques comme dans notre cas.

### Les nouvelles requêtes

#### Requête de sauvegarde d'une requête (POST)

Lors de la demande de sauvegarde d'une requête, une requête de demande d'écriture des données sera envoyée au serveur. Cette dernière devra comprendre les données suivantes:

* L'identifiant du groupe,
* Le libellé de la requête prédéfinie,
* La description de la requête prédéfinie,
* La portée de la requête prédéfinie,
* Le type de données auquel appartient la requête
* Une liste des colonnes retenues,

#### Une liste des critères retenus avec leurs valeurs. Requête de modification d'une requête enregistrée (PUT)

Lors de la demande de modification d'une requête enregistrée, une requête de demande de modification des données sera faite vers le serveur. Cette requête devra comprendre les données suivantes:

* L'identifiant de la requête,
* L'identifiant du groupe,
* Le libellé de la requête,
* La description de la requête,
* La portée de la requête,
* Le type de données auquel appartient la requête
* Une liste des colonnes retenues,
* Une liste des critères retenus avec leurs valeurs.

#### Récupération de la liste des groupes (GET)

Lors de la demande d'affichage de la liste des groupes, une requête de demande des données sera faite vers le serveur. Cette requête devra comprendre les données suivantes en retour pour chaque groupe :

* L'identifiant du groupe,
* Le libellé du groupe,
* La description du groupe.

#### Requête de création d'un groupe (POST)

Lors de la demande de création d'un groupe, une requête de demande d'écriture des données sera envoyée au serveur. Cette requête devra comprendre les données suivantes:

* Le libellé du groupe.
* La description du groupe.

#### Requête de modification d'un groupe (PUT)

Lors de la demande de modification d'un groupe uniquement, une requête de demande de modification des données sera envoyée au serveur. Cette requête devra comprendre les données suivantes:

* L'identifiant du groupe,
* Le nouveau libellé du groupe.
* La nouvelle description du groupe.

#### Requête de suppression d'une requête ou d'un groupe (DELETE)

Lors de la demande de suppression d'un groupe ou d'une requête enregistrée, une requête de demande de suppression des données sera envoyée au serveur. Cette requête devra comprendre en paramètre l'identifiant du groupe ou l'identifiant de la requête.

### Mise à jour des requêtes existantes

#### Récupération de la liste des requêtes

Il conviendra de mettre à jour la requête "query/ajaxgetpredefinedrequestlist" en ajoutant les données suivantes au JSON retourné :

* La portée de la requête (privée ou publique),
* Le libellé du "Type de donnée".

#### Récupération des données sur la requête

Il conviendra de renommer la requête "query/ajaxgetpredefinedrequestcriteria" en "query/ajaxgetpredefinedrequest " et d’ajouter les données suivantes au JSON retourné :

* Le libellé du type de donnée,
* Les libellés des colonnes,
* Les libellés et les valeurs de tous les critères.

## Côté Serveur

Six nouvelles actions devront être ajoutées dans le contrôleur de requêtes ("QueryController") correspondant chacune à l'une des requêtes REST décrites précédemment. Il sera également nécessaire de revoir deux fonctions. Il faudra également modifier le modèle de données des requêtes enregistrées pour répondre aux nouveaux besoins.

## Interface serveur/SGBD

Le framework PHP "Symfony" utilisé par l'application OGAM utilise la librairie "Doctrine" pour assurer l'interface avec la base de données.

## Données

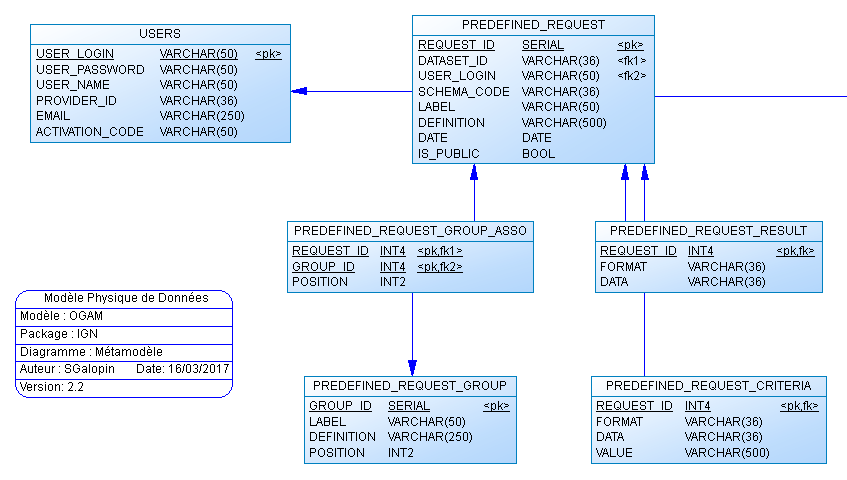


Figure 10 : Diagramme de classes des requêtes prédéfinies amendé

Le modèle de données doit être amendé avec des attributs de propriété et de visibilité afin de répondre aux besoins. On retrouve ainsi une colonne "IS\_PUBLIC" dans la table "PREDEFINED\_REQUEST" et un lien définissant le rattachement des requêtes à leurs propriétaires.

Les colonnes "REQUEST\_NAME" et "GROUP\_NAME" ont été renommées en "REQUEST\_ID" et "GROUP\_ID" et sont devenues de type "SERIAL" afin de permettre la création de groupes et de requêtes via l'interface de l'application.

La colonne "FIXED" a été supprimée de la table "PREDEFINED\_REQUEST\_CRITERIA" car elle n'est plus utilisée.

## Droits

L'application va se doter de trois nouveaux types de droit :

* Un droit d'administration (Création, Modification, Suppression) de ses requêtes privées, nommé « Gérer ses requêtes privées » ;
* Un droit d'administration (Création, Modification, Suppression) de ses requêtes publiques, nommé « Gérer ses requêtes publiques » ;
* Un droit d'administration (Création, Modification, Suppression) des groupes (qui sont toujours publics), nommé « Gérer les groupes de requêtes ».

Note :

Dans tous les cas il n’est possible d’administrer que ses propres requêtes qu’elles soient publiques ou privées.

# Mise en œuvre des développements

Afin de limiter les impacts et de maîtriser le planning, un découpage en trois phases est recommandé.

## Phase 1 – Enregistrement des requêtes

Cette phase consiste à reprendre ce qui a été fait par l’équipe Ginco, en le migrant en symfony2. Il s’agit de :

* La mise en place du formulaire de sauvegarde des requêtes,
* La mise à jour du tableau des requêtes enregistrées,
* La mise à jour du modèle,
* L'ajout des nouveaux droits.

## Phase 2 – Modification de l'affichage des requêtes enregistrées

Cette phase comprend :

* La modification de l'affichage des requêtes.

## Phase 3 – Gestion des groupes

Cette phase comprend :

* La fenêtre de gestion des groupes,
* Le formulaire de création et de modification des groupes.

## Recettes des développements

### Jeux de données de tests

Obtenir un jeu de données métier permettant de tester l'intégralité des fonctionnalités d'OGAM n'est pas chose facile et pose des problèmes de droits d'utilisation des données.

Afin de résoudre le problème, le produit OGAM dispose de son propre jeu de données fictives permettant de tester l'intégralité des fonctionnalités du produit. Celles-ci allant fortement s'étoffer, il sera nécessaire d'enrichir le jeu de données.

Cependant, concernant l’enregistrement de requêtes, il n’est pas nécessaire que l’application contienne des données pour pouvoir réaliser la majeure partie des tests.

### Recettes des phases par les développeurs

Afin d'assurer la recette technique du code, les développeurs devront présenter à fin de chaque nouvelle fonctionnalité majeure, leur code aux autres développeurs de l'action de développement, afin d'assurer la cohésion du code. Ils devront également présenter les tests unitaires mis en place pour tester leur code. La validation de l'ensemble des tests unitaires fait partie de la recette technique.

Afin d'assurer la recette fonctionnelle du code, un cahier de recette sera rédigé par le chef de produit. Les différents tests seront passés par les développeurs en fonction de l'attribution des développements.

### Recettes des phases par le chef de produit

#### Vérification d'aptitude (VA)

Le chef de produit devra à son tour passer les cahiers de tests de chaque livraison pour les valider.

#### Vérification en service régulier (VSR)

Les tests de montée en charge pourront se faire avec de gros volumes de données. Le projet GINCO, de par sa dimension nationale pourra réaliser les tests avec ses jeux de données. Ils seront au format du standard SINP.