



**MAKINA
CORPUS**

CR Aquitaine
Manuel d'exploitation

Projet Gr@ce
23 mai 2012

Réf. MEX/GRACE
Eric BREHAULT
36 rue Babinet
31100 TOULOUSE
05 34 56 69 58

<http://www.makina-corpus.com>

Table des Matières

1 Introduction	3
2 Architecture	3
[2.1] Architecture physique	3
[2.2] Architecture logicielle	3
3 Lancement / arrêt des services	4
4 Mode debug	5
5 Paramétrage	5
[5.1] Paramètres généraux	5
[5.1.1] Mots de passe	5
[5.1.2] Email de contact, URL du site, et autres	6
[5.2] Analyses scalaires	6
[5.3] Page d'accueil et pages de contenus	7
[5.4] Cartes	7
[5.4.1] Création de couches	7
[5.4.2] Styles	8
[5.4.3] Gestion des couches d'une carte	8
6 Évolutions	9
[6.1] Modification du code métier	9
[6.2] Déploiement de composants tiers	9
[6.3] Ajout/suppression de modules	10
[6.4] Mise à jour des jobs Talend	10

1 Introduction

La plate-forme [Gr@ce](#) est un outil web dont l'objectif est de mettre à jour le référentiel réseau de la région Aquitaine.

[Gr@ce](#) contient une base de données contenant ce référentiel, et cette base de données peut être mise à jour par les partenaires concernés de plusieurs façons :

- chargement d'une archive .zip contenant des fichiers GEO,
- chargement d'un dump SQL,
- synchronisation depuis un terminal mobile suite à des saisies offline sur le terrain,
- retouches manuelles depuis l'interface cartographique.

2 Architecture

[2.1] Architecture physique

L'application fonctionne sur deux machines virtuelles, CRA26 et CRA27. CRA26 héberge le frontal web et la base de données, et répond pour l'URL <http://grace.aquitaine.fr/>

CRA27 héberge le serveur cartographique Geoserver. Il répond pour l'URL <http://preprod2-grace.aquitaine.fr:8080/> mais cette adresse est réservée aux gestionnaires de la plate-forme.

[2.2] Architecture logicielle

Django gère toutes les requêtes entrantes (requêtes vers les pages du frontal web, mais également requêtes vers les services cartographiques). Il y a 2 bases de données qui sont actuellement sur la même instance PostgreSQL :

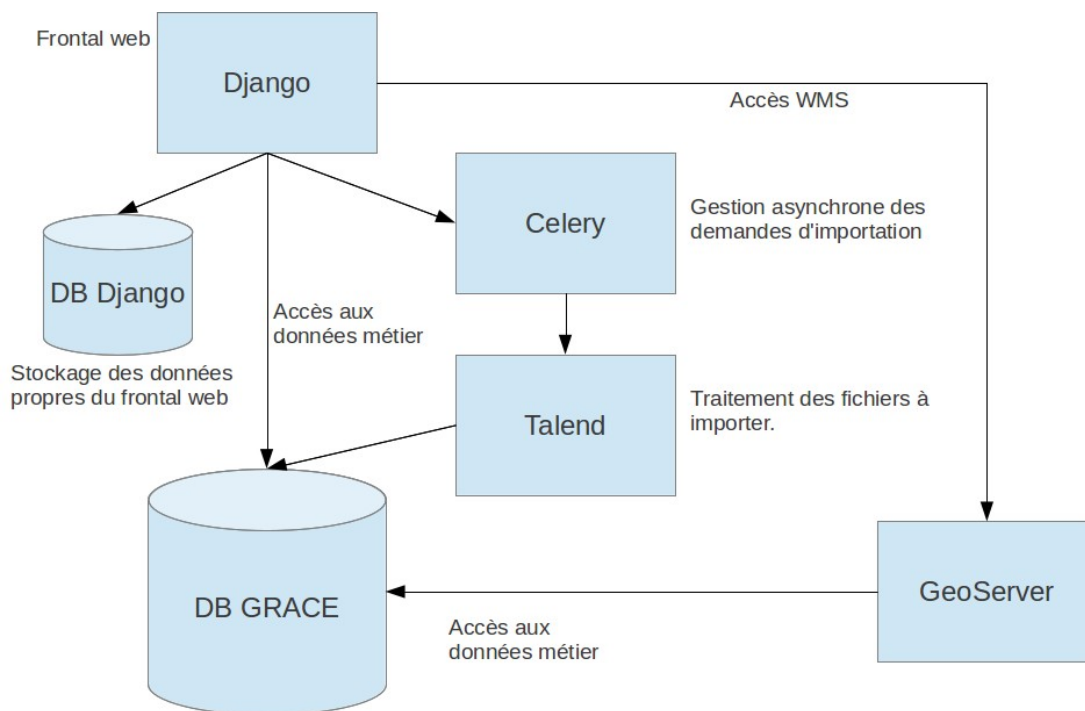
- la base de données propres à Django (comptes utilisateurs, paramètres, etc.)
- la base de données métier [Gr@ce](#).

Le traitement des fichiers à importer est assuré par des jobs Talend que Django lance de manière asynchrone via Celery, qui est un gestionnaire de tâches fonctionnant sur RabbitMQ.

Les jobs Talend écrivent directement dans la base [Gr@ce](#).

Django écrit également directement dans la base [Gr@ce](#) (dans le cadre des modules Mise à jour et Saisie terrain).

Les couches cartographiques sont produites par Geoserver, mais ce n'est pas Geoserver qui sert aux clients web, les appels étant proxifiés par Django afin d'assurer le filtrage spatial dynamique (assurant qu'un utilisateur ne voit pas les données extérieures à sa zone géographique).



3 Lancement / arrêt des services

Se connecter en ssh sur cra26 avec le compte root.

Pour les différents services les commandes de lancement / arrêt / redémarrage sont :

Apache :

```
/etc/init.d/apache2 start
/etc/init.d/apache2 stop
/etc/init.d/apache2 restart
```

RabbitMQ:

```
/etc/init.d/rabbitmq-server start
/etc/init.d/rabbitmq-server stop
/etc/init.d/rabbitmq-server restart
```

PostgreSQL :

```
/etc/init.d/postgresql start  
/etc/init.d/postgresql stop  
/etc/init.d/postgresql restart
```

Geoserver :

Se connecter en ssh sur cra27 avec le compte root, puis :

```
/etc/init.d/tomcat6 start  
/etc/init.d/tomcat6 stop  
/etc/init.d/tomcat6 restart
```

4 Mode debug

Pour mettre Django en mode debug, se connecter en ssh sur cra26, et avec le compte 'grace', ajouter :

```
DEBUG = True
```

en deuxième ligne du fichier settings.py :

```
cd /data/grace/venv  
vim parts/django/django_extrasettings/settings.py  
# faire la modification  
touch bin/django.wsgi
```

Attention : le mode debug affiche des informations sensibles sur l'interface web.

5 Paramétrage

[5.1] Paramètres généraux

[5.1.1] Mots de passe

Tous les mots de passe utilisés par Django pour se connecter aux autres composants (PostgreSQL, Geoserver, etc.) se trouvent dans le fichier :
/data/grace/venv/etc/sys/prod-settings.cfg

Toute modification ne sera prise en compte qu'en lançant le buildout :

```
cd /data/grace/venv  
su grace  
source bin/activate
```

```
bin/buildout -N buildout-prod.cfg  
touch bin/django.wsgi
```

[5.1.2] Email de contact, URL du site, et autres

L'email de contact et l'url du site se trouvent dans le fichier suivant :
/data/grace/venv/buildout-prod.cfg

Ce fichier comporte également d'autres paramètres, comme le chemin vers les jobs Talend ou l'URL du Geoserver, qu'il n'est pas nécessaire de modifier à priori.

Toute modification ne sera prise en compte qu'en lançant le buildout :

```
cd /data/grace/venv  
su grace  
source bin/activate  
bin/buildout -N buildout-prod.cfg  
touch bin/django.wsgi
```

[5.2] Analyses scalaires

Les analyses proposées par le module Analyse sont paramétrable via l'interface d'administration de Django :

- se connecter avec un compte administrateur sur le site,
- cliquer sur Administration (en haut à droite),
- cliquer sur Analyses.

Depuis cette page on peut créer de nouvelles analyses, ou modifier/supprimer les analyses existantes.

Le principe consiste à enregistrer une requête SQL qui retournera les résultats attendus pour l'analyse en question.

Comme les résultats sont (a priori) dépendant de la zone géographique de l'utilisateur, il existe deux marqueurs : ##CODES_INSEE## et ##GEOM##, qui sont automatiquement substitués par, respectivement, la liste des codes INSEE ou la géométrie de la zone de l'utilisateur.

Exemple :

```
select occp_ep as "Occupation", count(*) as Nombre from  
element_branchement_passif where id_com_insee_ep in  
(##CODES_INSEE##) group by occp_ep;
```

[5.3] Page d'accueil et pages de contenus

La pages d'accueil et les pages de contenus sont gérables depuis l'interface d'aministration de Django :

- se connecter avec un compte administrateur sur le site,
- cliquer sur Administration (en haut à droite),
- cliquer sur Pages.

Ce module propose un éditeur HTML classique (TinyMCE) qui permet de saisir un texte et de le mettre en forme.

Note : la page d'accueil utilise un modèle de page spécifique disposant de 3 zones de textes (pour les 3 actualités).

[5.4] Cartes

[5.4.1] Création de couches

On peut créer de nouvelles couches depuis l'interface d'administration de Geoserver, à cette URL :

<http://preprod2-grace.aquitaine.fr:8080/geoserver/web/>

Aller dans Layers / Add new resources.

Choisir la source de la couche. Actuellement, le Geoserver Grace dispose de 4 sources :

- la base PostgreSQL/PostGIS Grace (qui contient tous les objets métiers),
- les shapefiles Opérateurs,
- les shapefiles Données Autres,
- le serveur WMS Pigma.

Si la couche a ajouté ne provient pas d'une de ces sources, il faut au préalable ajouter une nouvelle source (Stores / Add new store).

Choisir la couche, et cliquer sur Publish.

La documentation détaillée de Geoserver est disponible ici :

<http://docs.geoserver.org/stable/en/user/> (Anglais)

<http://www.georchestra.org/documentation/administrateur/geoserver/>
(Français)

L'ajout de couches à une carte se fait via Django (voir Gestion des couches d'une carte).

[5.4.2] Styles

La création de nouveau style se fait depuis Geoserver, à cette URL :

<http://preprod2-grace.aquitaine.fr:8080/geoserver/web/>

Aller dans Styles / Add a new style.

Les styles sont saisis au format standard SLD (voir documentation ici :

<http://docs.geoserver.org/stable/en/user/styling/>).

L'approche la plus simple consiste à copier un style existant et à l'adapter.

Il est possible de réaliser des analyses graphiques (avec des rendus différents suivant les valeurs de certains attributs). Plusieurs exemples sont disponibles :

- artere_occupation
- artere_nb_fourreaux
- artere_type_implementation
- artere_type_logique

Ils montrent comment créer des règles (rules) dynamiques en fonction des valeurs des attributs.

Une fois les styles créés, il faut les rendre disponibles pour les couches concernées :

- aller dans Layers, et cliquer sur la couche,
- aller dans l'onglet Publishing,
- sélectionner les styles désirés dans Available styles, et les faire passer dans Selected styles.

[5.4.3] Gestion des couches d'une carte

Pour gérer les couches d'une carte, aller avec un compte administrateur sur la carte elle-même, par exemple :

<http://grace.aquitaine.fr/carto/1>

Puis cliquer sur Modifier la carte.

Sur cette page on peut gérer les droits d'accès sur la carte.

En cliquant sur Afficher ou modifier cette carte, on accède à l'éditeur de carte :

Ajout :

Le volet de gauche permet de changer l'ordre des couches (par glisser-déposer).

Le bouton Plus (+, en vert), permet d'ajouter une nouvelle couche. La fenêtre qui s'affiche liste toutes les couches disponibles sur le Geoserver Grace.

Choisir la couche et cliquer sur Add layers.

Suppression :

Dans le volet de gauche, se positionner sur une couche, puis cliquer sur le bouton Moins (-, en rouge).

Styles :

Dans le volet de gauche, se positionner sur une couche, puis cliquer sur le bouton Palette.

Dans la fenêtre qui s'affiche on peut choisir le style voulu (cette liste dépend des styles sélectionnés pour cette couche dans Geoserver, voir ci-dessus).

6 Évolutions

[6.1] Modification du code métier

Grace s'appuie principalement sur l'outil Geonode qui est installé dans le dossier **/data/grace/venv/src/geonode**.

Tous les développements spécifiques Grace ont été implémentés dans le dossier **/data/grace/venv/grace**, qui dérive et surcharge les classes de Geonode.

Chaque fonctionnalité est gérée dans une application Django distincte (ainsi le fonctionnalité d'importation est dans l'application /processing_file, la fonctionnalité de saisie terrain est dans l'application /mobile, etc.).

L'habillage du site, quant à lui, est implémenté dans le dossier /media conformément aux bonnes pratiques Django.

[6.2] Déploiement de composants tiers

Si on souhaite ajouter une application Django externe supplémentaire, il faut d'une part déployer l'egg correspondant, pour cela il suffit de déclarer le nom de cet egg dans la section eggs du fichier buildout.cfg et de faire tourner le buildout :

```
cd /data/grace/venv
su grace
source bin/activate
```

```
bin/buildout -N buildout-prod.cfg
```

Il faut ensuite déclarer l'application à Django en l'ajoutant dans `INSTALLED_APPS` dans **grace/settings.py** (il est possible que l'application nécessite d'autres paramètres, pour cela, se reporter à sa documentation).

Si l'application a besoin de tables dans la base PostgreSQL, la mise à jour se fait soit avec la commande Django `syncdb` ou bien avec le commande `South migrate` (se reporter à la documentation) :

```
cd /data/grace/venv
su grace
source bin/activate
django syncdb
# OU
django migrate nom_application
```

Toute adaptation et/ou surcharge doit être faite dans **/data/grace/venv/grace**.

[6.3] Ajout/suppression de modules

L'ajout de module nécessite d'abord son développement (voir Modification du code métier), ou son déploiement s'il s'agit d'une application Django tierce (voir Déploiement de composants tiers).

Pour mettre à jour la liste des modules proposés à l'utilisateur, il faut modifier les fichiers :

/data/grace/venv/grace/templates/tool_list.html (pour la première barre d'outils)

et **/data/grace/venv/grace/templates/other_links_list.html** (pour la seconde barre d'outils)

On y contrôle l'ordre des modules et également leur affichage en grisé ou pas en fonction du profil de l'utilisateur.

[6.4] Mise à jour des jobs Talend

Les jobs Talend sont déployés dans le dossier **/data/grace/jobs**.

Pour mettre à jour les jobs `Main_check_prod` ou `Main_prod`, supprimer le dossier du job, exporter le job correspondant depuis Talend en utilisant le contexte « prod », et extraire le zip résultant dans le dossier `/jobs`.

Si les noms des jobs devaient changer (et donc les noms des répertoires), il faudrait reporter le changement dans le fichier `buildout-prod.cfg` (voir plus haut).