



# Documentation UpdatEngine

## Installation et migration version 2.1.1 stable

Version de la documentation : doc-2.1.1-stable-1

Auteur: Yves Guimard

Référence internet : <http://www.updatengine.com>

Licence Creative Commons (BY – NC – ND) : Le titulaire des droits autorise l'utilisation de l'œuvre originale à des fins non commerciales, mais n'autorise pas la création d'œuvres dérivés.



## Table des matières

1 -Introduction.....	3
2 -Installation du serveur (nouveau serveur).....	4
2.1 - Installation Debian / Ubuntu – apache2 / MySQL.....	4
2.1.1 -Installation de l’environnement et des dépendances nécessaires.....	4
2.1.2 -Installation des dépendances dans l’environnement virtuel UE-environment.....	4
2.1.3 -Initialisation de la base de données MySQL.....	5
2.1.4 -Patch de Django 1.6.2.....	7
2.1.5 -Fin de configuration.....	8
2.2 - Installation d’UpdatEngine Server (Debian / Ubuntu – apache2/SQLite).....	9
2.2.1 -Installation de l’environnement et des dépendances nécessaires.....	9
2.2.2 -Installation des dépendances dans l’environnement virtuel UE-environment.....	9
2.2.3 -Initialisation de la base de données SQLite.....	10
2.2.4 -Patch de Django 1.6.2.....	11
2.2.5 -Fin de configuration.....	12
3 -Migration d'un serveur existant vers la version 2.1.1 stable.....	13
3.1 - Sauvegarde de l'application.....	13
3.2 - Procédure de mise à jour.....	13
3.2.1 -Mise à jour et bascule sur la branche stable (nommée master) :.....	13
3.2.2 -Mise à jour des modules:.....	13
Si vous utilisez le moteur de base de données MySQL :.....	13
Si vous utilisez le moteur de base de données SQLite :.....	14
3.2.3 -Mise à jour de la base.....	14
3.2.4 -Adaptation du fichier settings.py.....	14
3.2.5 -Chargement des groupes par défaut.....	15
3.2.6 -Patch de Django 1.6.2.....	15
3.2.7 -Redémarrage d'apache2.....	15

## **1 - Introduction**

Cette documentation a pour but de présenter comment installer un nouveau serveur UpdatEngine ou comment migrer un serveur existant dans la dernière version du logiciel.

## 2 - Installation du serveur (nouveau serveur)

Dans ce chapitre, vous verrez comment installer un nouveau serveur UpdatEngine. Deux types d'installations sont pour l'instant détaillées :

- Serveur Debian ou ubuntu utilisant apache2 comme serveur web et MySQL comme moteur de base de données
- Serveur Debian ou ubuntu utilisant apache2 comme serveur web et SQLite comme moteur de base de données

### 2.1 - Installation Debian / Ubuntu – apache2 / MySQL

#### 2.1.1 - Installation de l'environnement et des dépendances nécessaires

Installez tout d'abord les différentes dépendances nécessaires:

```
$ sudo apt-get install apache2 libapache2-mod-wsgi python-virtualenv python-pip libxml2-dev libxslt-dev python-dev libmysqlclient-dev git-core mysql-server
```

Créez maintenant l'environnement virtuel qui hébergera le serveur UpdatEngine. Nous le créerons directement sous `/var/www` dans un répertoire nommé UE-environment (veillez à respecter le nom de ce répertoire pour pouvoir profiter du fichier de configuration apache mis à disposition dans cet article).

```
$ sudo virtualenv --distribute --no-site-packages /var/www/UE-environment
```

```
$ cd /var/www/UE-environment/
```

```
$ sudo git clone https://github.com/updatengine/updatengine-server.git
```

L'ensemble des fichiers du serveur UpdatEngine sont alors disponibles sous `/var/www/UE-environment/updatengine-server`

Installons maintenant les dépendances nécessaires à UpdatEngine dans votre environnement virtuel:

#### 2.1.2 - Installation des dépendances dans l'environnement virtuel UE-environment

Les différentes dépendances sont détaillées dans le fichier `pip-packages.txt`.

Installez les avec l'utilitaire `/var/www/UE-environment/bin/pip install` ce qui permettra de n'installer les dépendances que dans votre environnement virtuel et dans les versions spécifiées.

```
$ sudo /var/www/UE-environment/bin/pip install --upgrade distribute
$ sudo /var/www/UE-environment/bin/pip install --upgrade setuptools
```

```
$ sudo /var/www/UE-environment/bin/pip install -r /var/www/UE-environment/updatengine-server/requirements/pip-packages.txt
```

À l'issue de l'exécution de la commande, vous disposez de tout le nécessaire pour faire fonctionner l'application.

**Remarque:** si vous êtes positionné derrière un proxy, pensez à utiliser l'option `--proxy=http://votre_proxy:port_proxy`

Par exemple :

```
$ sudo /var/www/UE-environment/bin/pip install --proxy=http://mon\_proxy:8080
--upgrade distribute
```

### 2.1.3 - Initialisation de la base de données MySQL

Créez tout d'abord une base de données dédiée à l'application (utilisez par exemple `updatengine` comme nom):

```
$ mysqladmin -u root -p create updatengine
```

puis créez un utilisateur dédié à cette base de données (`updatengineuser` dans cet exemple avec comme mot de passe `unmotdepasse`):

```
$ mysql -u root -p -e "GRANT ALL PRIVILEGES ON updatengine.* TO
'updatengineuser'@'localhost' IDENTIFIED by 'unmotdepasse' WITH GRANT OPTION;"
```

Enfin, chargez la table des fuseaux horaires :

```
$ mysql_tzinfo_to_sql /usr/share/zoneinfo | mysql -u root -p mysql
```

```
$ sudo service mysql restart
```

Vous pouvez maintenant modifier le fichier de configuration `/var/www/UE-environment/updatengine-server/updatengine/settings.py` pour y indiquer l'adresse de votre serveur et les paramètres de connexion à la base de données.

Copiez tout d'abord le fichier `settings.py.model_mysql` vers `settings.py`:

```
$ sudo cp /var/www/UE-environment/updatengine-server/updatengine/settings.py.model_mysql /var/www/UE-environment/updatengine-server/updatengine/settings.py
```

Ouvrez le fichier `/var/www/UE-environment/updatengine-server/updatengine/settings.py` et renseignez la variable **PROJECT\_URL** avec l'adresse et le port de votre serveur.

Par exemple, si l'adresse ip de votre serveur est **192.168.1.45**, vous devez renseigner la ligne de cette façon:

```
PROJECT_URL = 'http://192.168.1.45:1979'
```

**Remarque:** le port par défaut 1979 est défini dans le fichier de configuration `apache-updatengine`.

Renseignez maintenant les paramètres de connexion à la base de données dans le paragraphe suivant:

```
DATABASES = {
```

```
'default': {
  'ENGINE': 'django.db.backends.mysql', # Add 'postgresql_psycopg2',
'mysql', 'sqlite3' or 'oracle'.
  'NAME' : 'updatengine',
  'USER': 'updatengineuser',           # Not used with sqlite3.
  'PASSWORD': 'unmotdepasse',         # Not used with sqlite3.
  'HOST': '',                          # Set to empty string for localhost.
Not used with sqlite3.
  'PORT': '',                          # Set to empty string for default. Not
used with sqlite3.
}
```

Après avoir fait ces modifications, sauvegardez et fermez ce fichier.

**Remarque:** ces informations correspondent à l'exemple pris ci-dessus lors de la création de la base de données et de son utilisateur. Il faudra donc l'adapter à votre contexte.

Avant de redémarrer le service apache2, copiez le fichier de configuration standard puis activez ce nouveau site:

```
$ sudo cp /var/www/UE-environment/updatengine-server/requirements/apache-
updatengine.conf /etc/apache2/sites-available/apache-updatengine.conf
```

```
$ sudo a2ensite apache-updatengine
```

Activez également le mode wsgi si ce n'est pas déjà fait:

```
$ sudo a2enmod wsgi
```

Redémarrez maintenant le service apache2 pour prendre en compte la nouvelle configuration:

```
$ sudo service apache2 restart
```

Puis initialisez la base de données:

```
$ sudo /var/www/UE-environment/bin/python /var/www/UE-environment/updatengine-
server/manage.py syncdb
```

L'installateur vous demande alors si vous souhaitez créer un nouvel utilisateur pour l'application: Répondez **yes** à la question posée pour en créer un nouveau:

*You just installed Django's auth system, which means you don't have any superusers defined. Would you like to create one now? (yes/no): yes*

Saisissez ensuite le nom de l'utilisateur et entrez son adresse email:

*Username (leave blank to use 'root'): admin*

*E-mail address: [admin@local.domain](mailto:admin@local.domain)*

Saisissez ensuite un mot de passe pour cet utilisateur, confirmez le et validez par entrée pour terminer l'initialisation de la base de données:

*Password:*

*Password (again):*

La base de données est maintenant initialisée.

Pour terminer sa configuration, nous allons maintenant passer toutes les mises à jour effectuées depuis le début de son développement avec la commande migrate:

```
$ sudo /var/www/UE-environment/bin/python /var/www/UE-environment/updatengine-server/manage.py migrate
```

**Remarque:** le schéma de base de données est géré avec le module python South pour faciliter les évolutions et les migrations des bases de données entre versions.

Enfin chargez les données initiales de l'application (configuration par défaut):

```
$ sudo /var/www/UE-environment/bin/python /var/www/UE-environment/updatengine-server/manage.py loaddata /var/www/UE-environment/updatengine-server/initial_data/configuration_initial_data.yaml
```

```
$ sudo /var/www/UE-environment/bin/python /var/www/UE-environment/updatengine-server/manage.py loaddata /var/www/UE-environment/updatengine-server/initial_data/groups_initial_data.yaml
```

L'installation de l'application et l'initialisation de la base de données est maintenant terminée. Il ne reste plus qu'à rendre l'utilisateur www-data propriétaire des deux répertoires static et media:

```
$ sudo chown -R www-data:www-data /var/www/UE-environment/updatengine-server/updatengine/static/
```

```
$ sudo chown -R www-data:www-data /var/www/UE-environment/updatengine-server/updatengine/media/
```

## 2.1.4 - Patch de Django 1.6.2

Afin de corriger un bug Django pouvant (dans certain cas) conduire à l'enregistrement de mauvais enregistrements, veuillez passer le patch nommé patch-django\_1.6.2\_17118.patch :

```
$ sudo patch -p1 /var/www/UE-environment/local/lib/python2.7/site-packages/django/forms/models.py /var/www/UE-environment/updatengine-server/requirements/patches/patch-django_1.6.2_17118.patch
```

Après avoir lancé la commande, le shell doit vous indiquer :

```
patching file /var/www/UE-environment/local/lib/python2.7/site-packages/django/forms/models.py
```

## 2.1.5 - Fin de configuration

Vous pouvez maintenant relancer le service apache2:

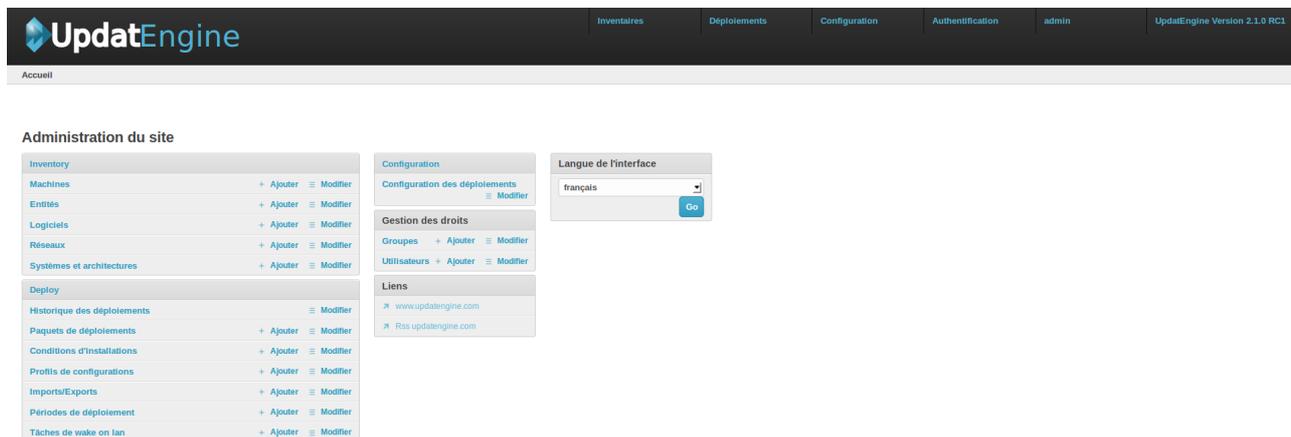
```
$ sudo service apache2 restart
```

Enfin, connectez-vous à l'application avec un navigateur web en saisissant comme adresse celle que vous avez renseigné pour la variable `PROJECT_URL` (`http://192.168.1.45:1979` dans l'exemple).

Vous arrivez alors sur la boîte de connexion. Connectez-vous avec le login et mot de passe renseigné précédemment :



Vous accédez alors à l'interface d'administration du serveur UpdatEngine :



Votre serveur UpdatEngine est maintenant installé.

## 2.2 - Installation d'UpdatEngine Server (Debian / Ubuntu – apache2/SQLite)

### 2.2.1 - Installation de l'environnement et des dépendances nécessaires

Installez tout d'abord les différentes dépendances nécessaires:

```
$ sudo apt-get install apache2 libapache2-mod-wsgi python-virtualenv python-pip libxml2-dev libxslt-dev python-dev libsqlite3-dev git-core
```

Créez maintenant l'environnement virtuel qui hébergera le serveur UpdatEngine. Nous le créerons directement sous /var/www dans un répertoire nommé UE-environment (veillez à respecter le nom de ce répertoire pour pouvoir profiter du fichier de configuration apache mis à disposition dans cet article).

```
$ sudo virtualenv --distribute --no-site-packages /var/www/UE-environment
$ cd /var/www/UE-environment/
$ sudo git clone https://github.com/updatengine/updatengine-server.git
```

L'ensemble des fichiers du serveur UpdatEngine sont alors disponibles sous /var/www/UE-environment/updatengine-server

Installons maintenant les dépendances nécessaires à UpdatEngine dans votre environnement virtuel:

### 2.2.2 - Installation des dépendances dans l'environnement virtuel UE-environment

Les différentes dépendances sont détaillées dans le fichier *pip-packages-sqlite.txt*

Installez les avec l'utilitaire /var/www/UE-environment/bin/pip install ce qui permettra de n'installer les dépendances que dans votre environnement virtuel et dans les versions spécifiées.

```
$ sudo /var/www/UE-environment/bin/pip install --upgrade distribute setuptools
$ sudo /var/www/UE-environment/bin/pip install -r /var/www/UE-environment/updatengine-server/requirements/pip-packages-sqlite.txt
```

A l'issue de l'exécution de la commande, vous disposez de tout le nécessaire pour faire fonctionner l'application.

**Remarque:** si vous êtes positionné derrière un proxy, pensez à utiliser l'option --proxy=http://votre\_proxy:port\_proxy

Par exemple :

```
$ sudo /var/www/UE-environment/bin/pip install --proxy=http://mon_proxy:8080 --upgrade distribute setuptools
```

## 2.2.3 - Initialisation de la base de données SQLite

Copiez tout d'abord le fichier `settings.py.model_sqlite` vers `settings.py`:

```
$ sudo cp /var/www/UE-environment/updatengine-server/updatengine/settings.py.model_sqlite /var/www/UE-environment/updatengine-server/updatengine/settings.py
```

Donnez ensuite les droits à l'utilisateur apache pour pouvoir écrire dans la base de données: `sudo`

```
$ sudo chown -R www-data:www-data /var/www/UE-environment/updatengine-server/updatengine/db
sudo chmod -R 775 /var/www/UE-environment/updatengine-server/updatengine/db
```

Vous pouvez maintenant modifier le fichier de configuration `/var/www/UE-environment/updatengine-server/updatengine/settings.py` pour y indiquer l'adresse de votre serveur et les paramètres de connexion à la base de données.

Ouvrez le fichier `/var/www/UE-environment/updatengine-server/updatengine/settings.py` et renseignez la variable **PROJECT\_URL** avec l'adresse et le port de votre serveur.

Par exemple, si l'adresse ip de votre serveur est **192.168.1.45**, vous devez renseigner la ligne de cette façon:

```
PROJECT_URL = 'http://192.168.1.45:1979'
```

Après avoir fait cette modifications, sauvegardez et fermez ce fichier.

**Remarque:** ces informations correspondent à l'exemple pris ci-dessus lors de la création de la base de données et de son utilisateur. Il faudra donc l'adapter à votre contexte.

Avant de redémarrer le service apache2, copiez le fichier de configuration standard puis activez ce nouveau site:

```
$ sudo cp /var/www/UE-environment/updatengine-server/requirements/apache-updatengine.conf /etc/apache2/sites-available/apache-updatengine.conf
```

```
$ sudo a2ensite apache-updatengine
```

Activez également le mode wsgi si ce n'est pas déjà fait:

```
$ sudo a2enmod wsgi
```

Redémarrez maintenant le service apache2 pour prendre en compte la nouvelle configuration:

```
$ sudo service apache2 restart
```

Puis initialisez la base de données:

```
$ sudo /var/www/UE-environment/bin/python /var/www/UE-environment/updatengine-server/manage.py syncdb
```

L'installateur vous demande alors si vous souhaitez créer un nouvel utilisateur pour l'application: Répondez **yes** à la question posée pour en créer un nouveau:

*You just installed Django's auth system, which means you don't have any superusers defined. Would you like to create one now? (yes/no): **yes***

Saisissez ensuite le nom de l'utilisateur et entrez son adresse email:

*Username (leave blank to use 'root'): **admin***

*E-mail address: **admin@local.domain***

Saisissez ensuite un mot de passe pour cet utilisateur, confirmez le et validez par entrée pour terminer l'initialisation de la base de données:

*Password:*

*Password (again):*

La base de données est maintenant initialisée.

Pour terminer sa configuration, nous allons maintenant passer toutes les mises à jour effectuées depuis le début de son développement avec la commande migrate:

```
$ sudo /var/www/UE-environment/bin/python /var/www/UE-environment/updatengine-server/manage.py migrate
```

**Remarque:** le schéma de base de données est géré avec le module python South pour faciliter les évolutions et les migrations des bases de données entre versions.

Enfin chargez les données initiales de l'application (configuration par défaut):

```
$ sudo /var/www/UE-environment/bin/python /var/www/UE-environment/updatengine-server/manage.py loaddata /var/www/UE-environment/updatengine-server/initial_data/configuration_initial_data.yaml
```

```
$ sudo /var/www/UE-environment/bin/python /var/www/UE-environment/updatengine-server/manage.py loaddata /var/www/UE-environment/updatengine-server/initial_data/groups_initial_data.yaml
```

L'installation de l'application et l'initialisation de la base de données est maintenant terminée. Il ne reste plus qu'à rendre l'utilisateur www-data propriétaire des deux répertoires static et media:

```
$ sudo chown -R www-data:www-data /var/www/UE-environment/updatengine-server/updatengine/static/
```

```
$ sudo chown -R www-data:www-data /var/www/UE-environment/updatengine-server/updatengine/media/
```

## 2.2.4 - Patch de Django 1.6.2

Afin de corriger un bug Django pouvant (dans certain cas) conduire à l'enregistrement de mauvais

enregistrements, veuillez passer le patch nommé patch-django\_1.6.2\_17118.patch :

```
$ sudo patch -p1 /var/www/UE-environment/local/lib/python2.7/site-packages/django/forms/models.py /var/www/UE-environment/updatengine-server/requirements/patches/patch-django_1.6.2_17118.patch
```

Après avoir lancé la commande, le shell doit vous indiquer :

```
patching file /var/www/UE-environment/local/lib/python2.7/site-packages/django/forms/models.py
```

## 2.2.5 - Fin de configuration

Vous pouvez maintenant relancer le service apache2:

```
$ sudo service apache2 restart
```

Enfin, connectez vous à l'application avec un navigateur web en saisissant comme adresse celle que vous avez renseigné pour la variable PROJECT\_URL (http://192.168.1.45:1979 dans l'exemple).

Vous arrivez alors sur la boîte de connexion. Connectez-vous avec le login et mot de passe renseigné précédemment :



Votre serveur UpdatEngine est maintenant installé.

## 3 - Migration d'un serveur existant vers la version 2.1.1 stable

Cette procédure a pour but de migrer un serveur UpdatEngine dans la toute nouvelle version 2.1.1 stable.

### 3.1 - Sauvegarde de l'application

Comme pour toute application à mettre à jour, commencez par faire une sauvegarde de la base de données et des fichiers de l'application (de façon à préserver vos paquets).

Vous pouvez également sauvegarder le répertoire contenant les paquets de déploiement situés dans le répertoire `/var/www/UE-environment/updatengine-server/updatengine/media/package-file/`

### 3.2 - Procédure de mise à jour

#### 3.2.1 - Mise à jour et bascule sur la branche stable (nommée master) :

Placez-vous dans le répertoire `updatengine-server` et assurez-vous tout d'abord de bien être en version stable (master)

```
$ cd /var/www/UE-environment/updatengine-server/  
$ sudo git checkout master
```

Lancez ensuite la commande `git pull` qui permettra de synchroniser votre installation avec les sources hébergées en ligne:

```
$ sudo git pull
```

#### 3.2.2 - Mise à jour des modules:

##### **Si vous utilisez le moteur de base de données MySQL :**

Commencez par mettre à jour les dépendances nécessaires à l'application :

```
$ cd /var/www/UE-environment/  
  
$ sudo /var/www/UE-environment/bin/pip install --upgrade distribute  
$ sudo /var/www/UE-environment/bin/pip install --upgrade setuptools  
  
$ sudo /var/www/UE-environment/bin/pip install -r /var/www/UE-environment/updatengine-server/requirements/pip-packages.txt
```

Puis, chargez la table des fuseaux horaires :

```
$ mysql_tzinfo_to_sql /usr/share/zoneinfo | mysql -u root -p mysql
$ sudo service mysql restart
```

**Remarque:** si vous êtes positionné derrière un proxy, pensez à utiliser l'option `--proxy=http://votre_proxy:port_proxy`

Par exemple :

```
$ sudo /var/www/UE-environment/bin/pip install --proxy=http://mon\_proxy:8080
--upgrade distribute
```

### **Si vous utilisez le moteur de base de données SQLite :**

Mettez à jour les dépendances nécessaires de l'application :

```
$ cd /var/www/UE-environment/
$ sudo /var/www/UE-environment/bin/pip install --upgrade distribute setuptools
$ sudo /var/www/UE-environment/bin/pip install -r /var/www/UE-
environment/updatengine-server/requirements/pip-packages-sqlite.txt
```

**Remarque:** si vous êtes positionné derrière un proxy, pensez à utiliser l'option `--proxy=http://votre_proxy:port_proxy`

Par exemple :

```
$ sudo /var/www/UE-environment/bin/pip install --proxy=http://mon\_proxy:8080
--upgrade distribute setuptools
```

### **3.2.3 - Mise à jour de la base**

Vérifiez tout d'abord que les noms des entités de votre base sont différentes car un critère d'unicité sur leurs noms est maintenant positionné.

Vous pouvez maintenant procéder à la mise à jour de la base de données:

```
$ sudo /var/www/UE-environment/bin/python /var/www/UE-environment/updatengine-
server/manage.py syncdb
puis
$ sudo /var/www/UE-environment/bin/python /var/www/UE-environment/updatengine-
server/manage.py migrate
```

### **3.2.4 - Adaptation du fichier settings.py**

Ouvrez le fichier `/var/www/UE-environment/updatengine-server/updatengine/settings.py`

Ajoutez la ligne `SHOW_PERM_CONFIG_AUTH = False` après l'option `LANGUAGE_CODE = 'fr'`

Votre fichier doit ressembler à ceci

```
# Language code for this installation. All choices can be found here:  
# http://www.i18nguy.com/unicode/language-identifiers.html  
LANGUAGE_CODE = 'fr'  
  
SHOW_PERM_CONFIG_AUTH = False
```

### 3.2.5 - Chargement des groupes par défaut

Afin de charger les nouveaux groupes mis à disposition avec cette version, lancez la commande suivante :

```
$ sudo /var/www/UE-environment/bin/python /var/www/UE-environment/updatengine-server/manage.py loaddata /var/www/UE-environment/updatengine-server/initial_data/groups_initial_data.yaml
```

### 3.2.6 - Patch de Django 1.6.2

Afin de corriger un bug Django pouvant (dans certain cas) conduire à l'enregistrement de mauvais enregistrements, veuillez passer le patch nommé patch-django\_1.6.2\_17118.patch :

```
$ sudo patch -p1 /var/www/UE-environment/local/lib/python2.7/site-packages/django/forms/models.py /var/www/UE-environment/updatengine-server/requirements/patches/patch-django_1.6.2_17118.patch
```

Après avoir lancé la commande, le shell doit vous indiquer :

```
patching file /var/www/UE-environment/local/lib/python2.7/site-packages/django/forms/models.py
```

### 3.2.7 - Redémarrage d'apache2

Enfin, relancez le service apache2 :

```
$ sudo service apache2 restart
```

Vous pouvez maintenant vous connecter à l'application. Après la connexion, vous devriez avoir une interface de ce type :



Votre serveur UpdatEngine est maintenant à jour.

