Installation de la plateforme Pod V3

Les commandes suivantes ont été lancées sur une distribution Debian 11.4

- Environnement
  - Création de l'utilisateur Pod
  - Installation de paquets
  - Création de l'environnement virtuel
  - Récupération des sources
- Applications tierces
  - Installation de toutes les librairies python
  - FFMPEG
  - Redis
  - Elasticsearch
    - Elasticsearch 6
    - Elasticsearch 7 et 8
    - Il est recommandé d'utiliser le mode security d'ES8.
    - Lancement et vérification d'Elasticsearch
    - Création de l'index Pod
  - Installation des dépendances
- Mise en route
  - Base de données SQLite intégrée
  - Fichier de configuration settings_local.py
  - SuperUtilisateur
  - Lancement des tests unitaires
  - Modules complémentaires
  - Serveur de développement
  - - Attention -

Environnement

Création de l'utilisateur Pod

```
user@pod:~$  sudo adduser pod
user@pod:~$  adduser pod sudo
user@pod:~$  su pod
user@pod:~$  cd /home/pod
```

Installation de paquets

```
pod@pod:~$ sudo apt-get install git curl vim
```

Création de l'environnement virtuel

```
pod@pod:~$ sudo python3 -V
pod@pod:~$ sudo python -V
pod@pod:~$ sudo apt-get install -y python3-pip
pod@pod:~$ sudo pip3 install virtualenvwrapper
```

À la fin du .bashrc, il faut ajouter ces lignes :

```
pod@pod:~$ vim .bashrc
      [..]
      export WORKON_HOME=$HOME/.virtualenvs
      export VIRTUALENVWRAPPER_PYTHON=/usr/bin/python3
      source /usr/local/bin/virtualenvwrapper.sh
      [..]
```

Puis prendre en charge ces modifications :

```
pod@pod:$ source .bashrc
```

Et enfin créez l'environnement virtuel :

```
pod@pod:~$ mkvirtualenv --system-site-packages --python=/usr/bin/python3 django_pod3
```

Récupération des sources

Concernant l'emplacement du projet, je conseille de le mettre dans `/usr/local/django_projects`

```
(django_pod3)pod@pod:~$ sudo mkdir /usr/local/django_projects
```

Vous pouvez faire un lien symbolique dans votre home pour arriver plus vite dans le répertoire `django_projects` :

```
(django_pod3)pod@pod:~$ ln -s /usr/local/django_projects django_projects
```

Placez-vous dans le répertoire `django_projects`

```
(django_pod3)pod@pod:~$ cd django_projects
```

Donnez les droits à l'utilisateur pod de lire et d'écrire dans le répertoire :

```
(django_pod3) pod@pod:~/django_projects$ sudo chown pod:pod /usr/local/django_projects
```

Vous pouvez enfin récupérer les sources :

Remarque

Si vous devez utiliser un proxy, vous pouvez le spécifier avec cette commande :

```
(django_pod3) pod@pod:~/django_projects$ git config --global http.proxy http://PROXY:PORT
```

La récupération des sources de la V3 se fait via cette commande : `git clone https://github.com/EsupPortail/Esup-Pod.git podv3`

```
(django_pod3) pod@pod:~/django_projects$ git clone https://github.com/EsupPortail/Esup-Pod.git podv3
Clonage dans 'podv3'...
remote: Counting objects: 4578, done.
remote: Compressing objects: 100% (378/378), done.
remote: Total 4578 (delta 460), reused 564 (delta 348), pack-reused 3847
Réception d'objets: 100% (4578/4578), 4.40 MiB | 3.88 MiB/s, fait.
Résolution des deltas: 100% (3076/3076), fait.
```

```
(django_pod3) pod@pod:~/django_projects$ cd podv3/
```

Applications tierces

Installation de toutes les librairies python

Il faut vérifier que l'on se trouve bien dans l'environnement virtuel (présence de "(django_pod3)" au début l'invite de commande. Sinon, il faut lancer la commande `$> workon django_pod3`

```
(django_pod3) pod@pod:~/django_projects/podv3$ pip3 install -r requirements.txt
```

De même, si vous devez utiliser un proxy :

```
(django_pod3) pod@pod:~/django_projects/podv3$ pip3 install --proxy="PROXY:PORT" -r requirements.txt
```

FFMPEG

Pour l'encodage des vidéos et la creation des vignettes, il faut installer **ffmpeg**, **ffmpegthumbnailer** et **imagemagick** (ne pas installer sur le serveur frontal si vous déportez l'encodage)

```
(django_pod3) pod@pod:~/django_projects/podv3$ sudo apt install ffmpeg ffmpegthumbnailer imagemagick
```

Redis

Voir la doc officielle https://redis.io/docs/getting-started/

Pour installer le cache Redis

```
(django_pod3) pod@pod:~/django_projects/podv3$ sudo apt install redis-server
```

En théorie le service démarre automatiquement. Si vous avez installé Redis sur la même machine que Pod, rien à faire de plus. Pour vérifier si le service est bien démarré :

```
(django_pod3) pod@pod:~/django_projects/podv3$ sudo service redis-server status
```

Si vous utilisez Redis sur une autre machine, n'oubliez pas de modifier le bind dans le fichier de configuration `/etc/redis/redis.conf`

Si vous ne souhaitez pas toucher au bind, vous pouvez aussi **modifier votre settings_local.py** et personnaliser cette extrait du settings.py par défaut avec votre <my_redis_host>

```
CACHES = {
    "default": {
        "BACKEND": "django_redis.cache.RedisCache",
        "LOCATION": "redis://<my_redis_host>:6379/3",
        "OPTIONS": {
            "CLIENT_CLASS": "django_redis.client.DefaultClient",
        },
        "KEY_PREFIX": "pod",
    },
    "select2": {
        "BACKEND": "django_redis.cache.RedisCache",
        "LOCATION": "redis://<my_redis_host>:6379/2",
        "OPTIONS": {
            "CLIENT_CLASS": "django_redis.client.DefaultClient",
        },
    },
}
SESSION_ENGINE = "redis_sessions.session"
SESSION_REDIS = {
    "host": "<my_redis_host>",
    "port": 6379,
    "db": 4,
    "prefix": "session",
    "socket_timeout": 1,
    "retry_on_timeout": False,
}
```

Elasticsearch

Selon la version d'*Elasticsearch* que vous allez utiliser, les versions des dépendances peuvent changer. Les versions 6 et 7 ne sont actuellement plus maintenues. La version 8 est à configurer plus bas. La version 6 est celle par défaut dans Pod.

Elasticsearch 6

Pour utiliser *Elasticsearch 6*, il faut avoir java 11 sur sa machine.

```
(django_pod3) pod@pod:~/django_projects/podv3$ sudo apt-get install openjdk-11-jre
```

Puis pour installer *Elasticsearch* sur Debian en utilisant les paquets, il faut suivre les instructions situées à cette adresse : https://www.elastic.co/guide/en/elasticsearch/reference/current/deb.html.

Voici :

```
(django_pod3) pod@pod:~/django_projects/podv3$ wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch |
sudo apt-key add -
OK
(django_pod3) pod@pod:~/django_projects/podv3$ sudo apt-get install apt-transport-https
(django_pod3) pod@pod:~/django_projects/podv3$ echo "deb https://artifacts.elastic.co/packages/6.x/apt stable
main" | sudo tee -a /etc/apt/sources.list.d/elastic-6.x.list deb https://artifacts.elastic.co/packages/6.x/apt
stable main
(django_pod3) pod@pod:~/django_projects/podv3$ sudo apt-get update && sudo apt-get install elasticsearch
```

Ensuite il faut paramétrer l'instance :

```
(django_pod3) pod@pod:~/django_projects/podv3$ sudo vim /etc/elasticsearch/elasticsearch.yml
```

Puis préciser ces valeurs :

```
cluster.name: pod-application
node.name: pod-1
discovery.zen.ping.unicast.hosts: ["127.0.0.1"]
```

Elasticsearch 7 et 8

Pour utiliser *Elasticsearch 7 ou 8*, il faut avoir java 17 sur sa machine.

```
(django_pod3) pod@pod:~/django_projects/podv3$ sudo apt-get install default-jdk
```

Puis pour installer *Elasticsearch* sur Debian en utilisant les paquets, il faut suivre les instructions situées à cette adresse : https://www.elastic.co/guide/en/elasticsearch/reference/current/deb.html.

Vous pouvez installer Elasticsearch en version 7 (plus maintenue) ou en version 8.

Voici pour ES7 :

```
(django_pod3) pod@pod:~/django_projects/podv3$ wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch |
sudo gpg --dearmor -o /usr/share/keyrings/elasticsearch-keyring.gpg
 OK
(django_pod3) pod@pod:~/django_projects/podv3$ sudo apt-get install apt-transport-https
(django_pod3) pod@pod:~/django_projects/podv3$ echo "deb [signed-by=/usr/share/keyrings/elasticsearch-keyring.
gpg] https://artifacts.elastic.co/packages/7.x/apt stable main" | sudo tee /etc/apt/sources.list.d/elastic-7.x.
list
(django_pod3) pod@pod:~/django_projects/podv3$ sudo apt-get update && sudo apt-get install elasticsearch
```

Voici pour ES8 :

```
(django_pod3) pod@pod:~/django_projects/podv3$ wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch |
sudo gpg --dearmor -o /usr/share/keyrings/elasticsearch-keyring.gpg
 OK
(django_pod3) pod@pod:~/django_projects/podv3$ sudo apt-get install apt-transport-https
(django_pod3) pod@pod:~/django_projects/podv3$ echo "deb [signed-by=/usr/share/keyrings/elasticsearch-keyring.
gpg] https://artifacts.elastic.co/packages/8.x/apt stable main" | sudo tee /etc/apt/sources.list.d/elastic-8.x.
list
(django_pod3) pod@pod:~/django_projects/podv3$ sudo apt-get update && sudo apt-get install elasticsearch
```

Ensuite il faut paramétrer l'instance :

```
(django_pod3) pod@pod:~/django_projects/podv3$ sudo vim /etc/elasticsearch/elasticsearch.yml
```

Puis préciser ces valeurs :

```
cluster.name: pod-application
node.name: pod-1
discovery.seed_hosts: ["127.0.0.1"]
cluster.initial_master_nodes: ["pod-1"]
```

Il est recommandé d'utiliser le mode security d'ES8.

Générer l'utilisateur pod pour ES :

```
sudo /usr/share/elasticsearch/bin/elasticsearch-users useradd pod -p podpod -r superuser
```

Génération des certificats (CA + cert) :

```
sudo /usr/share/elasticsearch/bin/elasticsearch-certutil ca
sudo /usr/share/elasticsearch/bin/elasticsearch-certutil cert --ca elastic-stack-ca.p12

sudo /usr/share/elasticsearch/bin/elasticsearch-keystore add xpack.security.http.ssl.keystore.secure_password
sudo /usr/share/elasticsearch/bin/elasticsearch-keystore add xpack.security.http.ssl.truststore.secure_password
```

Copier le fichier .p12 dans /etc/elasticsearch/

```
sudo cp /usr/share/elasticsearch/elastic-stack-ca.p12 /usr/share/elasticsearch/elastic-certificates.p12 /etc
/elasticsearch/
sudo chown pod:pod /etc/elasticsearch/elastic-stack-ca.p12 /etc/elasticsearch/elastic-certificates.p12
sudo chmod +r /etc/elasticsearch/elastic-stack-ca.p12 /etc/elasticsearch/elastic-certificates.p12
```

Dans */etc/elasticsearch/elasticsearch.yml* :

```
xpack.security.http.ssl.enabled: true
xpack.security.http.ssl.verification_mode: certificate
xpack.security.http.ssl.keystore.path: /etc/elasticsearch/elastic-certificates.p12
xpack.security.http.ssl.truststore.path: /etc/elasticsearch/elastic-certificates.p12
```

Lancement et vérification d'Elasticsearch

Il faut enfin le lancer et vérifier son bon fonctionnement :

```
(django_pod3) pod@pod:~/django_projects/podv3$ sudo /etc/init.d/elasticsearch start
```

```
(django_pod3) pod@pod:~/django_projects/podv3$ curl -XGET "127.0.0.1:9200"
```

ou pour ES8 `(django_pod3) pod@pod:~/django_projects/podv3$ curl -k -XGET 'https://127.0.0.1:9200' -u pod:podpod`

```
{
 "name" : "pod-1",
"cluster_name" : "pod-application",
  "cluster_uuid" : "5yhs9zc4SRyjaKYyW7uabQ",
  "version" : {
    "number" : "8.4.0",
    "build_flavor" : "default",
    "build_type" : "deb",
    "build_hash" : "f56126089ca4db89b631901ad7cce0a8e10e2fe5",
    "build_date" : "2022-08-19T19:23:42.954591481Z",
    "build_snapshot" : false,
    "lucene_version" : "9.3.0",
    "minimum_wire_compatibility_version" : "7.17.0",
    "minimum_index_compatibility_version" : "7.0.0"
  },
  "tagline" : "You Know, for Search"
}
```

Pour utiliser la recherche dans Pod, nous allons avoir besoin également du plugin ICU :

```
(django_pod3) pod@pod:~/django_projects/podv3$ cd /usr/share/elasticsearch/
(django_pod3) pod@pod:/usr/share/elasticsearch$ sudo bin/elasticsearch-plugin install analysis-icu
-> Downloading analysis-icu from elastic
[=================================================] 100%
-> Installed analysis-icu
(django_pod3) pod@pod:/usr/share/elasticsearch$ sudo /etc/init.d/elasticsearch restart
[ ok ] Restarting elasticsearch (via systemctl): elasticsearch.service.
```

Si vous utilisez derrière un proxy :

```
(django_pod3) pod@pod:~/django_projects/podv3$ cd /usr/share/elasticsearch/
(django_pod3) pod@pod:/usr/share/elasticsearch$ sudo ES_JAVA_OPTS="-Dhttp.proxyHost=proxy.univ.fr -Dhttp.
proxyPort=3128 -Dhttps.proxyHost=proxy.univ.fr -Dhttps.proxyPort=3128" /usr/share/elasticsearch/bin
/elasticsearch-plugin install analysis-icu
 -> Downloading analysis-icu from elastic
[=================================================] 100%
-> Installed analysis-icu
(django_pod3) pod@pod:/usr/share/elasticsearch$ sudo /etc/init.d/elasticsearch restart
[ ok ] Restarting elasticsearch (via systemctl): elasticsearch.service.
```

Attention, pour ES8 derrière un proxy :

```
(django_pod3) pod@pod:~/django_projects/podv3$ cd /usr/share/elasticsearch/
(django_pod3) pod@pod:/usr/share/elasticsearch$ sudo CLI_JAVA_OPTS="-Dhttp.proxyHost=proxy.univ.fr -Dhttp.
proxyPort=3128 -Dhttps.proxyHost=proxy.univ.fr -Dhttps.proxyPort=3128" /usr/share/elasticsearch/bin
/elasticsearch-plugin install analysis-icu
 -> Downloading analysis-icu from elastic
[=================================================] 100%
-> Installed analysis-icu
(django_pod3) pod@pod:/usr/share/elasticsearch$ sudo /etc/init.d/elasticsearch restart
[ ok ] Restarting elasticsearch (via systemctl): elasticsearch.service.
```

Création de l'index Pod

Pour une utilisation d'elasticsearch 7 ou 8, il faut absolument :

Ajouter **ES_VERSION = 7 ou ES_VERSION = 8** dans votre fichier de settings et modifiez la version du client elasticsearch dans le fichier **requirements.txt**

elasticsearch==7.17.9

ou

elasticsearch==8.9.0

Et ne pas oublier de relancer

```
(django_pod3) pod@podv3:~/django_projects/podv3$ pip3 install -r requirements.txt
```

Nous pouvons enfin vérifier le bon fonctionnement de l'ensemble (l'erreur affichée lors de la deletion est normale puisque l'indice n'existe pas, mais nous devons supprimer avant de créer un index dans ES) :

```
(django_pod3) pod@pod:/usr/share/elasticsearch$ cd ~/django_projects/podv3
(django_pod3) pod@pod:~/django_projects/podv3$ python manage.py create_pod_index
DELETE http://127.0.0.1:9200/pod [status:404 request:0.140s]
An error occured during index video deletion: 404-index_not_found_exception : no such index
Successfully create index Video
(django_pod3) pod@pod:~/django_projects/podv3$ curl -XGET "127.0.0.1:9200/pod/_search"
{"took":35,"timed_out":false,"_shards":{"total":2,"successful":2,"skipped":0,"failed":0},"hits":{"total":0,"
max_score":null,"hits":[]}}
```

Si la commande python ne fonctionne pas, créez d'abord l'index à la main avec un *curl -XPUT "http://127.0.0.1:9200/pod"* (options *-k --noproxy -u <user>:<pwd>* à prévoir si ES8 en mode security)

Si vous déportez l'elastic search sur une autre machine, rajoutez dans le fichier settings_local.py son URL d'accès :

```
(django_pod3) pod@pod:~/django_projects/podv3$ vim pod/custom/settings_local.py
```

Copiez la ligne suivante :

```
ES_URL = ['http://elastic.domaine.fr:9200/']
```

Avec le mode security et ES8, vous devrez parametrer les éléments suivants dans votre *settings_local.py* :

```
ES_URL = ['https://127.0.0.1:9200/'] # ou votre instance déportée
ES_OPTIONS = {'verify_certs' : False, 'basic_auth' : ('es_user', 'password')}
```

Installation des dépendances

Pour installer les dépendances, il faut en premier installer nodejs, npm et yarn. L'installation ne fonctionne pas avec nodejs 12.x

La référence est ici: https://github.com/nodesource/distributions

```
(django_pod3) pod@pod:~/django_projects/podv3$ sudo apt-get update
(django_pod3) pod@pod:~/django_projects/podv3$ sudo apt-get install -y ca-certificates curl gnupg
(django_pod3) pod@pod:~/django_projects/podv3$ sudo mkdir -p /etc/apt/keyrings
(django_pod3) pod@pod:~/django_projects/podv3$ curl -fsSL https://deb.nodesource.com/gpgkey/nodesource-repo.gpg.
key | sudo gpg --dearmor -o /etc/apt/keyrings/nodesource.gpg
(django_pod3) pod@pod:~/django_projects/podv3$ NODE_MAJOR=18 && echo "deb [signed-by=/etc/apt/keyrings
/nodesource.gpg] https://deb.nodesource.com/node_$NODE_MAJOR.x nodistro main" | sudo tee /etc/apt/sources.list.d
/nodesource.list
(django_pod3) pod@pod:~/django_projects/podv3$ sudo apt update && apt install -y nodejs
(django_pod3) pod@pod:~/django_projects/podv3$ corepack enable
```

CentOS

Alternativement, si vous êtes sur CentOS 8, installez les dépendances nodejs+npm et yarn ainsi :

```
root@pod:~/$ dnf module reset nodejs
root@pod:~/$ dnf module enable nodejs:14
root@pod:~/$ dnf module -y update nodejs
root@pod:~/$ yum install nodejs
root@pod:~/$ npm install yarn -g
```

Se placer dans le répertoire où est installé le package.json

```
(django_pod3) pod@pod:~/django_projects/podv3$ cd pod
```

Installez les dépendances.

```
(django_pod3) pod@pod:~/django_projects/podv3/pod$ yarn
```

Enfin, déployez les fichiers statiques.

```
(django_pod3) pod@pod:~/django_projects/podv3$ python manage.py collectstatic --no-input --clear
```

Mise en route

Base de données SQLite intégrée

Lancez le script présent à la racine afin de créer les fichiers de migration, puis de les lancer pour créer la base de données SQLite intégrée.

```
(django_pod3) pod@Pod:~/django_projects/podv3$ make createDB
```

Fichier de configuration settings_local.py

Vous devez créer un fichier de configuration local dans le dossier pod/custom.

Vous mettez dans ce fichier uniquement les variables dont vous voulez changer la valeur par défaut. Vous trouverez ci-dessous un exemple de fichier avec les principales variables à modifier : connexion à la base de données, un fichier CSS custom, le thème green de pod, retirer le langage nl, etc. Vous pouvez adapter ce fichier et le coller dans le vôtre.

```
(django_pod3) pod@Pod:~/django_projects/podv3$ vim pod/custom/settings_local.py
```

```
"""Django local settings for pod_project.Django version : 3.2"""

##
# The secret key for your particular Django installation.
#
# This is used to provide cryptographic signing,
# and should be set to a unique, unpredictable value.
#
# Django will not start if this is not set.
# https://docs.djangoproject.com/fr/3.2/ref/settings/#secret-key
#
# SECURITY WARNING: keep the secret key used in production secret!
# You can visit https://djecrety.ir/ to get it
SECRET_KEY = 'A_CHANGER'

##
# DEBUG mode activation
#
# https://docs.djangoproject.com/fr/3.2/ref/settings/#debug
#
# SECURITY WARNING: MUST be set to False when deploying into production.
DEBUG = True

##
# A list of strings representing the host/domain names
# that this Django site is allowed to serve.
#
# https://docs.djangoproject.com/fr/3.2/ref/settings/#allowed-hosts
ALLOWED_HOSTS = ['localhost']

##
# A tuple that lists people who get code error notifications
#   when DEBUG=False and a view raises an exception.
#
# https://docs.djangoproject.com/fr/3.2/ref/settings/#std:setting-ADMINS
#
ADMINS = (
    ('Name', 'adminmail@univ.fr'),
)


##
# Internationalization and localization.
#
# https://docs.djangoproject.com/fr/3.2/topics/i18n/
# https://github.com/django/django/blob/master/django/conf/global_settings.py
LANGUAGES = (
    ('fr', 'Français'),
    ('en', 'English')
)

#Hide Users in navbar
HIDE_USER_TAB = True
# Hide Types tab in navbar
HIDE_TYPES_TAB = True
# Hide Tags
HIDE_TAGS = True
# Hide disciplines in navbar
HIDE_DISCIPLINES = True

##
# eMail settings
```

```
#
# https://docs.djangoproject.com/fr/3.2/ref/settings/#email-host
# https://docs.djangoproject.com/fr/3.2/ref/settings/#email-port
# https://docs.djangoproject.com/fr/3.2/ref/settings/#default-from-email
#
#   username: EMAIL_HOST_USER
#   password: EMAIL_HOST_PASSWORD
#
EMAIL_HOST = 'smtp.univ.fr'
EMAIL_PORT = 25
DEFAULT_FROM_EMAIL = 'noreply@univ.fr'

# https://docs.djangoproject.com/fr/3.2/ref/settings/#std:setting-SERVER_EMAIL
SERVER_EMAIL = 'noreply@univ.fr'

##
# THIRD PARTY APPS OPTIONNAL
#
USE_PODFILE = True

##
# TEMPLATE Settings
#
TEMPLATE_VISIBLE_SETTINGS = {

    'TITLE_SITE': 'Lille.Pod',
    'TITLE_ETB': 'Université de Lille',
    'LOGO_SITE': 'img/logoPod.svg',
    'LOGO_COMPACT_SITE': 'img/logoPod.svg',
    'LOGO_ETB': 'img/logo_etb.svg',
    'LOGO_PLAYER': 'img/logoPod.svg',
    'FOOTER_TEXT': (
        '42, rue Paul Duez',
        '59000 Lille - France',
        ('<a href="https://goo.gl/maps/AZnyBK4hHaM2"'
            ' target="_blank">Google maps</a>')
    ),
    'LINK_PLAYER': 'http://www.univ-lille.fr',
    'CSS_OVERRIDE': 'custom/mycss.css',
    'PRE_HEADER_TEMPLATE': ''
}

##
# A string representing the time zone for this installation. (default=UTC)
#
# https://en.wikipedia.org/wiki/List_of_tz_database_time_zones
TIME_ZONE = 'Europe/Paris'
```

Vous trouverez l'ensemble des variables disponibles sur cette page :

Configuration de la plateforme

SuperUtilisateur

Il faut créer un premier utilisateur qui aura tous les pouvoirs sur votre instance.

```
(django_pod3) pod@Pod:~/django_projects/podv3$ python manage.py createsuperuser
```

Lancement des tests unitaires

Afin de vérifier que votre instance est opérationnelle, vous pouvez lancer les tests unitaires :

```
(django_pod3) pod@Pod:~/django_projects/podv3$ python3 -m pip install -r requirements-dev.txt
(django_pod3) pod@Pod:~/django_projects/podv3$ python manage.py test --settings=pod.main.test_settings
```

Modules complémentaires

Ce paragraphe a été rédigé pour pod v2.9.x, il est possible que cela ne concerne pas pod v3

Il existe plusieurs modules complémentaires pouvant être ajoutés à pod (enrichissement, diffusion en direct, etc). Pour activer ces modules, il faut ajouter la ligne suivante au **settings_local.py**

settings_local.py

```
##
# THIRD PARTY APPS OPTIONNAL
#
THIRD_PARTY_APPS = ["live", "enrichment"]
```

Enrichissement

Selon votre paramétrage système, il est possible que les enrichissements renvoient une erreur 403 forbidden.

Pour éviter ce problème, vous pouvez rajouter cette ligne dans **settings_local.py**

settings_local.py

```
FILE_UPLOAD_PERMISSIONS = 0o644 # Octal number
```

Serveur de développement

Le serveur de développement permet de tester vos futures modifications facilement.

N'hésitez pas à lancer le serveur de développement pour vérifier vos modifications au fur et à mesure.

À ce niveau, vous devriez avoir le site en français et en anglais et voir l'ensemble de la page d'accueil.

```
(django_pod3) pod@Pod:~/django_projects/podv3$ python3 manage.py runserver @IP/DNS:8080 --insecure
```

```
--> exemple : (django_pod3) pod@pod:~/django_projects/podv3$ python manage.py runserver pod.univ.fr:8080 --
insecure
```

- Attention -

> ***Quand le site est lancé, il faut se rendre dans la partie administration puis dans site pour renseigner le nom de domaine de votre instance de Pod (par défaut 'example.com').***
>
> ***Avant la mise en production, il faut vérifier le fonctionnement de la plateforme dont l'ajout d'une vidéo, son encodage et sa suppression.***
>
> ***Attention, pour ajouter une vidéo, il doit y avoir au moins un type de vidéo disponible. Si vous avez correctement peuplé votre base de données avec le fichier initial_data.json vous devez au moins avoir other/autres.***
>
> ***il faut vérifier l'authentification CAS, le moteur de recherche etc.***