

# Mise à jour / changement de version pod V2

- Avant une MAJ
  - Annoncez la maj aux utilisateurs :
  - Le jour J : (à partir de la 2.8.1)
- Commandes générales de mise à jour
- Pour la mise à jour en 2.8 :
- Si vous faites une mise à jour au delà de la version 2.2 :
  - Migration des Notes vers AdvancedNotes :
- Migration à partir de la version 2.3
- Migration 2.5.0 à 2.5.1
- Migration en 2.6.0
- Update de ElasticSearch
- Mise à jour des encodeurs déportés
- Optionnel - Mise à jour d'Opencast Studio



## Attention

Vous consultez une documentation pour une ancienne version de Pod (v2). Si vous utilisez Pod V3 rendez-vous sur [Mise à jour de version en pod V3](#).

## Avant une MAJ

### Annoncez la maj aux utilisateurs :

À partir de la version 2.8.2, vous pouvez aller dans l'administration de Pod, ([https://VOTRE\\_SERVEUR/admin/main/configuration/](https://VOTRE_SERVEUR/admin/main/configuration/)), vous y trouverez :

- Le champ "maintenance\_text\_sheduled" vous permet de définir un message de maintenance personnalisé.
- Le champ "maintenance\_sheduled" vous permet d'afficher/masquer (=1 / 0) ce message sur Pod.

### Le jour J : (à partir de la 2.8.1)

Basculez en mode maintenance (maintenance\_mode = 1), cela va désactiver certaines fonctionnalités, et afficher un bandeau "Maintenance en cours. Certaines fonctionnalités sont indisponibles".

## Commandes générales de mise à jour

Voici les commandes à lancer pour effectuer une mise à jour de Pod V2

```
pod@pod:~$ cd django_projects/podv2/
pod@pod:~/django_projects/podv2$ workon django_pod
(django_pod) pod@pod:~/django_projects/podv2$ git status
(django_pod) pod@pod:~/django_projects/podv2$ git pull origin master
(django_pod) pod@pod:~/django_projects/podv2$ pip3 install -r requirements.txt
(django_pod) pod@pod:~/django_projects/podv2$ python manage.py makemigrations
(django_pod) pod@pod:~/django_projects/podv2$ python manage.py migrate
# Attention : avant de lancer collectstatic --clear, assurez-vous d'avoir sauvegardé le dossier static/custom si
vous y avez mis des fichiers personnalisés.
(django_pod) pod@pod:~/django_projects/podv2$ python manage.py collectstatic --no-input --clear
(django_pod) pod@pod:~/django_projects/podv2$ sudo systemctl restart uwsgi-pod
```

Si vous aviez activé le mode maintenance, pensez à le désactiver (maintenance\_mode = 0), après avoir testé que tout est bien reparti 😊.

### Pour la mise à jour en 2.8 :

Pour éviter les conflits de clés étrangère lors de la migration / mise à jour, Il faut modifier votre configuration de base de données(mysql) et annuler la vérification des clés étrangères en ajoutant `foreign_key_checks = 0`:

ce qui donne : 'OPTIONS': { 'init\_command': "SET storage\_engine=INNODB, sql\_mode='STRICT\_TRANS\_TABLES', innodb\_strict\_mode=1, foreign\_key\_checks = 0", },

Lors de la migration de Pod vers une version 2.8, lors de l'exécution de la commande `python manage.py makemigrations`, il sera demandé s'il y a eu un changement de modèle en lien **Attendee** en lieu et place de **User**.

C'est le cas, la classe Attendee vient remplacer User (cf. `bbb/models.py`): il faut alors répondre **Yes** à la question posée à ce moment de la migration.

Ainsi, dans la base de données de Pod, la table `bbb_user` est renommée en `bbb_attendee`.

Enfin, si vous utilisez les groupes pour restreindre l'accès à vos vidéos dans votre instance de Pod, il faut migrer les groupes de gestion vers les nouveaux groupes d'accès (voir [Configurer les groupes d'accès](#))

Pour cela, nous avons mis à disposition une commande :

```
(django_pod) pod@pod:~/django_projects/podv2$ python manage.py accessgroups migrate_groups
```

Si vous faites une mise à jour au delà de la version 2.2 :

### Migration des Notes vers AdvancedNotes :

```
(django_pod) pod@pod:~/django_projects/podv2$ python manage.py shell
from pod.video.models import Notes, AdvancedNotes
from django.utils import timezone
if __name__ == "__main__":
    notes = Notes.objects.all()
    for n in notes:
        AdvancedNotes.objects.create(
            video=n.video, user=n.user,
            added_on=timezone.now(),
            modified_on=timezone.now(),
            note=n.note, timestamp=0,
            status='0'
        )
```

NB : les notes créées avant AdvancedNotes seront enregistrées avec un timestamp à 0 et un status « privé - » (uniquement l'auteur de la note peut la voir)

## Migration à partir de la version 2.3

Pour l'utilisation de l'autotranscription, il faut suivre la première partie de la documentation disponible à cette adresse : [Mise en place de l'autotranscription](#)

## Migration 2.5.0 à 2.5.1

A partir de la version 2.5.1 Pod utilise des commandes de la librairie deepspeech qui nécessite que les CPU supportent les instructions "AVX".

Ces instructions sont disponibles à partir des "Sandy Bridge" (intel) et "Bulldozer" (amd) en gros après 2011.

Si vous êtes dans une version antérieure à la 2.5.1, vous pouvez tester la compatibilité dans l'environnement podv2 avec la commande :

```
python -m deepspeech
```

si la réponse est :

```
No module named deepspeech.__main__;
```

```
'deepspeech' is a package and cannot be directly executed"
```

c'est bon, si la réponse est

```
"Illegal instruction (core dumped)"
```

ce n'est pas bon et il faudra soit utiliser un autre serveur soit désactiver l'autotranscription ainsi que l'importation de la librairie comme expliqué ci dessous.

## Migration en 2.6.0

lors de la commande

```
python manage.py makemigrations
```

une erreur apparait : **Instruction non permise**

**cette erreur est lié à deepspeech**

**il faut donc :**

```
pip3 uninstall deepspeech
```

```
nano pod/video/transcript.py
```

commenter la ligne **from deepspeech import Model**

puis continuer les commandes ...

## Update de Elasticsearch

En cas d'update du paquet Elasticsearch, le plugin analysis-icu ne se met pas à jour automatiquement et n'est plus compatible avec la version d'ElasticSearch. il faut le supprimer, le réinstaller, recréer l'index puis réindexer les vidéos.

```
cd /usr/share/elasticsearch/  
bin/elasticsearch-plugin remove analysis-icu  
bin/elasticsearch-plugin install analysis-icu  
service elasticsearch restart
```

```
su - pod  
cd django_projects/podv2  
python manage.py create_pod_index  
python manage.py index_videos --all
```

## Mise à jour des encodeurs déportés

```
pod@pod:~$ cd django_projects/podv2/  
pod@pod:~/django_projects/podv2$ workon django_pod  
(django_pod) pod@pod:~/django_projects/podv2$ git status  
(django_pod) pod@pod:~/django_projects/podv2$ git pull origin master  
(django_pod) pod@pod:~/django_projects/podv2$ pip3 install -r requirements.txt  
(django_pod) pod@pod:~/django_projects/podv2$ sudo /etc/init.d/celeryd restart
```

## Optionnel - Mise à jour d'Opencast Studio

Pour mettre à jour le studio d'Opencast dans votre instance de Esup-Pod, voici les étapes à suivre :

- Rendez-vous dans le dossier opencast-studio/
- Récupérer la dernière version d'Opencast Studio via la commande suivante :

```
git pull
```

- Régénérez l'Opencast Studio avec la bonne configuration pour Pod via les commandes suivantes :

```
export PUBLIC_URL=/studio  
npm install  
npm run build
```

- Le répertoire **build** est alors mis à jour. Renommez-le en **studio**, puis copier le dans le répertoire **pod/custom/static/opencast/**

```
mkdir -p pod/custom/static/opencast/studio  
cp -r build/* pod/custom/static/opencast/studio
```

- Finalement, n'oubliez pas de collecter vos fichiers statiques pour la mise en production via la commande :

```
(django_pod) [userpod@video][~/data/www/userpod/django_projects/podv2] python manage.py collectstatic
```