

Mise en place de l'obsolescence des vidéos dans pod

Nous avons ajouté une date de suppression pour chaque vidéo. Ce champ date est créé avec deux ans de plus que la date d'ajout par défaut. Ces deux ans sont paramétrables via le settings `DEFAULT_YEAR_DATE_DELETE`

1/ Attribut `date_delete`

Lors de l'ajout de la vidéo, quand l'upload est terminé est que la vidéo est enregistrée, cette date est modifiée si l'affiliation du propriétaire est précisé dans la variable `ACCOMMODATION_YEARS`.

Par exemple à Lille, nous avons `ACCOMMODATION_YEARS={'faculty': 3, 'employee': 3, 'staff': 3}`. Donc pour toute vidéo déposée sur Pod, pour tout le monde c'est 2 ans mais pour les enseignants, personnels et équipe, c'est 3 ans.

En gros, si vous avez `ACCOMMODATION_YEARS = {'student':1}` dans votre fichier de settings, si un étudiant poste une vidéo dans Pod alors sa date de suppression sera égale à un an de plus que sa date d'ajout, pour tous les autres, ca reste deux ans. (Attention, il ne faut pas avoir deux fois la variable dans votre fichier de settings ; combinez les 2 précédents ainsi : `ACCOMMODATION_YEARS={'faculty': 3, 'employee': 3, 'staff': 3, 'student':1}`)

Pour rappel et par défaut, voici les valeurs possible pour l'affiliation :

```
AFFILIATION = getattr(
    settings, 'AFFILIATION',
    (
        ('student', _('student')),
        ('faculty', _('faculty')),
        ('staff', _('staff')),
        ('employee', _('employee')),
        ('member', _('member')),
        ('affiliate', _('affiliate')),
        ('alum', _('alum')),
        ('library-walk-in', _('library-walk-in')),
        ('researcher', _('researcher')),
        ('retired', _('retired')),
        ('emeritus', _('emeritus')),
        ('teacher', _('teacher')),
        ('registered-reader', _('registered-reader'))
    )
)
```

Donc si vous mettez à jour votre Pod et que vous ne touchez à rien, toutes vos vidéos auront une date de suppression égale à deux ans suivant votre date de mise à jour de votre plateforme.

2/ Gestion de l'obsolescence et de la notification

Nous avons ajouté une variable `WARN_DEADLINES = getattr(settings, "WARN_DEADLINES", [])`. Elle est donc vide par défaut.

cette variable doit contenir le nombre de jour avant la date de suppression pour lesquels le propriétaire doit être prévenu.

par exemple, si vous mettez `WARN_DEADLINES = [60, 30, 7]`, les propriétaires de vidéos recevront un mail 60 jours avant la date de suppression, 30 jours avant et 7 jours avant.

Ensuite,

- s'ils sont "staff", le courriel envoyé leur précisera que leur vidéo va être bientôt supprimée mais qu'ils peuvent modifier la date dans l'interface d'édition avec un lien pour les y conduire.
- s'ils sont "non staff" (les étudiants), le mail les invitera à contacter les managers de la plateforme (`CONTACT_US_EMAIL` ou `MANAGER` de l'établissement si `USE_ESTABLISHMENT_FIELD` à True)

Les gestionnaires recevront en récapitulatif la liste des vidéos bientôt supprimées.

Pour les vidéos dont la date de suppression est dépassée, on a ajouté une variable `POD_ARCHIVE_AFFILIATION`. Cette variable est un tableau qui contient toutes les affiliations pour lesquelles on souhaite archiver la vidéo plutôt que de la supprimer. À Lille, `POD_ARCHIVE_AFFILIATION` contient les valeurs suivantes : `['faculty', 'staff', 'employee', 'affiliate', 'alum', 'library-walk-in', 'researcher', 'retired', 'emeritus', 'teacher', 'registered-reader']`

Archivage

Si l'affiliation du propriétaire est dans cette variable `POD_ARCHIVE_AFFILIATION`, alors :

- Les vidéos sont affectées à un utilisateur précis que l'on peut spécifier via le paramètre `ARCHIVE_OWNER_USERNAME`.
- Elles sont mise en mode brouillon (visible seulement par un superadmin + utilisateur "archive")
- Le mot `_("archived")` est ajouté à leur titre.

- Enfin, elles sont également ajoutées à l'ensemble "Vidéo à supprimer" (accessible via l'interface d'admin)



Attention

Si avant d'être archivée, une vidéo a été partagée via un lien contenant son hash code (quelque-chose comme '833e349770[...]4b5fdded763', accessible lorsqu'on partage une vidéo en mode brouillon), alors elle continue à être visibles aux personnes disposant de ce lien.

Sinon, les vidéos sont tout simplement directement supprimées.

Les gestionnaires recevront deux autres mails chaque jour. Un avec la liste des vidéos archivées et un autre mail avec la liste des vidéos supprimées (identifiant et titre).

De plus, deux fichiers CSV (deleted.csv et archived.csv) sont créés dans le répertoire log de Django et renseignés avec la liste des vidéos archivées ou supprimées.

3/ Mise en route du traitement automatique

Pour lancer le traitement quotidien des vidéos, il faut au préalable ajouter cette variable dans votre fichier de configuration `USE_OBSOLESCENCE = True`

Ensuite, il faut lancer une tâche cron qui va passer une fois par jour (ici à 5:00) avec la commande suivante :

```
0 5 * * * cd /home/pod/django_projects/podv2 && /home/pod/.virtualenvs/django_pod/bin/python manage.py
check_obsolete_videos
```

Annexes

Désarchiver une vidéo

Il peut arriver qu'une vidéo ait été archivée par erreur, que la date d'obsolescence était mal réglée, etc... Si la vidéo est archivée (et non supprimée), et si vous y prenez à temps, la vidéo peut encore être restaurée.

Pour cela, il vous faut préciser l'identifiant de la vidéo, et indiquer l'utilisateur à qui réattribuer la vidéo : (1er paramètre=video_id, 2e paramètre=user_id)

```
pod@pod:~$ python manage.py unarchive_video 1234 5678
```