

Mise en place du studio Opencast

- [Contexte](#)
- [Présentation du fonctionnement d'Opencast Studio](#)
- [Installation](#)
- [Configuration et pré-requis](#)
 - [Configuration](#)
 - [Pré-requis](#)
- [Exploitation](#)

Contexte

Depuis la version 2.8 de Pod, une solution permettant aux usagers de retrouver directement dans Pod un enregistreur écran + micro avait été mis en place. Grâce à cette solution, les usagers de Pod peuvent réaliser en autonomie des capsules vidéos, qui pouvaient alors être envoyées automatiquement dans Pod.

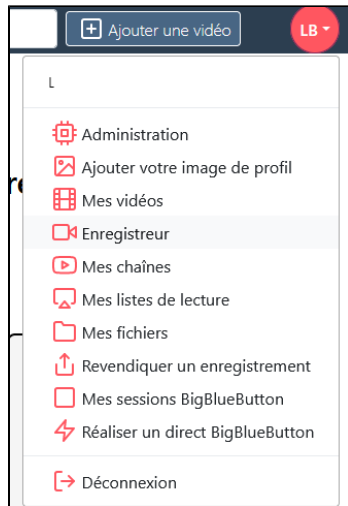
Ce système n'étant pas optimal, il a été remplacé dans la version 2.9 de Pod par l'**Opencast Studio**.

Présentation du fonctionnement d'Opencast Studio

Ce système d'Opencast studio est bien plus performant que l'ancien enregistreur, et permet aux usagers de Pod, en totale autonomie, d'enregistrer la caméra, le micro et l'écran de l'utilisateur.

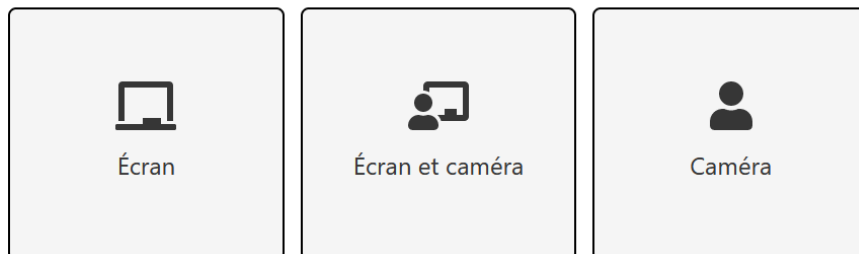
Ainsi, en plus de pouvoir enregistrer tout cela localement, l'utilisateur peut publier automatiquement l'ensemble sur la plateforme Pod avec différentes possibilités de rendu. (Voir présentation faite lors de la journée Esup).

- L'icône d'Enregistreur apparaît dans le menu de profil des usagers **connectés** à Pod



- L'utilisateur peut choisir les sources qu'il souhaite enregistrer

Quelle(s) source(s) vidéo(s) enregistrer?



- L'utilisateur peut paramétrer ses sources audio et vidéo (en terme de choix de matériel, de qualité)

Écran sélectionné

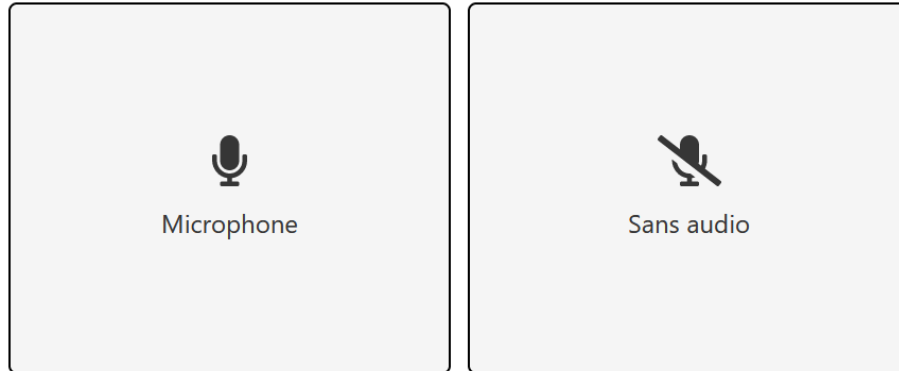


◀ Répétez la sélection de la (des) source(s)

Suivant ▶

- L'utilisateur peut également décider d'enregistrer l'audio ou non

Enregistrer l'audio ?



Retour

- Un bouton rouge permet l'enregistrement.



- Une fois l'enregistrement réalisé, il peut également couper le début et/ou la fin de celui-ci.

Vous êtes satisfait de votre enregistrement?



Début: 00:02,1
00:10,9 / 00:22,9

◀ Ignorer et réenregistrer

Suivant ▶

- A la fin de son enregistrement, et après les coupes - si besoin - , l'utilisateur peut choisir d'enregistrer en local ou d'envoyer directement sur Pod. En cas d'enregistrement de 2 flux vidéos, Pod les concatènera en une seule vidéo, selon l'option PIP choisie. La vidéo finale se retrouvera, comme d'habitude, dans le menu **Mes vidéos**.

Presque terminé

/// Téléverser vers Pod

TITRE

Présentation OpenCast Studio & POD

PLEASE SELECT RENDER YOU WANT

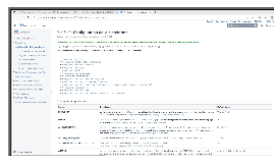
50/50 - Same size for screen and camera

50/50 - Same size for screen and camera

Pip - Camera inside the screen video to top right

Pip - Camera inside the screen video to bottom right

/// Télécharger l'enregistrement



⬇ Télécharger (923 KB)



⬇ Télécharger (1,8 MB)

◀ Retour

Installation

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

- Récupérer les sources Opencast Studio via la commande suivante :

```
git clone https://github.com/elan-ev/opencast-studio/
```

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

Installation d'Opencast Studio v 1.0 pour les versions d'**Esup-Pod <= 3.4.1**

```
cd opencast-studio/  
# cf la liste des tags: https://github.com/elan-ev/  
/opencast-studio/tags  
git checkout tags/2023-09-14  
export PUBLIC_URL=/studio  
npm install  
npm run build
```

Installation d'Opencast Studio v 2.0 pour les versions d'**Esup-Pod >= 3.5.0**

```
cd opencast-studio/  
export PUBLIC_PATH=/studio  
npm install  
npm run build:release
```

- Un répertoire **build** est alors généré. Renommez ce répertoire **build** 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
```



La génération d'Opencast Studio peut très bien se réaliser *sur un autre serveur*. Pod n'a besoin que du répertoire *build*, renommé en **studio**, et positionné dans le sous-répertoire **pod/custom/static/opencast**



Il est indispensable que le serveur soit configuré en **HTTPS** (et non en HTTP) pour que l'Opencast Studio puisse fonctionner.

Configuration et pré-requis

Configuration

Dans votre custom/settings-local.py, les paramètres suivants concernent l'utilisation d'Opencast Studio:

Propriété	Description	Valeur par défaut
USE_OPENCAST_STUDIO	Activer l'enregistreur Opencast Studio	False
OPENCAST_FILES_DIR	Répertoire de travail pour les fichiers générés par Opencast Studio (sera accessible dans le media)	"opencast-files"
ENCODE_STUDIO	Fonction appelée pour lancer l'encodage des vidéos intermédiaires du studio	"start_encode_studio"


```
(django_pod) [userpod@videol][ /data/www/userpod/django_projects/podv2] echo "from pod.recorder.models import Recorder; from pod.video.models import Type; type=Type.objects.get(pk=1); rec=Recorder.objects.create (name='Studio', address_ip='127.0.0.1', recording_type='studio', type=type)" | python manage.py shell
```

Exploitation

Les fichiers d'Opencast Studio sont générés dans le répertoire **/media/opencast-files/** et tout est loggué dans l'interface d'administration de Pod.

En effet, il est possible de retrouver les enregistrements réalisés par les usagers via le module Recorder / Enregistrement pour l'enregistreur de type Studio défini :

Administration de Pod

Accueil > Recorder > Enregistrements > test1

Modification de Enregistrement

Enregistreur :

Studio - 127.0.0.1

Enregistreur qui a fait cet enregistrement.

User :

L

Utilisateur qui a fait l'enregistrement

Titre :

test1

Type d'enregistrement :

Studio

Source file :

Commentaire :

```
*** generate_intermediate_video (presenter/source) /data/www/podtest/media/opencast-
files/1411c80a-4e88-47de-a541-5351f32a8232/presenter_source.webm ***- /opt/ffmpeg/ffmpeg
/ffmpeg -hide_banner -y -vsync 0 -i "/data/www/podtest/media/opencast-files/1411c80a-
4e88-47de-a541-5351f32a8232/presenter_source.webm" -ss 0.64 -to 4.84 -c:a aac -ar 48000 -c:v
h264 -profile:v high -pix_fmt yuv420p -crf 22 -sc_threshold 0 -force_key_frames
"expr:gte(t,n_forced*1)" -max_muxing_queue_size 4000 -deinterlace -threads 0 -vsync 0 -movflags
+faststart -f mp4 /data/www/podtest/media/opencast-files/1411c80a-4e88-47de-
a541-5351f32a8232/1411c80a-4e88-47de-a541-5351f32a8232.mp4

- Encoding Mp4: Mon Feb 21 09:42:38 2022
```

Supprimer