

# Mise en place de la publication des présentations Web de BigBlueButton vers Pod v2



Cette documentation ne concerne que la version 2.X de Pod et non la version 3. X de Pod.

Le module utilisé sera supprimé de Pod dans une future version de Pod.

D'autres fonctionnalités de Pod peuvent correspondre à vos besoins :

- <https://www.esup-portail.org/wiki/x/C4CFUQ> : migration d'infrastructure BigBlueButton, avec l'appui de Pod
- <https://www.esup-portail.org/wiki/x/BQCnSw> : importer une vidéo externe / un enregistrement BigBlueButton au format vidéo

- Contexte et solution apportée
  - Contexte
  - Solution
- Architecture de la solution
  - Choix du plugin permettant la conversion d'une présentation Web BigBlueButton en vidéo
  - Installation et configuration
    - Pré-requis
    - Installation de bbb-recorder sur les serveurs d'encodage
      - Installation de bbb-recorder sur CentOS7
      - Installation effective
  - Paramétrage
    - Configuration dans Pod
    - Mise en place du job CRON
      - Étape 1 : récupération des sessions BBB et des utilisateurs BBB
      - Étape 2 : recherche des enregistrements dans BBB
      - Étape 3 : recherche des correspondances entre les utilisateurs BBB et ceux de Pod
      - Étape 4 : publication des vidéos
      - Étape 5 : suppression des sessions BBB
    - Synthèse du processus global
  - Exploitation
    - L'interface d'administration
      - Les sessions BigBlueButton
      - Nouveauté de la version 2.8 de Pod
      - Les utilisateurs BigBlueButton



## Gestion de l'erreur Evaluation failed (Octobre 2021)

En cas d'erreur du type :

```
Error: Evaluation failed: TypeError: Cannot read properties of null (reading 'duration')
at __puppeteer_evaluation_script__:2:72
at ExecutionContext.evaluateInternal (/home/pod/bbb-recorder/node_modules/puppeteer/lib/ExecutionContext.js:122:13)
at process._tickCallback (internal/process/next_tick.js:68:7)
-- ASYNC --
at ExecutionContext.<anonymous> (/home/pod/bbb-recorder/node_modules/puppeteer/lib/helper.js:111:15)
at DOMWorld.evaluate (/home/pod/bbb-recorder/node_modules/puppeteer/lib/DOMWorld.js:112:20)
at process._tickCallback (internal/process/next_tick.js:68:7)
-- ASYNC --
at Frame.<anonymous> (/home/pod/bbb-recorder/node_modules/puppeteer/lib/helper.js:111:15)
at Page.evaluate (/home/pod/bbb-recorder/node_modules/puppeteer/lib/Page.js:860:43)
at Page.<anonymous> (/home/pod/bbb-recorder/node_modules/puppeteer/lib/helper.js:112:23)
at main (/home/pod/bbb-recorder/export.js:136:38)
at process._tickCallback (internal/process/next_tick.js:68:7)
```

A priori, en ajoutant une temporisation dans le script export.js de bbb-recorder, le problème ne se présente plus. En attendant une mise à jour de bbb-recorder, il peut-être nécessaire de modifier `bbb-recorder/export.js` et y ajouter à la ligne 125 :

```
// Waiting for page loading
await page.waitFor(10000)
```

D'ailleurs, un commit vient d'être fait dans bbb-recorder pour résoudre ce problème : cf. <https://github.com/tdebatty/bbb-recorder/commit/46e7511f035904344a06fc9c346d3f9b8a34ccdf>

### **i** Mise à jour en lien avec les présentations Web au format BigBlueButton 2.3 (Juillet 2021) => Résolu en Septembre 2021

Suite à la mise à jour de BigBlueButton ou de Scalelite, il peut arriver que les liens des enregistrements BigBlueButton changent de format. Ainsi, avant la mise à jour, les liens étaient sous la forme : [https://bbb.univ.fr/playback/presentation/2.0/playback.html?meetingId=INTERNAL\\_MEETING\\_ID](https://bbb.univ.fr/playback/presentation/2.0/playback.html?meetingId=INTERNAL_MEETING_ID)  
Après la mise à jour, les liens peuvent se retrouver sous la forme : [https://bbb.univ.fr/playback/presentation/2.3/INTERNAL\\_MEETING\\_ID/?meetingId=INTERNAL\\_MEETING\\_ID](https://bbb.univ.fr/playback/presentation/2.3/INTERNAL_MEETING_ID/?meetingId=INTERNAL_MEETING_ID)

Pour gérer ce changement de format vis-à-vis de ce système de publication vers Pod, il est nécessaire de réaliser les modifications suivantes :

- Remplacer, sur les serveurs d'encodage, le fichier `/home/%USERPOD%/bbb-recorder/export.js` par le fichier `export.js` fourni dans ce projet Github : <https://github.com/amirhoseinsalimi/bbb-recorder/tree/bbb-v23>  
⚠ En effet, il semblerait que le projet bbb-recorder (<https://github.com/jibon57/bbb-recorder>) n'ait pas encore été mis à jour pour prendre en compte ce format 2.3 des enregistrements.
- Mettre à jour la version de Pod en v2.8.2 ou supérieure.
- Ajouter à votre `settings_local.py` : `BBB_VERSION_IS_23 = True`

### **i** Astuce en lien avec Scalelite (Juillet 2021) => Résolu en Octobre 2021 (Scalelite v 2.0.1.2)

Cela n'a rien à voir avec le système de publication pour Pod, mais si vous avez un Scalelite et que, parfois, certains enregistrements n'apparaissent pas aux utilisateurs, cela peut provenir de Scalelite.

Après recherche, dans mon cas, il s'avérait que certains enregistrements restaient bloqués, au format `.tar`, dans le répertoire **pool de Scalelite** (`/mnt/scalelite-recordings/var/bigbluebutton/spool/`) et n'étaient alors jamais publiés.

Au final, en mettant à jour la base de données de Scalelite et en le redémarrant, tout est rentré dans l'ordre. Voici les commandes utilisées :

```
sudo docker exec -t scalelite-api bundle exec rake db:migrate  
sudo systemctl restart scalelite.target
```

## Contexte et solution apportée

### Contexte

Suite à la pandémie de COVID-19, il a été mis en place, à l'université de Montpellier, un système de classe virtuelle Open Source reposant sur BigBlueButton (BBB).

Pour informations, BigBlueButton (<https://bigbluebutton.org/>) est un outil de classe virtuelle ayant les fonctionnalités suivantes :

- Vidéo/webcam
- Audio
- Chat
- Partage de document + annotation
- Partage d'écran
- Sondage
- Enregistrement
- Création de groupes
- Prises de notes partagées
- Intégration de vidéos externes
- Intégration Moodle et WordPress

Il faut savoir que l'enregistrement d'une session BBB se fait sous la forme d'une présentation Web. Cette présentation Web peut alors être lue directement depuis un navigateur, en ayant récupéré l'adresse dans le client BBB (typiquement Moodle ou Greenlight).

Bien que cette façon de faire est des avantages - en particulier concernant l'espace de stockage (une présentation Web n'étant pas une vidéo, sa taille est très petite) - elle présente quelques inconvénients, à savoir :

- cette présentation Web n'est pas une vidéo, elle ne peut être publiée sur des plateformes telles Pod ou Youtube,
- cette présentation Web n'est pas une vidéo, elle ne peut être coupée,
- cette présentation Web ne peut être consultée qu'au travers des clients BBB (Moodle ou Greenlight), ou donnée via un lien dans un mail. Si l'on souhaite donner d'autres accès, cela n'est pas possible,
- cette présentation Web ne peut être lue que sur les navigateurs Firefox et Chrome dans une version à jour.

Au final, certains enseignants souhaitaient convertir ces présentations Web sous la forme de vidéos, et les publier directement dans la plateforme vidéo de l'université, qui repose sur Pod v2.

# Solution

Pour résoudre cette problématique et répondre favorablement aux demandes des enseignants de l'université, j'ai conçu la solution suivante, qui se repose totalement sur Pod v2.

Cette solution fonctionne pour BigBlueButton mais également pour Scalelite (cf. <https://github.com/blindsidenetworks/scalelite>), un système de répartition de charge pour BigBlueButton.

## **i** Pourquoi cette solution ?

Ne souhaitant pas que l'ensemble des présentations Web BigBlueButton soient converties automatiquement en vidéo, surtout qu'elles n'ont pas toutes vocation à l'être, il fallait trouver une solution permettant que cela soit l'utilisateur qui choisisse s'il souhaite, ou non, convertir ses présentations Web.

De plus, cette solution permet également de n'encoder une présentation Web BigBlueButton qu'une seule fois au maximum; dans le cas où plusieurs modérateurs ont participé à la session BBB, n'importe lequel pourra réaliser cette conversion. Les autres ne pourront plus convertir cette présentation Web, mais ils auront accès à l'information de qui a réalisé cet encodage. Ils pourront alors contacter directement cet utilisateur pour lui demander de partager la vidéo avec eux, voire de les mettre en tant que propriétaire additionnelle de la vidéo.

L'idée de cette solution est de :

- Récupérer régulièrement les informations concernant les sessions en cours, ainsi que les usagers, dans BigBlueButton,
- Positionner ces informations dans la base de données de Pod,
- Traiter ces données pour permettre aux utilisateurs, connectés dans Pod, de pouvoir consulter leurs enregistrements des sessions BigBlueButton qu'ils ont réalisés,

Par exemple, voici la liste des sessions BigBlueButton réalisée à des fins de test.

The screenshot shows the Pod interface with the following elements:

- Navigation Bar:** 'VIDEO' logo, 'Chaînes Utilisateurs Types Directs', search bar, 'Ajouter une vidéo' button, 'LB' dropdown, and 'fr' language selector.
- Breadcrumbs:** 'Accueil / Mes sessions BigBlueButton'.
- Section Header:** '11 enregistrements trouvés'.
- Text:** 'Voici la liste des sessions BigBlueButton enregistrées pour lesquelles vous étiez modérateur. Ce module vous permet de créer une vidéo à partir d'une présentation BigBlueButton. Peu de temps après la publication de la présentation, la vidéo correspondante apparaîtra dans vos vidéos. Remarque: une présentation BigBlueButton peut être encodée par un autre modérateur. Si un tel cas se produit, ces informations seront affichées directement dans la liste. Vous pouvez alors contacter directement cet utilisateur pour lui demander de partager la vidéo avec vous, voire de vous mettre en tant que propriétaire supplémentaire de la vidéo.'
- Session Cards:**
  - 1. 'Dernier test LB (6 octobre 2020 14:29)' - Button: 'Publier cette présentation'
  - 2. 'Home Room (6 octobre 2020 13:34)' - Button: 'Déjà publié'
  - 3. 'Séance introductive aux projets tutorés 2020/2021 (14 septembre 2020 10:44)' - Button: 'Déjà publié'
  - 4. 'TEST BBB -> Pod (7 septembre 2020 11:09)' - Button: 'Déjà publié'
  - 5. 'Home Room (4 septembre 2020 14:55)' - Button: 'Déjà publié'
  - 6. 'Home Room (4 septembre 2020 14:25)' - Button: 'Déjà publié'
  - 7. 'Questions/Réponses durant les cours (4 septembre 2020 07:54)' - Button: 'Déjà publié'
  - 8. 'Home Room (3 septembre 2020 16:42)' - Button: 'Déjà publié'

- Permettre à ces utilisateurs, connectés dans Pod, de créer des vidéos à partir de ses présentations BigBlueButton.

Voici l'écran permettant la confirmation de la création d'une vidéo à partir d'une présentation BigBlueButton.

**VIDEO** Chaînes Utilisateurs Types Directs Rechercher Ajouter une vidéo LB fr

Accueil / Mes sessions BigBlueButton / Publier la présentation BigBlueButton

## Voulez-vous vraiment publier cette présentation BigBlueButton ?

Une vidéo sera créée à partir de la présentation BigBlueButton et sera disponible sur cette plateforme.  
Attention: ce traitement peut durer longtemps. Vous recevrez un e-mail lorsque ce traitement sera terminé.

[Publier cette présentation](#)

UNIVERSITÉ DE MONTPELLIER

> Contactez nous > Legal notice > Top of page  
> Help > Université de Montpellier

- Une fois la vidéo créée à partir de la présentation Web BigBlueButton, celle-ci apparaît dans la liste habituelle de mes vidéos.

Voici le rendu d'une vidéo créée à partir d'une présentation Web BigBlueButton.

**VIDEO** Chaînes Utilisateurs Types Directs Rechercher Ajouter une vidéo LB fr

Accueil / Vidéos / Dernier test LB

The screenshot shows a video player interface. The main content area is split into two parts. On the left, a presentation slide is visible with the following text: "septembre 19, 2019", "PS", "The following notes provide guidance on setting up the Extron Media Processor and Encoders for publishing live streams to third party services such as Facebook Live via RTMP. For more information, please call your Extron Application Engineer.", "SPECIAL NOTES", "Streaming to 3rd party services requires a user to have an account with those services. Accounts with third party providers are the responsibility of those maintaining the stream and its content.", and "UNIVERSITY OF MONTPELLIER Page 1". On the right, a video frame shows a hand making a peace sign. Below the video frame, there is a chat window with the following messages: "loic:TEST1", "loic:Test2", "loic:Test3", and "loic:Test4". The video player controls at the bottom show a progress bar at 0:47 and a play button. The video title is "Dernier test LB [6 octobre 2020]" with a warning icon.

Dernier test LB [6 octobre 2020]

Une vidéo créée à partir d'une présentation Web BigBlueButton est réellement identique à cette présentation Web, ce qui signifie qu'elle contient :

- le document de présentation,
- l'audio,
- la vidéo/webcam,
- le partage d'écran,
- le chat public,
- le tableau blanc.

### **Aucun impact sur BigBlueButton**

Cette solution repose totalement sur Pod et n'impacte en rien BigBlueButton. Aucune modification n'est à réaliser côté BigBlueButton.

### **Les utilisateurs dans BigBlueButton**

Les informations concernant les modérateurs dans BigBlueButton dépendent du client BBB utilisé : **Greenlight** ou le plugin **mod\_bigbluebuttonn pour Moodle**.

Le système réalisé n'a été testé qu'avec le plugin **mod\_bigbluebuttonn pour Moodle** (CASifié, donc les données proviennent de notre annuaire LDAP); cela signifie que les modérateurs sont définis - dans mon cas - sous la forme "Prenom Nom".

Il est possible de paramétrer ce format, via le paramètre **BBB\_USERNAME\_FORMAT** (à partir de la version 2.7.2). A l'heure actuelle, les formats "Prenom Nom" ou "Nom Prenom" peuvent être gérés via ce paramétrage.

 **Ce point est crucial pour que le système fonctionne correctement : une correspondance doit exister sur le "Prenom Nom" ou "Nom Prenom" (selon la configuration BBB\_USERNAME\_FORMAT) des modérateurs de BigBlueButton et le "Prenom Nom" des utilisateurs dans la base de données de Pod.**

 A priori, cela devrait pouvoir fonctionner avec **Greenlight** si celui-ci est configuré pour utiliser l'**annuaire LDAP** de l'établissement et les champs **givenName** et **sn**.

## Architecture de la solution

### Choix du plugin permettant la conversion d'une présentation Web BigBlueButton en vidéo

Pour convertir une présentation Web BigBlueButton en vidéo, certains plugins existent déjà sur Github; il ne paraît pas raisonnable de redévelopper un tel système.

Pour arriver à faire mon choix, j'ai testé les 3 solutions suivantes :

- **bbb-recorder** (<https://github.com/jibon57/bbb-recorder>) : un plugin, indépendant de BigBlueButton, qui permet de convertir - via un script - une présentation Web BigBlueButton en fichier vidéo. Ce plugin permet également une diffusion en direct (flux RTMP) d'un cours BigBlueButton.
- **bbb-download** (<https://github.com/createwebinar/bbb-download>) : un plugin, totalement couplé à BigBlueButton, qui convertit automatiquement les présentations Web de BigBlueButton en fichier vidéo.
- **BigBlueButton-liveStreaming** (<https://github.com/aau-zid/BigBlueButton-liveStreaming>) : un plugin, indépendant de BigBlueButton, permettant de publier en live (via RTMP) une session BigBlueButton et de l'enregistrer.

### **Plugin non testé**

A l'époque de mes tests, le plugin **bbb-video-download** (<https://github.com/tilmanmoser/bbb-video-download>) n'existait pas encore. Il pourrait être intéressant de le tester par la suite, s'il s'avère plus prometteur que **bbb-recorder**.

Au final, voici ce qui est ressorti de mes tests.

	<b>bbb-recorder</b>	<b>bbb-download</b>	<b>BigBlueButton-liveStreaming</b>
<b>Enregistre un cours BBB en vidéo</b>	 <i>format webm, mp4</i>		 <i>peut enregistré le cours qu'il publie en live au format mkv</i>
<b>Exporte un cours BBB en direct live</b>			
<b>Enregistre un cours en temps réel</b>			
<b>Indépendance vis-à-vis de BigBlueButton</b>	 <i>peut-être installé sur n'importe quel serveur</i>	 <i>doit être installé sur tous les serveurs BBB</i>	 <i>peut-être installé sur n'importe quel serveur</i>

<b>Contenu de la vidéo finale</b>	<ul style="list-style-type: none"> <li>✔ présentation</li> <li>✔ audio</li> <li>✔ vidéo</li> <li>✔ partage d'écran</li> <li>✔ chat</li> <li>✔ whiteboard</li> </ul>	<ul style="list-style-type: none"> <li>✔ présentation</li> <li>✔ audio</li> <li>✘ vidéo</li> <li>✔ partage d'écran</li> <li>✘ chat</li> <li>✘ whiteboard</li> </ul>	<ul style="list-style-type: none"> <li>✔ présentation</li> <li>✔ audio</li> <li>✔ vidéo</li> <li>✔ partage d'écran</li> <li>✔ chat</li> <li>✔ whiteboard</li> </ul>
<b>Technologies</b>	NodeJS, xvfb, Chrome, ffmpeg, shell	Python, ruby, ffmpeg, shell	Docker, python, xvfb, ffmpeg, shell
<b>Mise à jour régulière</b>	✔	✘ Dernière mise à jour en 2018	✔
<b>Notion de charge</b>	💡 Le fichier vidéo fait entre 2 et 4Mo par minute, en webm ou mp4	💡 Encode nécessairement toutes les Web conférences en vidéo	<ul style="list-style-type: none"> <li>📘 1 stream nécessite entre 4 et 6 vCPU + 4 Go RAM</li> <li>✔ possibilité de faire plusieurs streams sur une même VM</li> <li>💡 le fichier vidéo fait ~19Mo par minute, en mkv</li> </ul>
<b>Commentaires</b>	Facilement modifiable (scripts JS pour NodeJS)	Complètement intégré à BBB (une fois un cours enregistré terminé, une vidéo - en plus de la présentation - est générée)	Il faut bien respecter l'ordre de démarrage (session BBB avant liveStreaming).
<b>Remarques sur mes tests</b>	Peut également publier en live le cours (publication RTMP).		

Aux vues des besoins, j'ai alors choisi **bbb-recorder** comme solution pour la conversion des présentations Web BigBlueButton en fichier vidéo.

## Installation et configuration

### Pré-requis

Techniquement, la solution repose sur :

- Le développement d'un **plugin BBB** pour Pod v2,
- L'utilisation du plugin spécifique **bbb-recorder** (<https://github.com/jibon57/bbb-recorder>).

Le fait d'exécuter le script bbb-recorder réalise les étapes suivantes :

- Lance un navigateur Chrome en arrière-plan,
- Chrome visite le lien - correspondant à la présentation Web BigBlueButton - fourni,
- Il effectue l'enregistrement d'écran sous la forme d'un fichier vidéo.

### Installation de bbb-recorder sur les serveurs d'encodage

Il est nécessaire d'installer bbb-recorder sur les serveurs d'encodage.

La documentation de référence est accessible ici : <https://github.com/jibon57/bbb-recorder>

Pour ma part, sur les serveurs CentOS 7, voici ce qui a été réalisé.

#### Installation de bbb-recorder sur CentOS7

📘 *Ce plugin n'a pas besoin d'être installé sur un serveur BigBlueButton.*

Installation de Chrome et des pré-requis (sous root)

##### Installation réalisée sur un serveur d'encodage, compte root

```
# Install xvfb
podtest@ts-sun:~/$ yum install xorg-x11-server-Xvfb
podtest@ts-sun:~/$ wget https://dl.google.com/linux/direct/google-chrome-stable_current_x86_64.rpm
podtest@ts-sun:~/$ yum localinstall google-chrome-stable_current_x86_64.rpm
```

📘 *Étant un serveur d'encodage, je considère que ffmpeg est déjà installé. Si besoin, il est nécessaire d'installer ffmpeg.*

#### Installation effective

Voici l'installation pour un utilisateur **%userpod%** (pensez à remplacer **%userpod%** par votre utilisateur).

### Installation réalisée sur un serveur d'encodage, compte %userpod%

```
%userpod%@ts-sun:~/ $ cd ~
%userpod%@ts-sun:~/ $ git clone https://github.com/jibon57/bbb-recorder
%userpod%@ts-sun:~/ $ cd bbb-recorder
%userpod%@ts-sun:~/ bbb-recorder/ $ npm install --ignore-scripts
%userpod%@ts-sun:~/ bbb-recorder/ $ cp .env.example .env
```

Gestion du répertoire contenant les vidéos : dans mon cas `/data/www/%userpod%/bbb-recorder` et du répertoire de logs `/data/www/%userpod%/bbb-recorder/logs`.

### Création des répertoires

```
%userpod%@ts-sun:~/ bbb-recorder/ $ mkdir /data/www/%userpod%/bbb-recorder
%userpod%@ts-sun:~/ bbb-recorder/ $ mkdir /data/www/%userpod%/bbb-recorder/logs
```



Si bbb-recorder n'a pas été installé avec le bon utilisateur (%userpod%), les fichiers vidéos générés ne seront sûrement pas accessibles par l'utilisateur Pod (%userpod) et ne pourront alors être encodés par les serveurs d'encodage.

*Dans les faits, cela se traduit par un 1<sup>o</sup> encodage réussi : la présentation Web de BBB sera convertie en fichier vidéo, mais ce fichier vidéo ne sera pas accessible à Pod et ne pourra être converti en vidéo Pod.*

## Paramétrage

- Édition du fichier de configuration `~/bbb-recorder/.env` pour paramétrer le RTMP (*inutile ici*) et surtout le répertoire des vidéos.

### Edition de `/home/sun/bbb-recorder/config.json`

```
{
  "rtmpUrl": "rtmp://xxxxxxxx:xxxxxxxxxxx.umontpellier.fr:1935/live/stream",
  "ffmpegServer": "ws://localhost",
  "ffmpegServerPort": 4000,
  "auth": "xxxx",
  "copyToPath": "/data/www/%userpod%/bbb-recorder"
}
```

- Si besoin, réaliser le paramétrage dans le fichier `examples/index.js` (pour réaliser un live ou enregistrer en direct une Web conférence) :

```
const BBBUrl = "https://xxxx.umontpellier.fr/bigbluebutton/",
BBBSalt = "xxxxxxxxxxxxxxxxxxxxxxxx",
joinName = "recorder";
```

- Si vous le souhaitez, vous pouvez configurer le bitrate pour contrôler la qualité de la vidéo exportée en ajustant la propriété `videoBitsPerSecond` dans `background.js`.



### Répertoire Downloads

Il faut bien penser que bbb-recorder utilise un répertoire temporaire pour générer une vidéo, avant que celle-ci ne soit copiée dans le répertoire configurée (cf. `copyToPath`). Ce répertoire temporaire correspond à `../Downloads`.

Ainsi, dans le cas d'une installation dans le home directory de l'utilisateur %userpod%, le répertoire temporaire créé et utilisé par bbb-recorder est `/home/%userpod%/Downloads`.

Il est nécessaire qu'un espace de stockage suffisant soit alors prévu.

## Configuration dans Pod

Une fois bbb-recorder installé sur les différents serveurs d'encodage, il reste à configurer le plugin bbb directement dans Pod, via l'édition de fichier **custom/settings\_local.py** (sur les encodeurs et sur le frontal) :

#### Configuration dans custom/settings\_local.py

```
##
# BigBlueButton settings
#
# Use of BigBlueButton
USE_BBB = True
# Directory of bbb-recorder plugin (see documentation https://github.com/jibon57/bbb-recorder)
# bbb-recorder must be installed in this directory, on all encoding servers
# bbb-recorder create a directory 'homedir'/Downloads that needs disk space
DEFAULT_BBB_PLUGIN = '/home/%userpod%/bbb-recorder/'
# Directory that will contain the video files generated by bbb-recorder
DEFAULT_BBB_PATH = '/data/www/%userpod%/bbb-recorder/'
# BigBlueButton or Scalelite server URL, where BBB Web presentation and API are
BBB_SERVER_URL = 'https://bbb.univ.fr/'
# BigBlueButton key or Scalelite LOADBALANCER_SECRET
BBB_SECRET_KEY = 'xxxxxxxxxxxxxxxxxxxxxxxxxxxx'
# Username format in BBB
BBB_USERNAME_FORMAT = 'first_name last_name'
# Type of the generated video by default
DEFAULT_BBB_TYPE_ID = 1
# Number of days before removal the meetings (and associated users) not already published
# To not remove old meetings, set 0 value
BBB_NUMBER_DAYS_BEFORE_DELETE = 0
```

Les éléments de paramétrage sont les suivants :

- USE\_BBB : utilisation (True/False) du plugin BBB pour Pod
- DEFAULT\_BBB\_PLUGIN : répertoire d'installation de bbb-recorder sur les serveurs d'encodage (cf. paragraphe précédent)
- DEFAULT\_BBB\_PATH : répertoire qui contiendra les fichiers vidéos générés par bbb-recorder et les fichiers de logs inhérents à la conversion
- BBB\_SERVER\_URL : URL du serveur BigBlueButton ou Scalelite où est positionné les présentations Web et l'API bbb
- BBB\_SECRET\_KEY : la clé de sécurité de BigBlueButton (côté BigBlueButton, il est possible d'obtenir cette clé via la commande *sudo bbb-conf secret*).
- DEFAULT\_BBB\_TYPE\_ID : type par défaut de la vidéo générée
- BBB\_USERNAME\_FORMAT : format du nom d'utilisateur dans BBB. *Disponible à partir de la version v2.7.2.*

Valeurs possibles : 'first\_name last\_name', 'last\_name first\_name'



#### BBB\_USERNAME\_FORMAT

*Exemple 1* : si dans le serveur BBB, vous avez un utilisateur au format 'John Doe', mettez BBB\_USERNAME\_FORMAT = 'first\_name last\_name'

*Exemple 2* : si dans le serveur BBB, vous avez un utilisateur au format 'Doe John', mettez BBB\_USERNAME\_FORMAT = 'last\_name first\_name'

- BBB\_NUMBER\_DAYS\_BEFORE\_DELETE : Nombre de jours avant la suppression des sessions et des utilisateurs associés, non déjà publiés. *Pour ne pas supprimer les anciennes sessions, utiliser la valeur 0.*

Concernant le répertoire contenant les fichiers vidéos générés par bbb-recorder (**DEFAULT\_BBB\_PATH**), il est à créer manuellement - en même temps que son sous-répertoire des **logs** - avec les lignes de commande suivantes; n'hésitez pas à les modifier à votre convenance selon votre architecture système et vos droits :

## Création du répertoire DEFAULT\_BBB\_PATH et son sous-répertoire logs

```
%userpod%@ts-sun:~/ $ mkdir /data/www/%userpod%/bbb-recorder
%userpod%@ts-sun:~/ $ mkdir /data/www/%userpod%/bbb-recorder/logs
%userpod%@ts-sun:~/ $ chown %userpod%:www-data /data/www/%userpod%/bbb-recorder/logs
# Ou, selon l'environnement : %userpod%@ts-sun:~/ $ chown %userpod%:nginx /data/www/%userpod%/bbb-recorder/logs
```



### Répertoires

Il est vrai qu'une création automatique de ces répertoires auraient pu être possible, mais aux vues des problèmes que cela aurait pu engendrer, en lien avec l'architecture et les droits, il m'a paru préférable que l'administrateur de Pod créé ces 2 répertoires à la main.

Il sait ce qu'il fait et pourra ainsi choisir son emplacement, ses droits ou autres.

## Mise en place du job CRON

Comme expliqué préalablement, le système repose principalement sur un job CRON. Ce job CRON est à installer sur un serveur Pod de votre choix et devra tourner régulièrement (toutes les 5 minutes me paraît un délai correct).

**i** Un délai de 5 minutes est possible en cas d'utilisation du système de publication des présentations Web, mais il est préférable d'utiliser un délai de **2 minutes** pour le système de diffusion de webinaires (cf. <https://www.esup-portail.org/wiki/x/BgApQQ>).

Le script à lancer est positionné dans `django_projects/podv2/pod/video/management/commands/bbb.py` et permet de gérer les enregistrements effectués par BigBlueButton.

Personnellement, mon CRON est configuré de la sorte :

### Job CRON

```
crontab -e
*/5 * * * * /usr/bin/bash -c 'export WORKON_HOME=/data/www/%userpod%/.virtualenvs; export
VIRTUALENVWRAPPER_PYTHON=/usr/bin/python3.6; cd /data/www/%userpod%/django_projects/podv2; source /usr/bin
/virtualenvwrapper.sh; workon django_pod; python manage.py bbb main'
```

Voici les explications complémentaires sur ce que réalise cette tâche :

### Étape 1 : récupération des sessions BBB et des utilisateurs BBB

Ce script réalise une connexion au serveur BBB / Scalelite pour obtenir des informations sur les réunions en cours et enregistre les informations dans la base de données de Pod.

Ceci est utile pour obtenir les sessions actuelles et la liste des modérateurs (cf. l'avertissement plus bas sur la correspondance des utilisateurs) de ces réunions.

### Étape 2 : recherche des enregistrements dans BBB

Le script recherche ensuite les enregistrements disponibles pour les réunions sauvegardées. Par précaution, cela ne recherche que les présentations Web des réunions réalisées depuis moins de **4 jours**.

L'idée des 4 jours est d'éviter de traiter les enregistrements qui ont été supprimés ou avec de mauvaises données dans la base de données.



### Statut recording

Au final, l'API BigBlueButton présente un statut d'enregistrement (attribut **recording**) mais qui s'avère peu fiable. Dans l'environnement UM, il est quasiment tout le temps à vrai, alors que les sessions n'ont pas été enregistrées.

### Étape 3 : recherche des correspondances entre les utilisateurs BBB et ceux de Pod

Le script tente de réaliser une correspondance entre les utilisateurs BBB - connus par leur prénom et nom seulement - et les utilisateurs existants de pod.

À chaque utilisation de ce script, nous recherchons les correspondances pour les utilisateurs BBB qui ne sont pas encore connus en tant qu'utilisateurs de Po

### Les utilisateurs dans BigBlueButton

Je me répète, mais les informations concernant les modérateurs dans BigBlueButton dépendent du client BBB utilisé : **Greenlight** ou le plugin **mod\_bigbluebuttonbn pour Moodle**.

Le système réalisé n'a été testé qu'avec le plugin **mod\_bigbluebuttonbn pour Moodle** (CASifié, donc les données proviennent de notre annuaire LDAP); cela signifie que les modérateurs sont définis - dans mon cas - sous la forme "Prenom Nom".

Il est possible de paramétrer ce format, via le paramètre **BBB\_USERNAME\_FORMAT**. A l'heure actuelle, les formats "Prenom Nom" ou "Nom Prenom" peuvent être gérés via ce paramétrage.

 **Ce point est crucial pour que le système fonctionne correctement : une correspondance doit exister sur le "Prenom Nom" ou "Nom Prenom" (selon la configuration BBB\_USERNAME\_FORMAT) des modérateurs de BigBlueButton et le "Prenom Nom" des utilisateurs dans la base de données de Pod.**

 A priori, cela devrait pouvoir fonctionner avec **Greenlight** si celui-ci est configuré pour utiliser l'**annuaire LDAP** de l'établissement et les champs **givenName** et **sn**.

Il est également possible de modifier le code source pour que cette correspondance se réalise, selon le contexte de l'établissement cible.

## Étape 4 : publication des vidéos

Le script vérifie ensuite l'existence de fichiers vidéos dans le répertoire - paramétré via **DEFAULT\_BBB\_PATH**. Cela signifie que ces fichiers vidéo ont été générés par **bbb-recorder** - configuré via **DEFAULT\_BBB\_PLUGIN**.

Si des fichiers vidéos sont trouvés, le script va lancer la tâche d'encodage correspondante pour les enregistrer en tant que vidéo Pod.

### Ne pas oublier la publication par l'utilisateur

Il faut bien comprendre que cette étape 4 vient après la **publication** de présentations Web BigBlueButton par l'utilisateur, dans Pod.

Au final, en publiant une présentation Web BigBlueButton, cette présentation est convertie en fichier vidéo qui sera enregistré dans le répertoire, paramétré via **DEFAULT\_BBB\_PATH**.

L'étape 4 du script, lancé par le CRON, **recupèrera alors ce fichier vidéo et la convertira en vidéo Pod**, qui apparaîtra alors dans "Mes vidéos" de l'utilisateur.

Ainsi, 2 encodages sont réalisés (1 pour convertir la présentation Web BBB en fichier vidéo, 1 pour convertir le fichier vidéo en vidéo Pod), mais, aux vues de mes besoins, je n'ai pas trouvé d'autres solutions.

## Etape 5 : suppression des sessions BBB

Si vous avez positionné une valeur pour **BBB\_NUMBER\_DAYS\_BEFORE\_DELETE** dans votre fichier de configuration, le script va alors supprimer les anciennes sessions BigBlueButton - **non publiées** - antérieures au nombre de jours défini, ainsi que les utilisateurs associés.

*Par exemple : si vous avez positionné **BBB\_NUMBER\_DAYS\_BEFORE\_DELETE = 90**, les sessions BigBlueButton vieilles de plus de 90 jours, qui n'ont pas été publiées précédemment, seront automatiquement supprimées.*

 Si ce paramètre **BBB\_NUMBER\_DAYS\_BEFORE\_DELETE** n'existe pas ou a pour valeur 0, aucune suppression n'est réalisée.

Finalement, s'il y a eu au moins une erreur, un e-mail est envoyé aux administrateurs de Pod.

## Synthèse du processus global



## Exploitation

Dans le cas normal, où tout se passe bien, ce système ne devrait pas engendrer d'exploitation particulière et vous ne recevrez d'emails de la part du système qu'en cas d'erreurs (typiquement si le serveur BigBlueButton ou Scalelite ne répond plus...).

Cependant, une interface d'administration a été prévue afin de suivre des sessions BigBlueButton mais également pour réaliser l'exploitation en cas d'incidents.

## L'interface d'administration

Via l'administration de Pod, vous aurez accès à un nouveau menu, **BBB**, qui contient les options **Sessions** et **Utilisateurs**.

# Administration de Pod

## Administration du site

**AUTHENTIFICATION ET AUTORISATION**

<b>Groupes</b>	<a href="#" style="color: #2c5e8c; text-decoration: none;">+ Ajouter</a> <a href="#" style="color: #2c5e8c; text-decoration: none; margin-left: 10px;">✎ Modifier</a>
<b>Utilisateurs</b>	<a href="#" style="color: #2c5e8c; text-decoration: none;">+ Ajouter</a> <a href="#" style="color: #2c5e8c; text-decoration: none; margin-left: 10px;">✎ Modifier</a>

**BBB**

<b>Sessions</b>	<a href="#" style="color: #2c5e8c; text-decoration: none;">+ Ajouter</a> <a href="#" style="color: #2c5e8c; text-decoration: none; margin-left: 10px;">✎ Modifier</a>
<b>Utilisateurs</b>	<a href="#" style="color: #2c5e8c; text-decoration: none;">+ Ajouter</a> <a href="#" style="color: #2c5e8c; text-decoration: none; margin-left: 10px;">✎ Modifier</a>

## Les sessions BigBlueButton

Voici l'interface de listing des sessions réalisées dans BigBlueButton, et récupérées via le job :

Sélectionnez l'objet Session à changer

AJOUTER SESSION +

Action :  Envoyer 0 sur 100 sélectionné

<input type="checkbox"/>	ID	2	DATE	3	NOM DE LA SESSION	ÉTAPE DE L'ENCODAGE	ENREGISTRÉE	ENREGISTREMENT DISPONIBLE	UTILISATEUR	1
<input type="checkbox"/>	666		6 octobre 2020 13:34		Home Room	3	✔	✔	Loic	
<input type="checkbox"/>	372		14 septembre 2020 10:44		Seance introductive aux projets tutorés 2020/2021	3	✔	✔	Loic	
<input type="checkbox"/>	36		7 septembre 2020 11:09		TEST BBB -> Pod	3	✔	✔	Loic	
<input type="checkbox"/>	25		1 septembre 2020 15:39		Home Room	3	✔	✔	Loic	
<input type="checkbox"/>	22		4 septembre 2020 14:55		Home Room	3	✔	✔	Loic	
<input type="checkbox"/>	19		4 septembre 2020 14:25		Home Room	3	✔	✔	Loic	
<input type="checkbox"/>	11		4 septembre 2020 07:54		Questions/Réponses durant les cours	3	✔	✔	Loic	
<input type="checkbox"/>	9		3 septembre 2020 16:42		Home Room	3	✔	✔	Loic	
<input type="checkbox"/>	1		3 septembre 2020 13:06		Cours introduction et paléocécologie	3	✔	✔	Loic	
<input type="checkbox"/>	6		3 septembre 2020 16:20		formation BBB 28 août - Créer un cours avec Moodle	3	✔	✔	Test Portail	

Il est possible de modifier les données si nécessaire.

# Administration de Pod

Accueil > Bbb > Sessions > TEST BBB -> Pod - eb20ca5b38741336383a2bd548e240ccb777d066

## Modification de Session

Identifiant de la session :

eb2 :b777

Identifiant de la session BBB

Identifiant interne de la session :

c59 5142

Identifiant interne de la session BBB

Nom de la session :

TEST BBB -> Pod

Nom de la session BBB

Date :

Date : 07/09/2020 Aujourd'hui | 

Heure : 11:09:38 Maintenant | 

Étape de l'encodage :

3

Étape d'encodage pour la conversion de la présentation BBB en fichier vidéo.

Enregistrée

La présentation BBB a-t-elle été enregistrée ?

Enregistrement disponible

Un enregistrement de la présentation BBB est-il disponible ?

URL de l'enregistrement :

https://. .r/playback/pre

URL de l'enregistrement de la présentation BBB.

URL de la vignette :

https://. r/presentation

URL de la vignette correspondant à l'enregistrement de la présentation BBB.

Utilisateur :

Loic  

Utilisateur ayant converti la présentation BBB en fichier vidéo.



### Exploitation

Par exemple, via cette interface, il est possible de relancer le processus global pour cette session en modifiant l'étape de l'encodage à 0 et en ne choisissant aucun utilisateur (dans la liste de sélection : -----). Suite à ces changements, il sera possible de republier la présentation Web.

## Nouveauté de la version 2.8 de Pod

**i** Il est maintenant possible à un administrateur de réencoder une session BigBlueButton qui a déjà été encodée par un utilisateur.

Pour ce faire, il suffit de sélectionner le ou les sessions BBB à réencoder dans le module d'administration et d'utiliser la fonction de réencodage :

Administration de Pod BIENVENUE, LOIC. VOIR LE SITE / DÉCONNEXION

Accueil · Bbb · Sessions

Sélectionnez l'objet Session à changer AJOUTER SESSION +

Rechercher 25 résultats (222 résultats)

Action : ----- Envoyer 3 sur 25 sélectionnés

ID	NOM DE LA SESSION	ÉTAPE DE L'ENCODAGE	ENREGISTRÉE	ENREGISTREMENT DISPONIBLE	UTILISATEUR
221	Test salon 2	3	✓	✗	Loic
217	15 février 2021 16:19 Webinaire de test 01	3	✓	✗	Loic
214	15 février 2021 16:11 Webinaire de test 01	3	✓	✗	Loic
179	15 février 2021 10:49 Webinaire de test 01	3	✓	✗	Loic
169	15 février 2021 10:25 Webinaire de test 01	3	✓	✗	Loic

**!** Réencoder une session qui n'a pas déjà été encodée par un utilisateur ne fonctionnera pas : l'idée de cette fonctionnalité est de permettre à un administrateur de réencoder une session en cas de problème quelconque.

## Les utilisateurs BigBlueButton

Voici l'interface de listing des modérateurs de sessions dans BigBlueButton, et récupérées via le job :

Administration de Pod BIENVENUE, LOIC. VOIR LE SITE / DÉCONNEXION

Accueil · Bbb · Utilisateurs

Sélectionnez l'objet Utilisateur à changer AJOUTER UTILISATEUR +

Action : ----- Envoyer 0 sur 100 sélectionné

ID	NOM COMPLET	RÔLE DE L'UTILISATEUR	USERNAME / IDENTIFIANT UTILISATEUR	SESSION	UTILISATEUR
12	Antoine	MODERATOR		Questions/Reponses durant les cours - 24L	Loic
1278	loic	MODERATOR	p00000013909	Dernier test LB - et	Loic
1273	loic	MODERATOR	p00000013909	Home Room - et	Loic
29	loic	MODERATOR	p00000013909	TEST BBB -> Pod - et	Loic

Il est possible de modifier les données si nécessaire.

## Administration de Pod

Accueil · Bbb · Utilisateurs · loic - MODERATOR

### Modification de Utilisateur

Session :	<input type="text" value="Dernier test LB - e"/>	▼ ✎ +
Nom complet :	<input type="text" value="loic"/>	
<small>Nom complet de l'utilisateur dans BBB.</small>		
Rôle de l'utilisateur :	<input type="text" value="MODERATOR"/>	
<small>Rôle de l'utilisateur dans BBB.</small>		
Username / Identifiant utilisateur :	<input type="text" value="p"/>	
<small>Nom d'utilisateur / ID utilisateur, si l'utilisateur BBB correspond à un utilisateur existant de Pod.</small>		
Utilisateur :	<input type="text" value="Loic"/>	▼ ✎ +
<small>Utilisateur dans la base de données de Pod, si l'utilisateur est trouvé.</small>		



#### Exploitation

Par exemple, via cette interface, il est possible d'affecter un utilisateur à une session, voire modifier une correspondance si celle-ci n'a pas été trouvée ou est erronée.

#### Exploitation du script lancé par le job CRON

Si nécessaire, il est possible de voir les traitements réalisés par le script - qui doit normalement être exécuté via un job CRON.

Pour ce faire :

- soit il faut configurer Pod pour être en **mode debug** (cf. `settings_local.py`).
-  **Attention** : la mode debug ne doit pas être activé dans un environnement de production.
- soit modifier directement le script `/django_projects/podv2/pod/video/management/commands/bbb.py` pour qu'il affiche directement les traitements réalisés (cf. fonction `print_if_debug`).

Il ne reste plus qu'à exécuter ce script `/django_projects/podv2/pod/video/management/commands/bbb.py` manuellement via la commande `python manage.py bbb main` :

#### Exécution du script manuellement, compte %userpod%

```
cd /data/www/%userpod%/django_projects/podv2 workon django_pod python manage.py bbb main
```

En mode debug, le script affiche l'ensemble des traitements réalisés :

```
*** Check BBB/Scalelite actual meetings ***
Request on URL : https://...fr/bigbluebutton/api/getMeetings?checksum... , status : 200

- Meeting : ebd2081d1407b94b
+ Meeting already exists in Pod database.
+ User already exists in Pod database : Tiberiu

- Meeting : cd4a3f06414b30e
+ Meeting already exists in Pod database.
+ User already exists in Pod database : Sylvie

- Meeting : e8aba1671c3678
+ Meeting already exists in Pod database.
+ User already exists in Pod database : Valerie
+ User already exists in Pod database : Valerie

- Meeting : 65e7a18958a41
+ Create the meeting in Pod database. internal_meeting_id : 65e7a18958a41

- Meeting : 96bb4de76512
+ Meeting already exists in Pod database.
+ User already exists in Pod database : Aurelie

- Meeting : 5120478e984e
+ Meeting already exists in Pod database.
+ User already exists in Pod database : Marie-george
+ User already exists in Pod database : Marie-george

- Meeting : 78ba4369be2
+ Meeting already exists in Pod database.
+ User already exists in Pod database : Marina

- Meeting : bcbad06d69c
+ Meeting already exists in Pod database.
+ User already exists in Pod database : Souhila

- Meeting : ace4d55453d
+ Meeting already exists in Pod database.
+ User already exists in Pod database : Bruno
```

## Exploitation courante

Ce système utilise les différents fichiers de logs suivants :

- Fichier de logs **Celery** : le nom de ce fichier de logs - disponible sur les serveurs d'encodage - dépend de votre configuration (worker), mais typiquement, il s'agit de : **/var/log/celery/worker1.log**

Ce fichier permettra de surveiller l'encodage et des présentations Web BigBlueButton et des encodages standards.

- Fichier de logs en lien avec **bbb-recorder** : pour chaque encodage de la présentation Web BigBlueButton, un fichier de logs est créé.

Ce fichier de logs est créé dans le répertoire configuré via **DEFAULT\_BBB\_PATH/logs** (dans mon cas : **/data/www/%userpod%/bbb-recorder/logs**).

A chaque encodage, un fichier est généré; son nom correspondant à l'identifiant interne (*internal\_meeting\_id*) de BigBlueButton de la session concernée.

Vous pouvez utiliser la base de données ou l'interface d'administration pour retrouver cette information.