



Projet OBS Multistreaming

Daniel Gaël

Moulin Kilian

Tholimet Alexis

Université Paris-Est Créteil

Réalisation d'applications : conception, développement, validation

Table des matières

Table des matières	2
Introduction	3
Objectif du projet	3
Méthode 1 : Application	4
Méthode 2 : Script OBS	6
Méthode 3 : Script lançant le plugin multi-RTMP	7
Conclusion	8
Méthode 1 : Application	8
Méthode 2 : Script OBS	8
Méthode 3 : Plugin multi-RTMP	8
Perspectives	8

Introduction

De nos jours, les plateformes de streaming permettent de diffuser du contenu en ligne via des sites spécifiques. Par exemple, en utilisant le logiciel OBS et en récupérant la clé de streaming de Twitch, il est possible de permettre aux utilisateurs de cette plateforme de nous regarder effectuer diverses activités. Jusqu'à récemment, Twitch ne permettait de diffuser que sur sa propre plateforme, assurant ainsi l'exclusivité des streamers utilisant son site ou son application mobile, notamment pour les partenaires.

Cependant, cela a changé récemment, et il est maintenant possible de diffuser simultanément sur plusieurs plateformes, comme YouTube et Twitch. Le problème est que cette configuration n'est ni simple ni rapide à mettre en place, car il faut configurer et démarrer le stream sur plusieurs outils différents, ce qui implique d'avoir de nombreuses pages ouvertes, rendant l'expérience peu ergonomique. Néanmoins, cette approche augmente la visibilité des personnes souhaitant diffuser du contenu sur plusieurs médias, en touchant des audiences non habituées à une plateforme spécifique.

Objectif du projet

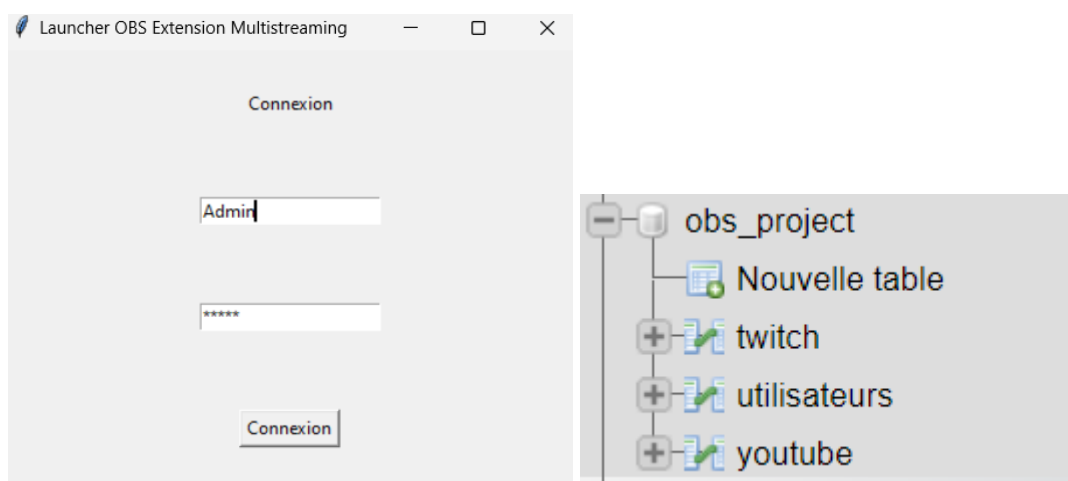
Notre objectif est de créer une application web, un script ou bien un plugin qui permet de diffuser simultanément sur plusieurs plateformes en un seul clic. Certaines personnes peuvent avoir des difficultés à mettre en place leur diffusion sur plusieurs services en même temps pour atteindre un public plus large, car elles ne sont pas toutes expertes en la matière. Avoir plusieurs pages web ouvertes pour configurer un stream, avec des configurations et des chats différents, n'est pas très ergonomique.

Nous visons à regrouper toutes ces fonctionnalités sous une seule interface, facilitant ainsi la vie des utilisateurs. Une application qui centralise tous les streams sur une seule page. Ce projet sera basé sur des technologies telles que Python ou C++, car elles sont compatibles avec WebSocket d'OBS.



Méthode 1 : Application

Notre premier choix pour la création de ce projet fut une application en python en utilisant TKinter afin de créer quelque chose de tangible pour l'utilisateur, un moyen alternatif que le terminal de commande. En effet, nous avons pensé à une fenêtre de connexion en utilisant un utilisateur et un mot de passe avec une base de données (MySQL).

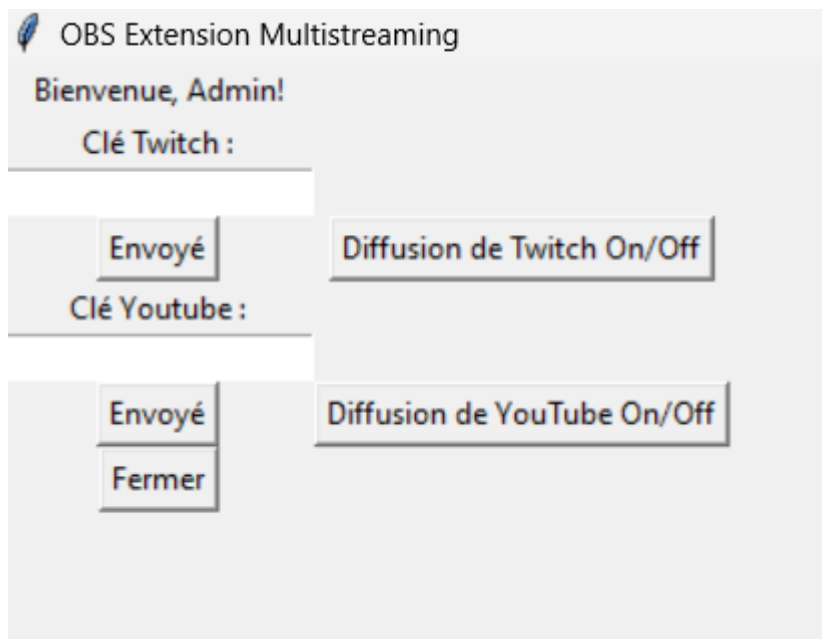


Celle-ci enregistrait les utilisateurs, et toutes les données en rapport avec nos deux principales plateformes de streaming en vue de notre projet, Twitch et Youtube.

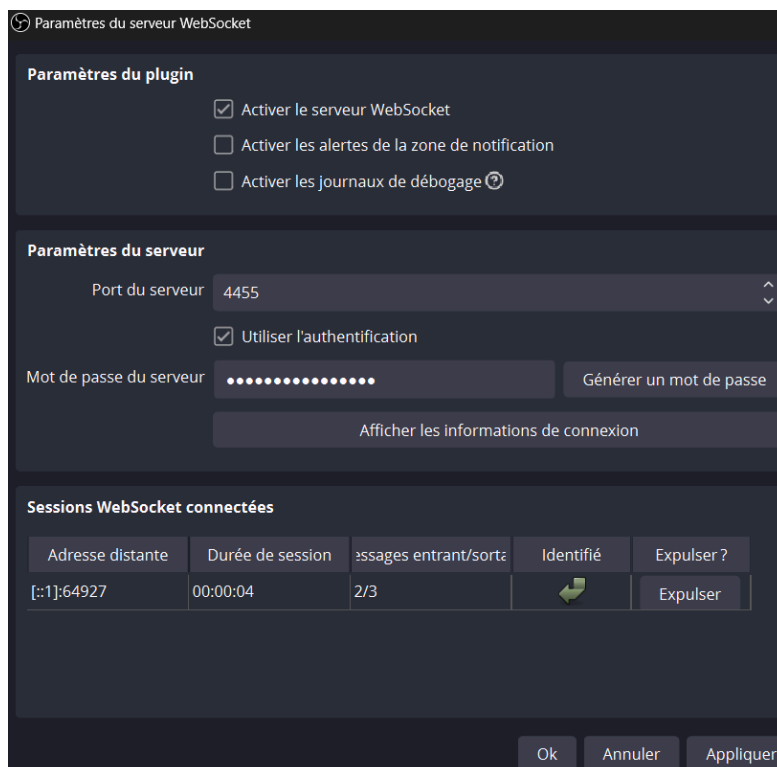
#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra	Action
<input type="checkbox"/>	1 id	int(255)			Non	Aucun(e)		AUTO_INCREMENT	Modifier Supprimer Plus
<input type="checkbox"/>	2 username	varchar(15)	utf8mb4_general_ci		Non	Aucun(e)			Modifier Supprimer Plus
<input type="checkbox"/>	3 twitch_key	varchar(100)	utf8mb4_general_ci		Non	Aucun(e)			Modifier Supprimer Plus
<input type="checkbox"/>	4 OAuth_Key	varchar(100)	utf8mb4_general_ci		Non	Aucun(e)			Modifier Supprimer Plus
<input type="checkbox"/>	5 client_id	varchar(255)	utf8mb4_general_ci		Non	Aucun(e)			Modifier Supprimer Plus
<input type="checkbox"/>	6 client_secret	varchar(100)	utf8mb4_general_ci		Non	Aucun(e)			Modifier Supprimer Plus

Toutes les clés de streaming, les tokens de vérification, client secret étaient stockés dans notre BDD grâce au profil de la personne connectée

La construction de notre application était simple, l'utilisateur renseigne ses différentes informations dessus, afin de lier une connexion aux plateformes, n'avait plus qu'à appuyer sur le bouton de diffusion/arrêt et pouvait streamer librement sur plusieurs sites en même temps



Une connexion au web socket de OBS était primordial, en suivant la documentation nous avons effectué cette tâche.

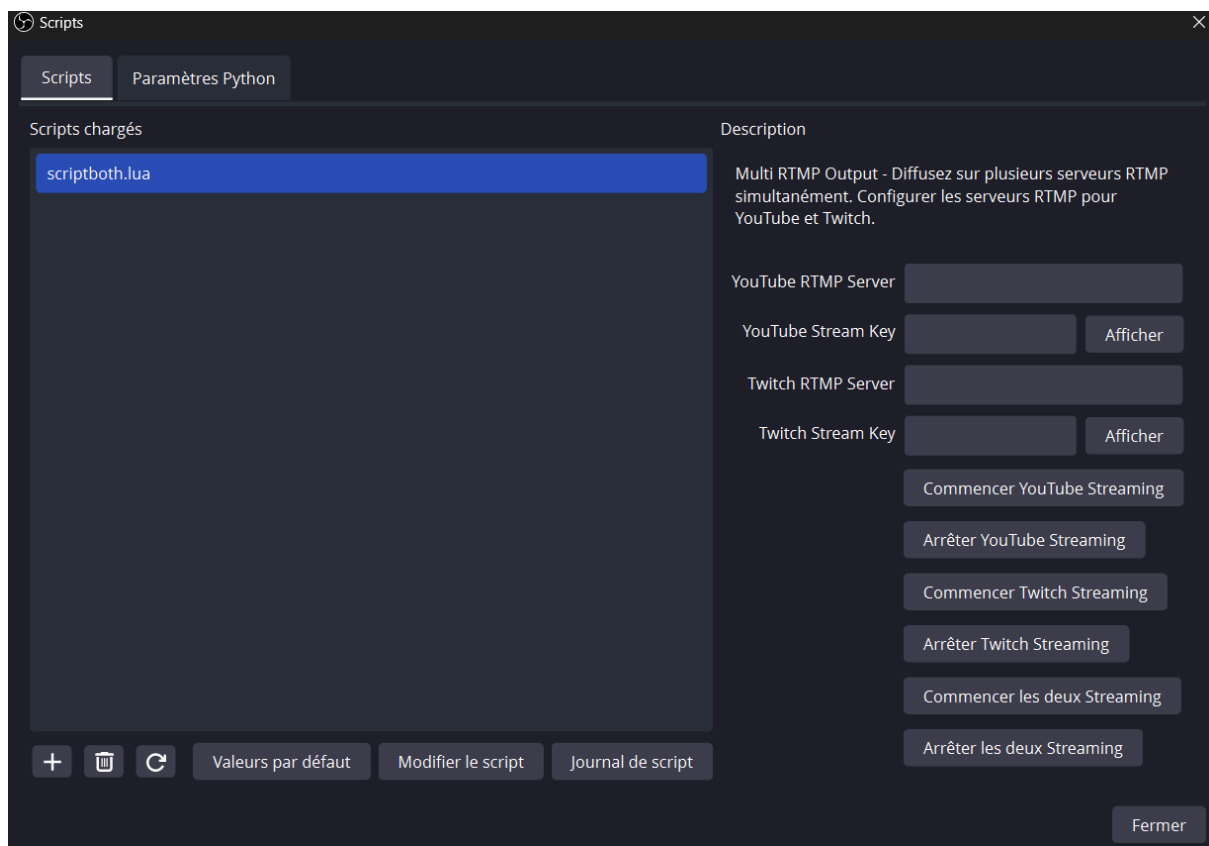


Comme on peut le constater, une liaison s'était créée sur l'application OBS mais aucun flux n'était redirigé vers les plateformes concernées.

Cependant, nous nous sommes rendu compte que des méthodes permettant le démarrage et l'arrêt d'un stream, ne sont pas possibles d'un point de vue extérieur. Après maintes et maintes recherches sur le sujet, nous ne trouvions rien de concret en termes d'informations et de documentation afin de démarrer ne serait-ce qu'un flux vidéo en utilisant des fonctions liées à OBS.

Méthode 2 : Script OBS

Notre second choix s'est porté sur un script directement implanté dans la section Script d'OBS. Notre but était simple, avoir des champs pour les plateformes où nous voulions diffuser du contenu, en récupérant les clés RTMP serveur et les clés de diffusion propre à chacune.



Une fois celles-ci récupérées, il nous fallait juste appuyer sur le bouton "Démarrer le Stream" du logiciel OBS, et les différents flux vidéos auraient été redirigés vers leurs sites respectifs afin d'avoir un multistream.

Nous avons eu un problème à ce niveau là puisque les vidéos sont encodées de façon différentes pour Twitch et Youtube, il fallait encoder les vidéos d'une certaine façon afin d'assurer le bon fonctionnement des flux vidéos.

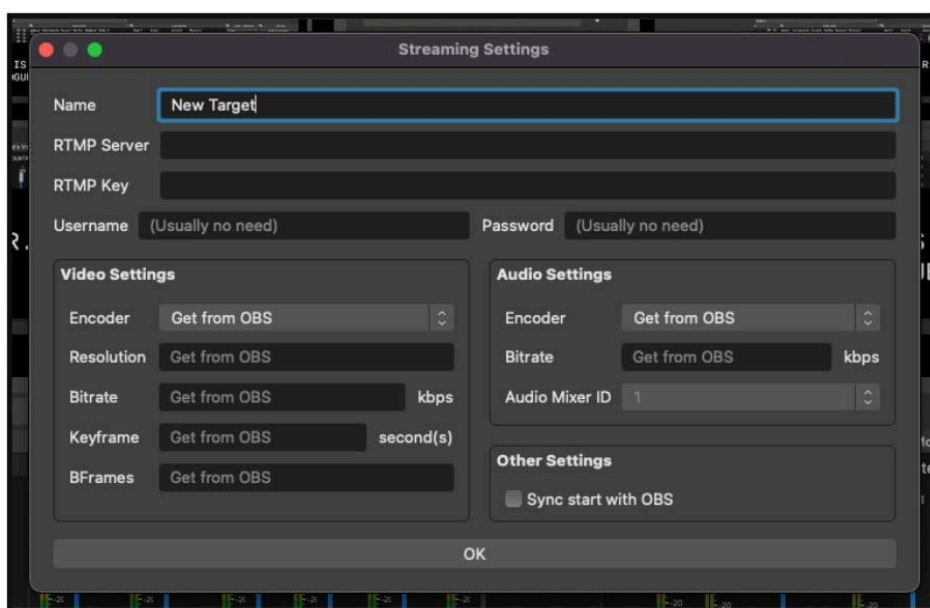
Le deuxième problème a été la gestion de deux flux vidéos alors que le logiciel est paramétré pour un flux principal que nous sommes censés référencer dans les paramètres d'OBS. Les clés renseignées pour le serveur et celle de diffusion ne sont pas prises en compte car le logiciel prend en priorité celle qui est paramétré dans le logiciel. En ne mettant aucune information, une erreur systématique en résultait et de même en donnant des informations test pour passer au dessus avec nos flux.

Pour conclure, le logiciel prenait systématiquement comme flux principal, les informations du logiciel, et non les données renseignées par l'utilisateur.

Méthode 3 : Script lançant le plugin multi-RTMP

Notre troisième choix s'est porté sur le plugin existant "Multiple RTMP outputs plugin", permettant le multistreaming sur les différentes plateformes de streaming. Celui-ci répondait parfaitement à nos besoins pour notre script directement sur OBS, et nous souhaitions l'utiliser comme base pour développer un script lançant ses fonctionnalités.

Cependant, après une analyse approfondie du code, nous avons constaté que la tâche de créer notre propre plugin était extrêmement complexe. La documentation, principalement en japonais, rendait la compréhension difficile, et le langage C++ ne facilitait pas les choses. Ce plugin prenait en compte l'encodage vidéo pour chaque plateforme, ainsi que d'autres paramètres complexes à configurer. Face à ces défis techniques et au manque de temps croissant, nous n'avons pas réussi à développer notre propre plugin.



Conclusion

En conclusion, notre projet avait pour objectif de simplifier la diffusion simultanée sur plusieurs plateformes de streaming, comme Twitch et YouTube, via une seule interface ergonomique. Malgré des efforts considérables et plusieurs approches explorées, nous avons rencontré des défis techniques majeurs qui nous ont empêchés de parvenir à une solution fonctionnelle et intégrée.

Méthode 1 : Application

Notre première approche, consistant à développer une application en Python avec TKinter, a permis de créer une interface utilisateur conviviale et d'établir des connexions aux plateformes de streaming via WebSocket d'OBS. Cependant, la difficulté à trouver des méthodes externes fiables pour démarrer et arrêter un flux vidéo nous a empêchés de rediriger les flux vers les plateformes concernées.

Méthode 2 : Script OBS

Ensuite, nous avons tenté de développer un script directement intégré dans OBS pour gérer les clés RTMP et de diffusion des différentes plateformes. Cette méthode a rencontré des obstacles liés à l'encodage vidéo différencié requis par chaque plateforme et la gestion des flux multiples dans OBS. Les configurations et priorités du logiciel OBS ne permettaient pas de rediriger les flux vidéo correctement, malgré nos tentatives d'adaptation des paramètres.

Méthode 3 : Plugin multi-RTMP

Enfin, nous avons exploré l'utilisation du plugin existant "Multiple RTMP outputs plugin" pour OBS, qui semblait offrir les fonctionnalités de multistreaming souhaitées. Cependant, la complexité du code en C++ et la documentation en japonais ont rendu cette approche difficile à mettre en œuvre et à personnaliser selon nos besoins spécifiques.

Perspectives

Bien que notre projet n'ait pas abouti à une solution opérationnelle, ces tentatives nous ont permis de mieux comprendre les défis techniques du multistreaming et les limitations actuelles des outils disponibles. Nous recommandons de poursuivre les recherches dans ce domaine, peut-être en collaboration avec des développeurs expérimentés en C++ ou en explorant de nouvelles technologies et frameworks de streaming qui pourraient offrir une meilleure compatibilité et flexibilité.

Notre expérience souligne l'importance d'une documentation claire et accessible, ainsi que le besoin d'outils plus robustes pour les créateurs de contenu souhaitant diffuser simultanément sur plusieurs plateformes. Nous espérons que notre travail pourra servir de base pour de futurs développements et innovations dans le domaine du streaming multi-plateformes.