**Guide utilisateur IousephAsAService**

Bienvenue dans votre nouveau service de Streaming musical tout en un.

Dans ce document, vous trouverais comment exécuter le projet à partir du code source ou des exécutables et comment consommer l’API REST IouserphAsAService et utiliser l’interface graphique IousephFX que nous fournissons pour profiter du service.

Ce service vous permet de profiter des larges collections de musiques des services **SoundCloud**, **Spotify** et **Deezer** à travers un seul. Vous pouvez créer un compte et effectuer des recherches de morceaux de musiques dans les 3 services, accéder et gérer vos propre playlists et d'écouter de la musique.

Cette version Beta comporte :

* La création de compte sur le service Iouseph,
* la recherche de musique,
* la gestion de playlists,
* l'écoute de musique.

L’application client vous permettra de profiter de ce service en toute simplicité.

1. **Importation et exécution du projet**

Pour récupérer le code source du projet, vous pouvez soit l’extraire du fichier « IousephAsAService.zip » fournie avec ce document, soit cloner le répertoire Github <https://github.com/EyMenZ/IousephAsAService.git> .Commencez par importer les deux projets dans votre IDE. Nous conseillons fortement d’utiliser e(fx)clipse (http://efxclipse.bestsolution.at/install.html#all-in-one) qui gère les dépendances liées à JavaFX du projet IousephFX. Pour exécuter le serveur, vous lancez un Run sur la classe com.iouseph.SparkServerService du projet IousephAsAService. Et pour exécuter le client, vous lancez Run sur la classe com.iouspeh.MainApp du projet IousephFX.

Vous pouvez aussi utiliser les exécutables fournis avec ce document. Par lignes de commandes java –jar ##/\*\*.jar (\*\* étant le nom de l’exécutable et ## son chemin) ou par simple double clic sur le fichier. Nous conseillons d’utiliser la ligne de commande pour l’application serveur. Le fichier « IousephAsAService.jar » est l’exécutable de l’application serveur et « IousephFX.jar » celle du client.

Notez que vous pourrez exécuter une application par code source et l’autre par exécutable. Cela n’affectera pas le fonctionnement du projet, tant que vous ne modifiez pas le code source.

1. **Consommation du service**

Le code source du projet contient un fichier README explicatif des différentes requêtes GET et POST possibles, ainsi que le format du JSON retourné en réponse.

Nous conseillons l’implémentation de l’interface com.iouseph.api.Iapi si vous développer votre propre client. Elle représente toutes les méthodes qui est possible d’implémenter pour consommer le service Iouseph.

Pour le parsage du JSON, si vous développez un client Java, vous profiterez d’un parseur déjà existant, en utilisant directement com.iouseph.api.IousephParser ou vous pouvez implémenter l’interface com.iouseph.api.IParser, si vous voulez créer votre propre parseur. Pour d’autres langages, vous pourrez suivre son exemple.

1. **Utilisation du l’interface graphique**

L’interface graphique, représentée dans la figure ci-dessous, est intuitive et simple d’utilisation. (Les rectangles de couleur rouge et verte sont présentent à titre indicatif, et ne s’affichent pas lors de l’exécution de l’application).

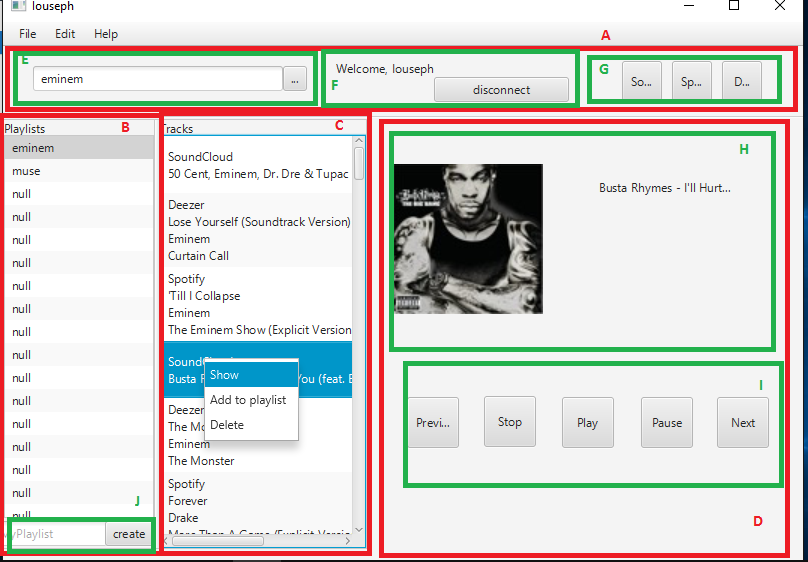


Figure : Interface graphique

Elle se compose de 4 parties principales. En haut, une partie horizontale pour la recherche et la connexion. À gauche, une liste qui contient les playlists d’un utilisateur connecté. Au milieu, une liste où s’affichera le résultat de la recherche. À droite, le lecteur de musique et le gestionnaire de playlists.

**Prise en mains :**

Ici, nous expliquons, comment interagir avec chacune des parties de l’interface décrites précédemment.

Partie A : se compose de 3 blocs :

* Bloc E : contient un champ de recherche où vous pouvez saisir un titre de chanson (complet ou en partie). Ensuite, en appuyant sur le bouton « ENTER » du clavier ou en en cliquant sur le bouton « Search » de l’interface, vous lancez la recherche. Le résultat de la recherche s’affiche sur la partie C.
* Bloc F : contient un champ texte « Welcome, Guest » et un bouton « connect ». Lorsque vous cliquez sur le bouton, l’interface de la Figure 2 s’affiche. Vous pouvez alors saisir vos paramètres de connexion, vous avez le choix de vous connecter ou de vous enregistrer. Si vous saisissez des informations erronées le texte de l’interface change pour une couleur rouge. Une fois les bonnes données entrées, la seconde interface se referme et votre pseudo s’affiche, le texte du bouton devient « disconnect » et les playlists enregistrées sur votre profil s’affichent dans la partie B.

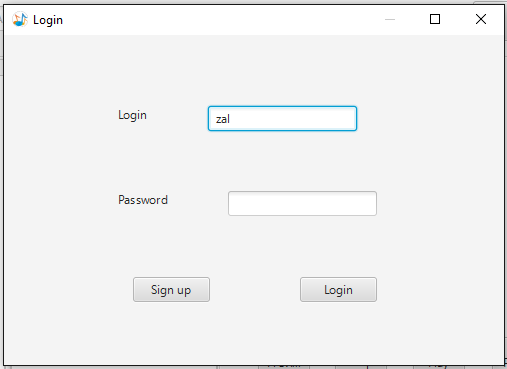


Figure : Interface de connexion

* Bloc G : contient trois boutons, un pour chaque service de streaming. Cette fonctionnalité n’est pas encore prise en charge, mais dans une prochaine version, vous pourrez activer/désactiver un service de la recherche.

Parie B : vous playlists (si vous êtes connecté) sont affichées ici, vous pouvez interagir avec cette liste de deux façons :

* Un clic sur un élément pour la sélectionner. Cela vous permettra d’y ajouter des chansons par la suite.
* Un clic droite affiche un menu contextuel où vous pouvez soit afficher la playlist dans la partie C soit supprimer la playlist.
* Vous pouvez ajouter une nouvelle playlist en écrivant son nom dans le champ de saisie en bas de la liste et en cliquant sur « create ».

Partie C : dans cette partie s’affiche le résultat de la recherche effectuée dans A ou du contenu d’une playlist :

* En cliquant sur un élément, vous l’affichez dans la partie D.
* Un clic droit sur un élément, affiche un menu contextuel où vous pouvez affiche la chanson dans D, ajouter la chanson à la playlist sélectionnée ou la supprimer de cette même playlist (modifier une playlist consiste à en supprimer des chansons pour l’instant).

Partie D : contient 2 blocs :

* Bloc H : l’image de l’album, le titre, le nom de l’artiste et le nom de l’album de la chanson sélectionnée dans C sont affichés ici.
* Bloc I : 3 boutons pour la lecture, la mise en pause et l’arrêt de l’écoute. Et un bouton pour revenir au morceau précédent et un cinquième pour le suivant.

1. **Futurs versions :**

Dans les prochaines versions, nous prévoyons d’ajouter les fonctionnalités suivantes (S pour le service / C pour le client / A pour autre):

* Gestion de profil utilisateur (S+C),
* Synchronisation avec les comptes SoundCloud, Spotify et Deezer (S+C),
* Choix des services utilisés (S+C),
* Ajout de services (S+C),
* Créer une authentification sécurisée (S),
* Améliorer l’interface graphique et ajouter des animations (C),
* Améliorer la recherche (récupérer plus de chansons, recherche par album et par artiste) (S+C),
* Ajouter des contrôles de saisie, des raccourcis et des messages pop-up (C),
* Connecter le plug-in du TP2 au service REST (C\*),
* Développer une version mobile (C\*),
* Création de site internet et comptes sur les réseaux sociaux (A).

ET plus encore.