

MISSIER Samuel

1 – Sujet

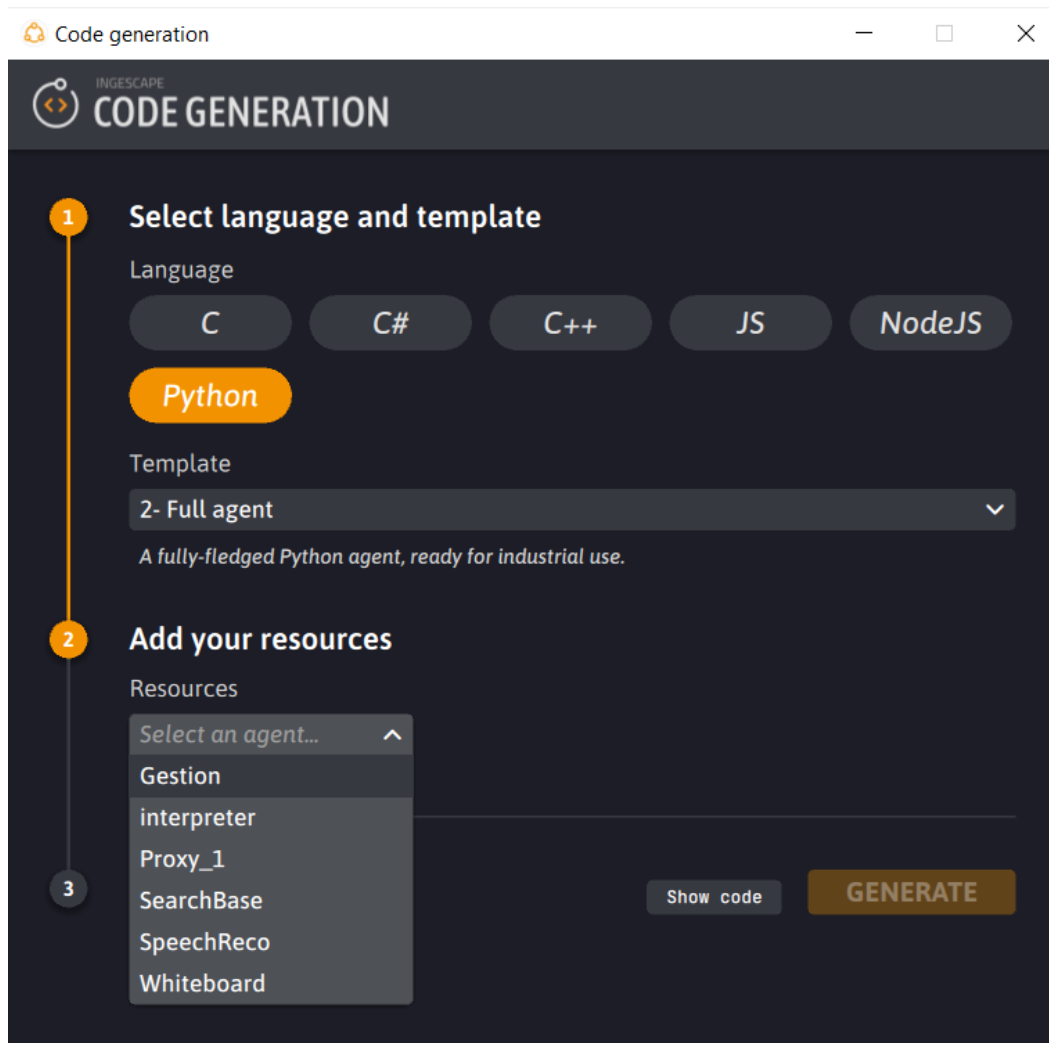
2 – Mise en œuvre

The diagram illustrates the system architecture with the following components and their interactions:

- Impulsion**: The starting point, connected to **SpeechReco** via a purple line.
- SpeechReco**: Contains inputs **Parole** (purple dot) and **Ecoule** (yellow dot). It outputs **reco** (green dot) to the **interpreter**.
- interpreter**: Receives **stringReco** (green dot) from **SpeechReco**. It outputs **titleWhiteBoard** (green dot) to the **Whiteboard** component. It also has internal outputs: **Depart** (green dot), **Arrivee** (green dot), and **fini** (yellow dot).
- SearchBase**: Receives **depart** (green dot) and **arrivee** (green dot) from the **interpreter**. It outputs **resultTravelAvion** (green dot), **resultTravelTrain** (green dot), and **resultTravelBus** (green dot) to the **Gestion** component.
- Gestion**: Receives results from **SearchBase** and outputs **result_voyage** (green dot) and **fin** (yellow dot).
- Whiteboard**: Receives **titleWhiteBoard** (green dot) from the **interpreter**. It has inputs **title** (green dot), **backgroundColor** (green dot), **chatMessage** (green dot), **clear** (purple dot), and **ui_command** (green dot). It outputs **lastChatMessage** (green dot), **lastAction** (green dot), and **ui_error** (green dot) to the **Service chat** component.

Nous avons fait en sorte que chacun des agents ne s'occupe que d'une seule tâche pour que la lecture du schéma soit plus lisible.

Ensuite, une option de ingescape nous permet de générer du code facilement selon nos caractéristiques :



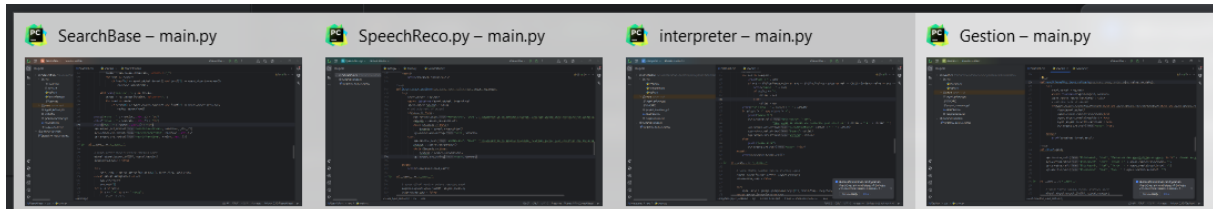
Dans notre cas, nous avons choisi de coder en python ; et sur conseil des intervenants, nous avons choisi le « full agent » comme template. Nous avons plus qu'à choisir les agents à générer.

```
def Ecoute_input_callback(iop_type, name, value_type, value, my_data):  
    try:  
        agent_object = my_data  
        assert isinstance(agent_object, SpeechReco)  
        agent_object.EcouteI = value  
        # add code here if needed
```

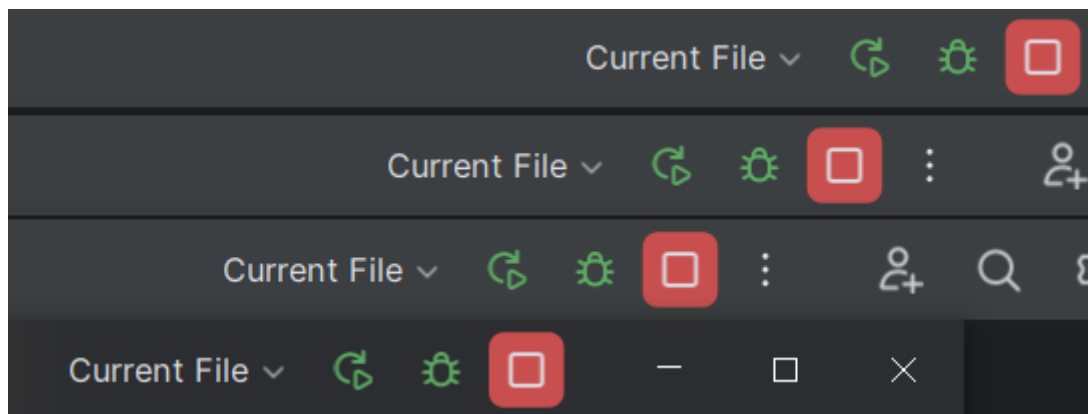
Une fois le code généré, il ne reste plus qu'à compléter les appels des callback « add code here », pour que le comportement espéré puisse être obtenu.

3 – Installation du projet

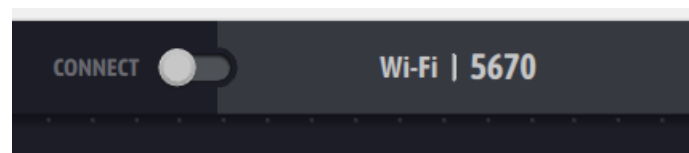
- Récupérer le répertoire du Whiteboard (fourni par ingénuité)
- Récupérer l'archive de notre projet « Ingescape.zip »
- Lancer « TravelBooking.igsplatform » (ref. Figure1)
- Lancer l'application “Whiteboard.exe” present dans le répertoire « Whiteboard »
- Dans votre IDE, ouvrir chacun des agents présents dans Ingescape\sandbox



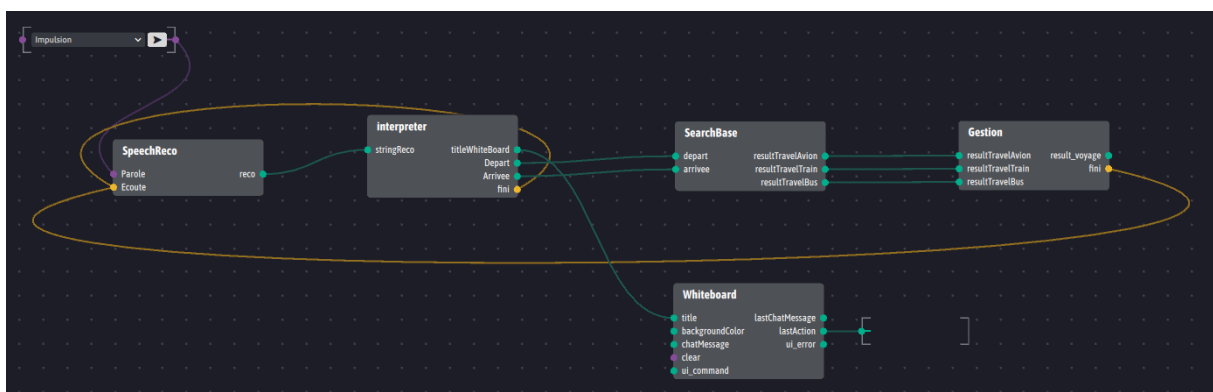
- Lancer les 4 main en même temps



- Sur la plateforme « TravelBooking », n’oubliez pas de vous connecter



- Maintenant que la plateforme, le Whiteboard et les agents sont lancés, vous devriez voir tous les agents allumés sur votre plateforme :



- Pour le premier lancement, il suffit d'aller activer l'impulsion d'en haut à gauche. Puis il n'y a plus qu'à suivre ce qui est indiqué sur le Whiteboard
 - Pour faire une requête :
 - Evoquer la ville de départ PUIS la destination (Exemple : Je veux aller de X à Y)
 - Les villes prises en considérations sont : Toulouse, Paris, Bordeaux, Lyon

4 - Résultats

A - Demande valide

The screenshot shows a web application window titled 'Whiteboard'. On the left, there is a menu with 'File', 'Debug', and a 'Clear' button. The main area displays 'Paris - Toulouse' in large, bold black text. On the right, a dark sidebar contains a chat-like interface with the following sections:

- interpreter**: Vous venez de lancer une recherche pour aller de Paris à Toulouse
- Gestion**: Recherche des possibilités en cours.
Format du resultat : Ville de depart, ville arrivee, heure depart, heure arrivee, prix
- Gestion**: Avion : [['Paris', 'Toulouse', '09:30', '10:45', '130.90']]
- Gestion**: Train : [['Paris', 'Toulouse', '10:00', '15:00', '68.90']]
- Gestion**: Bus : [['Paris', 'Toulouse', '10:00', '18:00', '52.50']]
- SpeechReco**: Traitement de la demande terminée, veuillez parler pour effectuer une nouvelle demande

Dans le cas où la requête utilisateur est valide, alors s'affiche en gros la ville de départ puis la ville d'arrivée. De plus, sur la partie droite du Whiteboard, on peut suivre les résultats liés à la demande et la consigne pour continuer d'utiliser l'application.

B - Demande invalide



Dans le cas d'une requête invalide, alors un message s'affiche pour l'indiquer, et demander une nouvelle demande.