

AMBLARD Lucas

BRISSAUD Tobias

MISSIER Samuel

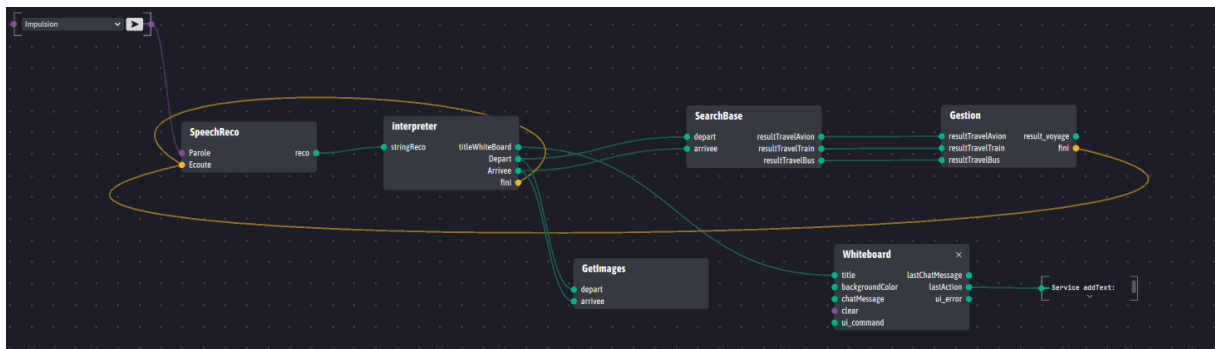
## Documentation Whiteboard Ingescape

### 1 – Sujet

Pour la réalisation de ce travail, nous devons choisir un projet qui nous permettrait d'utiliser les différentes options mises à disposition. Nous avons donc choisi de nous orienter sur de la recherche de voyage afin de pouvoir créer différents agents qui permettrait de mettre en œuvre de l'interaction distribuée et d'utiliser les moyens mis à disposition.

### 2 – Mise en œuvre

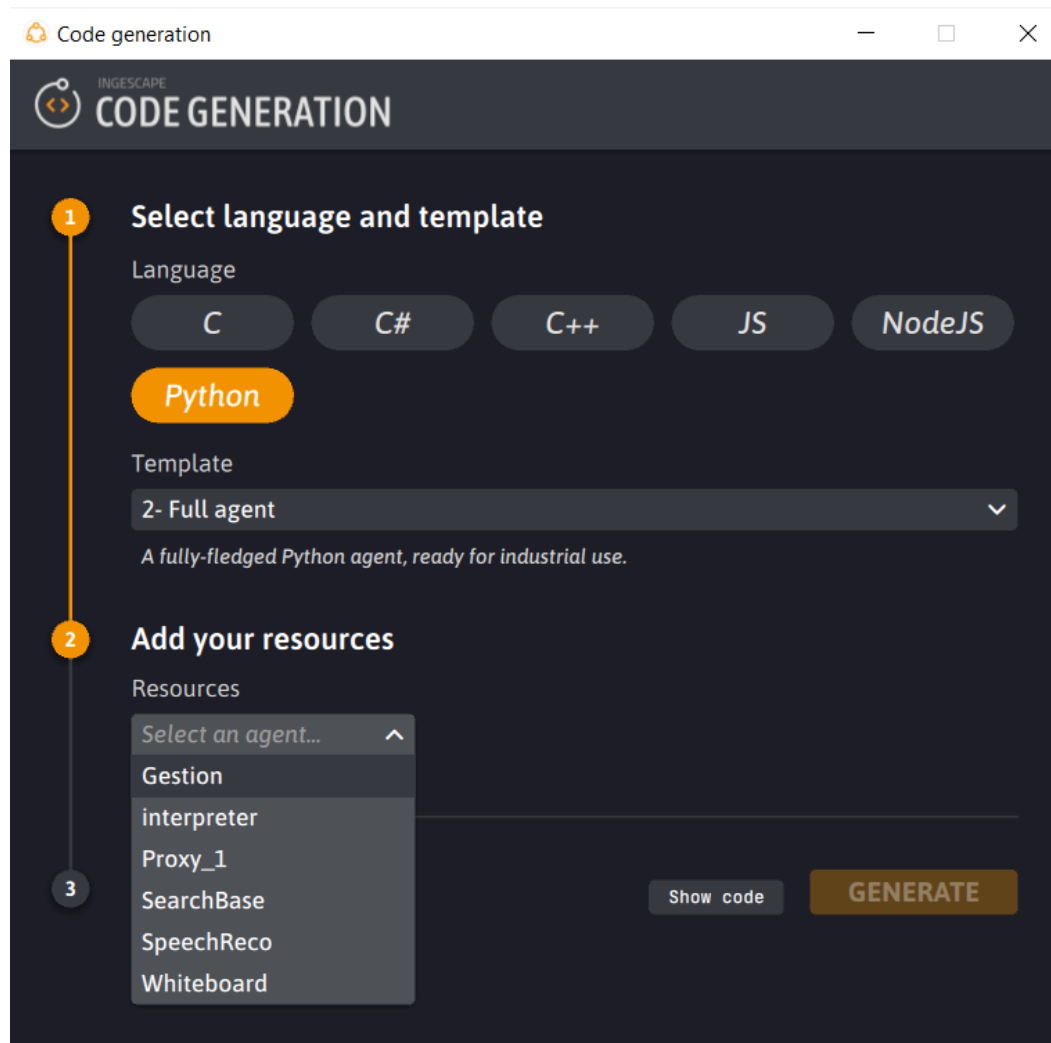
Dans un premier temps, nous avons utilisé « Ingescape Circle » dans le but de modéliser et d'avoir une représentation des agents que nous allons faire interagir entre eux. De plus, nous avons pu découvrir les services fournis par un agent déjà existant, « Whiteboard ».



Cette image permet de mettre en évidence les différents agents que nous avons utilisés, mais aussi leurs interactions.

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 :



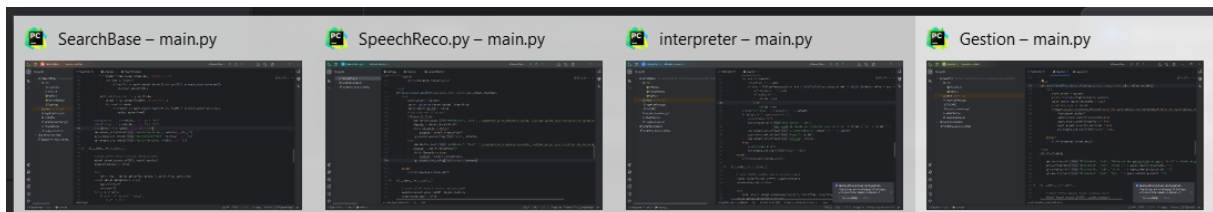
Dans notre cas, nous avons choisi de coder en python ; et sur conseil des intervenants, nous avons choisi le « full agent » comme template. Nous avons plus qu'à choisir les agents à générer.

```
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
```

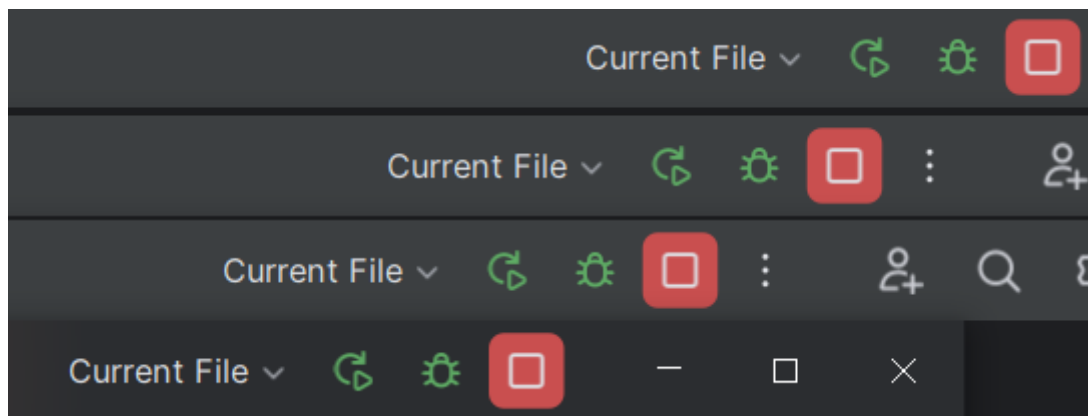
Une fois le code généré, il ne reste plus qu'à compléter les appels des callback « add code here », pour que le comportement espéré puisse être obtenu.

### 3 – Installation du projet

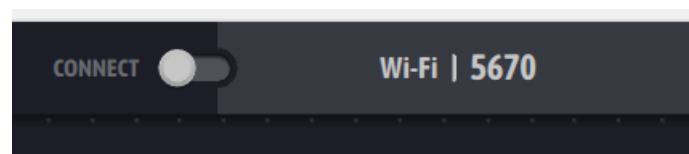
- Récupérer le répertoire du Whiteboard (fourni par ingénuité)
- Récupérer l'archive de notre projet « Ingescape.zip »
- Lancer « TravelBooking.igsplatform »
- Lancer l'application “Whiteboard.exe” présent dans le répertoire « Whiteboard », et l'ouvrir en plein écran
- Dans votre IDE, ouvrir chacun des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



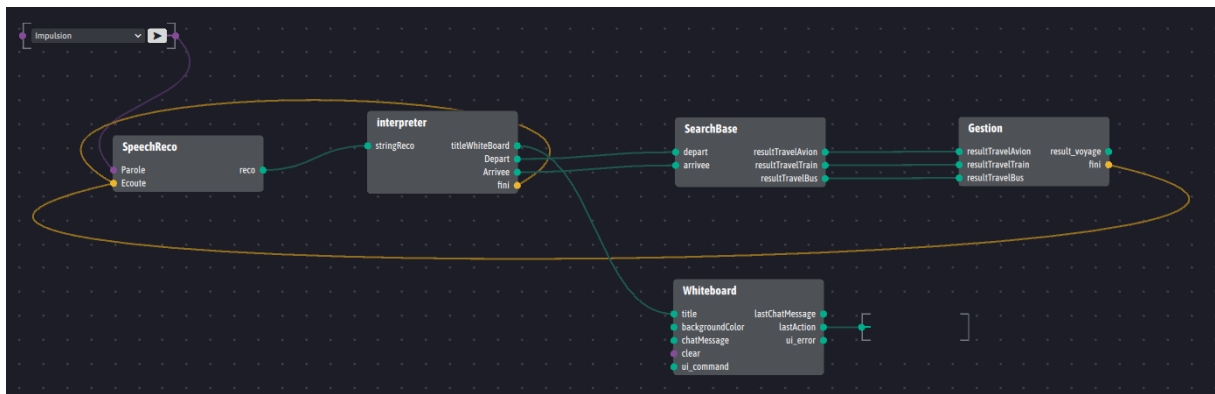
- Lancer les 4 main en même temps



- 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 :



## 4 – Mode d’emploi

- 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
  - Faire une requête orale
  - Pour la 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
  - Une fois que l’impulsion a été lancée la première fois, que vos requêtes suivantes soient valides ou invalides, le système est constamment à l’écoute et donc aucune action supplémentaire n’est requise pour faire de nouvelles demandes

## 4 - Résultats

### A - Demande valide

**Toulouse - Paris**

Item0 X

Item1 X

Toulouse

Item2 X

Item3 X

Paris

**interpreter**  
Vous venez de lancer une recherche pour aller de Toulouse à Paris

**Gestion**  
Recherche des possibilités en cours.  
Format du resultat :  
Ville de depart, ville arrivee, heure depart, heure arrivee, prix

**Gestion**  
Avion : [['Toulouse', 'Paris', '13:00', '14:15', '145.25']]

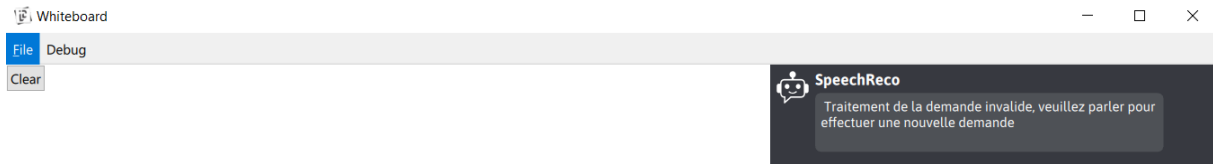
**Gestion**  
Train : [['Toulouse', 'Paris', '11:00', '16:00', '73.25']]

**Gestion**  
Bus : [['Toulouse', 'Paris', '08:00', '16:00', '55.75']]

**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.

## C - Tests Unitaire des agents

### A - Interpréter

Le but de cet agent est de pouvoir retrouver les noms de villes dans la string qui lui est passé en entrée.

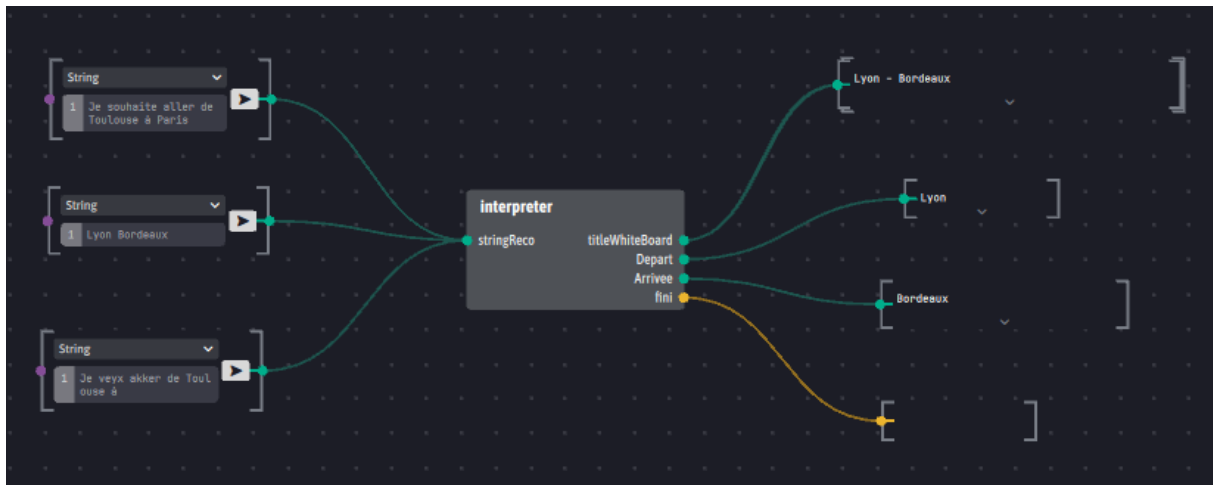


Dans cet exemple, en entrée nous avons utilisé la chaîne de caractères « Je souhaite aller de Toulouse à Paris ».

Cette formulation étant valide, l'agent devrait trouver la ville de départ « Toulouse », la ville d'arrivée « Paris » et retourner « Toulouse – Paris » qui sera envoyé au titre du Whiteboard.

La variable de sortie « fini » n'est pas utilisée en cas de formulation valide.

On voit donc que dans ce cas, l'agent réagit correctement d'après ce qu'on peut voir dans les observateurs.

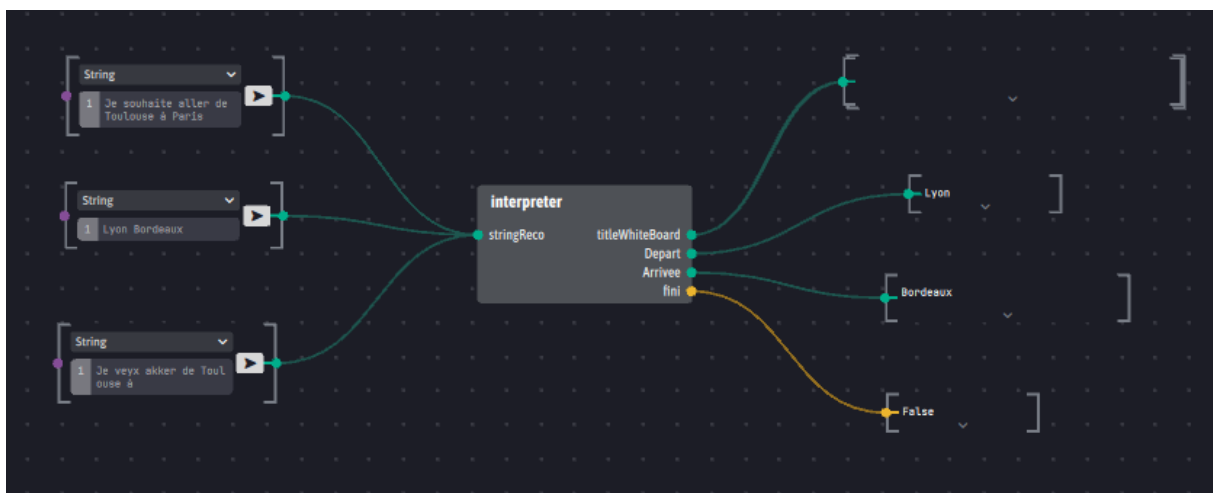


Dans cet exemple, en entrée nous avons utilisé la chaîne de caractères « Lyon Bordeaux ».

Cette formulation étant valide, l'agent devrait trouver la ville de départ « Lyon », la ville d'arrivée « Bordeaux » et retourner « Lyon – Bordeaux » qui sera envoyé au titre du Whiteboard.

La variable de sortie « fini » n'est pas utilisée en cas de formulation valide.

On voit donc que dans ce cas, l'agent réagit correctement d'après ce qu'on peut voir dans les observateurs.



Dans cet exemple, en entrée nous avons utilisé la chaîne de caractères « Je veux aller de Toulouse à ».

Cette formulation étant invalide (car il manque une ville), l'agent envoie rien sur les sorties car il n'a pas pu trouver les deux villes. De plus, dans ce cas le titre du whiteboard sera égale à « » car on n'a pas les deux villes.

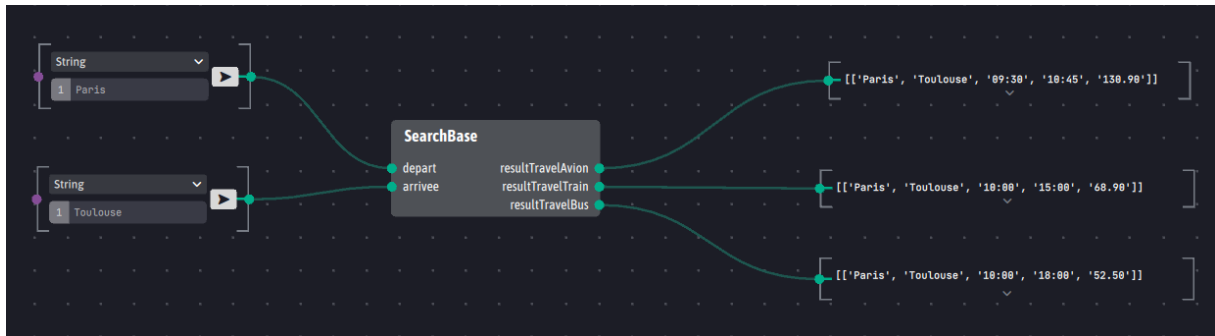
La variable de sortie « fini » est dans ce cas utilisée et mise à « False » car la formulation est invalide.

On voit donc que dans ce cas, l'agent réagit correctement d'après ce qu'on peut voir dans les observateurs.

PS : les deux villes n'étant pas détectés dans ce cas, l'observateur a conservé les villes de la phrase précédente « Lyon Bordeaux ».

## **B - SearchBase**

Le but de cet agent est de pouvoir trouver les différentes possibilités de voyage à présenter à l'utilisateur.

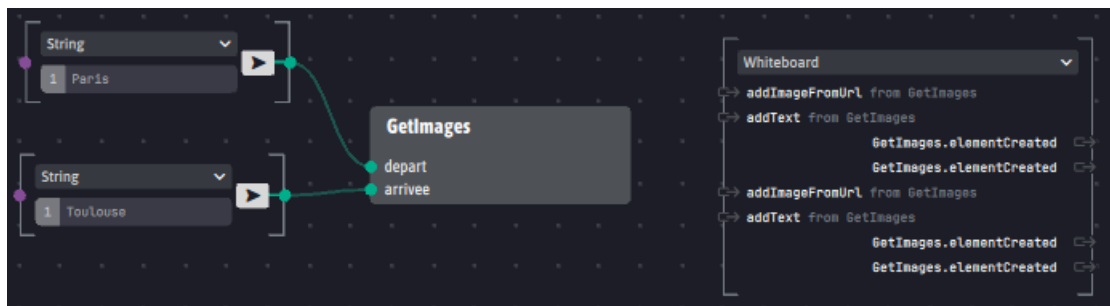


Dans cet exemple, en entrée nous avons utilisé 2 chaînes de caractères représentant les villes de départ et d'arrivée.

D'après les bases de données que nous avons, l'agent réagit correctement d'après ce qu'on peut voir dans les observateurs, car ce sont bien les options possibles que nous possédons.

## **C - GetImages**

Le but de cet agent est d'afficher sur le Whiteboard, une image pour chacune des villes, et le nom de la ville associé.



Dans cet exemple, en entrée nous avons utilisé 2 chaînes de caractères représentant les villes de départ et d'arrivée.

Nous utilisons un observateur de services car nous utilisons les services du Whiteboard. On voit bien que lorsqu'on envoie les deux string, pour chacune des villes on a un appel aux services « addImageFromUrl » et « addText ». L'agent réagit donc correctement d'après ce qu'on peut voir dans les observateurs