

TP 5 : Communication avec SPADE

Dr. Douae AHMADOUN

Objectif

Implémenter un **système de livraison multi-agents** en utilisant le framework **SPADE** (Smart Python Agent Development Environment).

Prérequis

Installation

```
pip install spade
```

Exécuter le code pour tester l'installation

```
python main.py
```

Le serveur XMPP est lancé automatiquement !

Instructions

1. Forker et cloner le repository
2. Installer SPADE : `pip install spade`
3. Compléter `exercices.py`
4. Tester
5. Prenez une capture du résultat pour la joindre au TD
6. Push votre solution

Fichiers

Fichier	Description
<code>exercices.py</code>	À COMPLÉTER
<code>main.py</code>	Pour tester votre code

Travail demandé

Partie 1 : LivreurAgent

Compléter la classe `RecevoirCFP` :

- Si `cfp` reçu : répondre avec `propose` (coût) ou `refuse`
- Si `accept-proposal` : envoyer `inform(done)`
- Si `reject-proposal` : afficher un message

Partie 2 : GestionnaireAgent

Compléter les 3 comportements :

- LancerAppelOffres : envoyer cfp à tous les livreurs
- CollecterPropositions : stocker les propositions reçues
- SelectionnerMeilleur : choisir le minimum et envoyer accept/reject

Partie 3 : Fonction main()

- Créer 3 livreurs et 1 gestionnaire
- Démarrer les agents
- Lancer une livraison vers (3, 4)

Rappels SPADE

Si besoin n'hésitez pas de consulter la documentation de spade sur <https://spade-mas.readthedocs.io/en/latest/>

Envoyer un message

```
msg = Message(to="destinataire@localhost")
msg.set_metadata("performative", "inform")
msg.body = "contenu"
await self.send(msg)
```

Recevoir un message

```
msg = await self.receive(timeout=5)
performative = msg.get_metadata("performative")
contenu = msg.body
```

Bon courage !