

## # Documentation du Projet Boxy2

### ## Table des matières

1. [Introduction] ([#introduction](#))
2. [Structure du Projet] ([#structure-du-projet](#))
3. [Composants Principaux] ([#composants-principaux](#))
4. [Configuration et Installation] ([#configuration-et-installation](#))
5. [Développement] ([#développement](#))
6. [Tests] ([#tests](#))
7. [Déploiement] ([#déploiement](#))
8. [Licence] ([#licence](#))

### ## Introduction

Boxy2 est un projet basé sur LangGraph, implémentant un chatbot intelligent avec une mémoire de conversation persistante. Le projet utilise des technologies modernes d'IA pour maintenir des conversations cohérentes sur plusieurs interactions.

### ## Structure du Projet

Le projet est organisé selon l'architecture suivante :

- ``/src`` : Code source principal
  - ``/agent`` : Logique du chatbot
    - ``graph.py`` : Définition du graphe de conversation
    - ``configuration.py`` : Configuration du système
- ``/app`` : Application web
- ``/tests`` : Tests unitaires et d'intégration
- ``/scripts`` : Scripts utilitaires
- ``/infra`` : Configuration d'infrastructure
- ``/rag`` : Composants de RAG (Retrieval-Augmented Generation)
- ``/utils`` : Utilitaires divers
- ``/swarm`` : Composants de swarm intelligence
- ``/static`` : Ressources statiques

### ## Composants Principaux

#### ### Agent de Chat

- Implémenté dans ``src/agent/graph.py``
- Gère la logique de conversation
- Maintient l'historique des conversations
- Génère des réponses contextuelles

### ### Configuration

- Gérée via ``pyproject.toml``
- Variables d'environnement dans ``env``
- Configuration Azure dans ``azure.yaml``

### ### Interface Utilisateur

- Interface graphique via LangGraph Studio
- Support pour le débogage et l'itération
- Intégration avec LangSmith pour le traçage

## ## Configuration et Installation

### 1. Prérequis :

- Python 3.x
- LangGraph Studio installé
- Clés API nécessaires

### 2. Installation :

```
``bash
# Copier le fichier d'environnement
cp .env.example .env

# Configurer les clés API dans .env
``
```

### 3. Configuration :

- Modifier le prompt système dans ``configuration.py``
- Choisir le modèle d'IA approprié
- Configurer les outils personnalisés si nécessaire

## ## Développement

### ### Bonnes Pratiques

- Utiliser le hot-reload pour le développement
- Tester les modifications dans LangGraph Studio
- Documenter les nouvelles fonctionnalités

### ### Extensions Possibles

- Ajout de nouveaux outils
- Intégration d'APIs externes
- Implémentation de logiques conditionnelles

- Personnalisation du comportement du chatbot

## ## Tests

Le projet inclut :

- Tests unitaires
- Tests d'intégration
- CI/CD via GitHub Actions

Pour exécuter les tests :

```
```bash
```

```
make test
```

```
```
```

## ## Déploiement

Le projet peut être déployé sur :

- Azure (configuration dans ``azure.yaml``)
- Autres plateformes cloud

## ## Licence

Le projet est sous licence MIT. Voir le fichier ``LICENSE`` pour plus de détails.

## ## Ressources Additionnelles

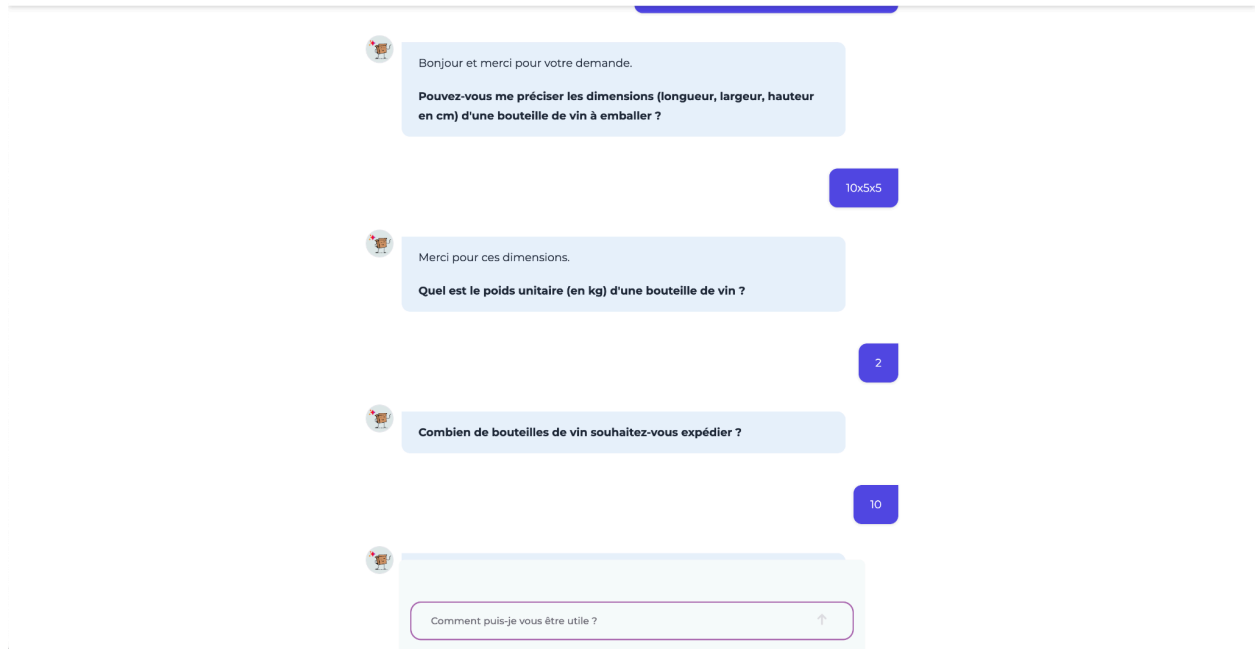
- [Documentation LangGraph] (<https://github.com/langchain-ai/langgraph>)
- [LangGraph Studio] (<https://github.com/langchain-ai/langgraph-studio>)
- [LangSmith] (<https://smith.langchain.com/>)

# DOCUMENTATION TECHNIQUE

Voici la page d'accueil de boxy:



Après avoir demandé ce qu'on voulait, il faut suivre le fil de la discussion et répondre aux questions :



Une fois le lot de question répondu, un dernier message nous est envoyé pour nous donner toute les informations ainsi que les produits :



Pour expédier des bouteilles de vin, il est crucial d'utiliser des caisses spécifiques avec croisillons renforcés, surtout pour assurer la sécurité et la conformité lors du transport par transporteur. Pour garantir l'intégrité des bouteilles et faciliter la manutention, optez pour des caisses adaptées avec calage intégré.

Je commence par sélectionner une caisse spécialement conçue pour 5 bouteilles de vin avec croisillons renforcés, conforme au guide.

### Étape 1 : Utiliser une caisse carton d'expédition spécifique bouteilles avec croisillons renforcés

- Pourquoi : Cette caisse double cannelure, conçue pour 5 bouteilles, offre un calage optimal, maintien renforcé, et répond aux exigences de sécurité pour le transporteur.
- **Lien produit** : [Caisse croisillons magnum vin ou champagne](#)

### Étape 2 : Fermer la caisse avec un adhésif adapté

- Pourquoi : Un ruban adhésif en papier kraft est conseillé pour la double cannelure et le poids (jusqu'à 20 kg). Il sécurise la fermeture pendant le transport.
- **Lien produit** : [Pack ruban adhésif kraft + dévidoir](#)

### Étape 3 : Apposer les étiquettes d'expédition et de signalisation

- Pourquoi : Pour signaler la fragilité et le sens de manipulation des caisses, et répondre à l'obligation d'étiquetage en expédition.
- **Étiquette d'expédition** : [Étiquette d'expédition 120x80 mm](#)
- **Étiquette FRAGILE** : [Étiquette fragile 140x40 mm](#)
- **Étiquette HAUT** : [Étiquette haut 165x55 mm](#)

**Cette solution apporte une protection renforcée, un calage optimal et garantit une expédition conforme aux contraintes de sécurité et de fragilité.**

Pour toute question ou besoin d'adaptation (palettisation, nombre de caisses...), n'hésitez pas à préciser !

Recommandations (5)

Caisses



Caisse carton  
d'expédition pour...

À partir de

**2,76 € HT**

le colis



Adhésifs



Pack mini-rolls ruban  
adhésif en papier kraf...

À partir de

**21,45 € HT**

le pack de 6



Étiquette



Étiquette d'expédition  
en rouleau 120x90 mm

À partir de

**47,05 € HT**

le colis de 1000



Étiquette



Étiquettes d'expédition  
Fragile

À partir de

**37,10 € HT**

le colis de 2



Étiquette



Étiquette d'expédition  
Haut 165x55 mm

À partir de

**42,60 € HT**

le colis de 2



Comment puis-je vous être utile ?



Je suis une intelligence artificielle en constante recherche de perfectionnement. Actuellement en phase de test, j'apprends et m'améliore chaque jour grâce à vos demandes et vos questions. En cas de besoin nos conseillers sont à votre écoute au 01 49 90 49 90 ou par chat.