

# Projet FastAPI & Chabot - Classification de Tickets

## Problématique

Les entreprises reçoivent quotidiennement des tickets de demande ou d'incident de la part de leurs employés. Traiter manuellement ces tickets est coûteux et chronophage, surtout pour trier et orienter vers le bon service (IT, RH, Finance, etc.). De plus, l'interaction avec un système technique (API) peut être une barrière pour les utilisateurs non techniques.

## Apport Business

Ce projet automatise deux aspects clés :

- Le **routing intelligent des tickets** vers le bon département via un modèle de classification.
- L'**interface conversationnelle** simplifie l'usage des outils techniques (API CRUD) en langage naturel. Cela réduit les coûts de traitement, augmente la réactivité des services, et améliore l'expérience utilisateur.

## Approche technique

Deux modules principaux :

- **API FastAPI** avec endpoints CRD (Create, Read, Delete) pour gérer les tickets. Chaque nouveau ticket est automatiquement labellisé via un modèle ML (classification de texte).
- **Chatbot en console** qui comprend les intentions utilisateur (create, read, delete) via un modèle NLP (TF-IDF + régression logistique) et tape sur l'API. Données persistées dans la base de données SQLite. Modèles sauvegardés en fichiers Pickle (.pkl).

## Choix techniques

- **FastAPI** : framework rapide et moderne pour API REST, compatible avec Pydantic pour validation.
- **Pickle** : pour sérialiser les modèles ML. Projet FastAPI & Chatbot - Classification de Tickets
- **TF-IDF + Régression Logistique** : méthode simple, rapide, efficace pour classer les intentions.
- **SQLite** : base de données légère et facile à implémenter

## Résultats obtenus

- Le Chatbot comprend bien les intentions comme "je veux ajouter un ticket", "affiche les tickets", ou "supprime le ticket 123".
- L'API crée des tickets automatiquement labellisés avec un taux de précision correct sur des jeux d'essai.
- Le prototype est fonctionnel, facilement extensible et améliorable avec une base de données ou une UI.

## Architecture

