

# TP - Module 2

Objectif global : développer un module d'intégration API de qualité professionnelle, en utilisant un LLM comme assistant et en fournissant tous les prompts utilisés.

---

## Mission 1 - Génération et structuration du projet

Durée : 1h

Objectif : Créer l'architecture du module.

### Consignes

1. Générer un squelette de projet via un LLM (Python ou Node.js).
2. Imposer dans le prompt :
  - architecture en dossiers (services/, utils/, config/),
  - gestion des erreurs,
  - fonctions d'appel API séparées,
  - utilisation d'un fichier .env.
3. Vérifier, corriger et nettoyer le code généré.
4. Documenter les prompts utilisés dans /prompts/squelette.md.

### Livrable

Arborescence + squelette propre.

---

## Mission 2 - Implémentation des fonctions d'intégration

Durée : 1h30

Objectif : Développer les appels API essentiels.

### Consignes

Créer les fonctions permettant de récupérer au minimum :

- wind\_speed\_10m
- wind\_gusts\_10m
- pressure\_msl

Pour chaque fonction :

- URL dynamique,
- paramètres météo,
- validation des paramètres (lat/long),

- gestion des erreurs 4xx/5xx,
- parsing des réponses.

Documenter les prompts → /prompts/appels\_api.md.

## Livrable

weatherService.js ou weather\_service.py fonctionnel.

---

# Mission 3 - Sécurisation, robustesse et refactorisation

Durée : 1h

Objectif : Professionnaliser le module.

## Consignes

- Ajouter un timeout configurable.
  - Protéger les variables dans .env.
  - Ajouter des logs simples.
  - Refactoriser avec l'aide d'un LLM (mais avec validation humaine).
- Documenter → /prompts/refacto.md.

## Livrable

Version robuste et sécurisée du module.

---

# Mission 4 - Tests d'intégration

Durée : 1h

Objectif : Vérifier la fiabilité du module.

## Tests à implémenter

- vent 10m renvoyé correctement,
- pression atmosphérique présente,
- gestion des paramètres invalides,
- gestion du timeout.

Documenter → /prompts/tests.md.

## Livrable

Dossier /tests + capture des tests passants.

---

## Mission 5 - Documentation et schéma technique

Durée : 45 min

Objectif : Préparer le module pour l'équipe.

### Contenu attendu

- README (installation + usage + exemples d'appels).
- Schéma Mermaid/Draw.io du module.
- Section "Comment j'ai utilisé le LLM et pourquoi".  
Documenter → /prompts/documentation.md.

### Livrable

README complet + schéma.

---

## Mission 6 - Débogage & revue de code

Durée : 45 min

Objectif : Finaliser un rendu propre.

### Activités

- Débogage assisté par LLM (contrôlé).
- Revue croisée entre apprenants.
- Corrections finales.  
Documenter → /prompts/debug.md.

### Livrable

Version finale propre + prompts.

---

## Livrables

- Module d'intégration opérationnel
- Tests d'intégration
- README complet
- Schéma d'architecture
- Captures Postman
- Dossier prompts complet
- Revue de code finale