

# Manuel de Test - API Flask

---

## Lancement des Scripts de Test

### Étape 1 : Configuration (optionnel)tch

cd ift6758\backend\test call 1-set\_env.batConfigure les variables d'environnement (WANDB\_API\_KEY, FLASK\_LOG, PORT)

### Étape 2 : Démarrer le serveur

call 2-start\_server.batLe serveur démarre sur <http://localhost:5000>. **Garder cette fenêtre ouverte.**

### Étape 3 : Lancer les tests

Dans une **nouvelle fenêtre** : cd ift6758\backend\test call 3-test\_api.bat Le script exécute automatiquement tous les tests et fait une pause entre chaque test.

---

## Cas de Test Exécutés par `test_api.bat`

### Test 1/5 : GET /logs

Vérifie que l'endpoint de consultation des logs fonctionne.

- **Commande** : GET <http://localhost:5000/logs>
- **Résultat attendu** : HTTP 200 avec contenu du fichier de log

### Test 2/5 : POST /download\_registry\_model (modèle valide)

Télécharge le modèle `logreg-distance` version `v0` depuis Wandb.

- **Commande** : POST `/download_registry_model` avec `{"model": "logreg-distance", "version": "v0"}`
- **Résultat attendu** : HTTP 200 avec `status: "success"` ou `"already_loaded"`

### Test 3/5 : POST /download\_registry\_model (modèle invalide)

Teste la gestion d'erreur pour un modèle non supporté.

- **Commande** : POST `/download_registry_model` avec `{"model": "invalid-model"}`
- **Résultat attendu** : HTTP 400 avec message d'erreur

### Test 4/5 : POST /predict (sans modèle chargé)

Vérifie que l'API refuse les prédictions si aucun modèle n'est chargé.

- **Commande** : POST `/predict` avec `{"distance_net": 10.5}`
- **Résultat attendu** : HTTP 400 avec message "Model not loaded"

## Test 5/5 : Flux complet - Download puis Predict

### Étape 1 : Télécharge `logreg-distance`

- **Résultat attendu** : Modèle chargé avec succès

### Étape 2 : Effectue une prédiction avec `distance_net: 10.5`

- **Résultat attendu** : HTTP 200 avec `prediction` (valeur entre 0 et 1)

**Test supplémentaire** : Teste aussi `logreg-distance-angle` avec `distance_net` et `angle_net`

---

## Notes

- Les tests s'exécutent dans l'ordre et font une pause entre chaque test
- Appuyer sur une touche pour passer au test suivant
- Le serveur doit rester démarré pendant l'exécution des tests
- En cas d'erreur, vérifier les logs avec `GET /logs` ou
- le fichier `flask.log` crée dans le repertoire backend