

Travaux Pratiques 1 : Développement d'une API RESTful

Objectifs pédagogiques

- Comprendre les principes fondamentaux de l'architecture REST.
- Appliquer les verbes HTTP (GET, POST, PUT, DELETE) dans le contexte d'une API.
- Utiliser un framework web pour faciliter le développement d'une API REST.
- Tester et documenter une API REST.

Sujet : Gestion d'une bibliothèque

Contexte : Développer une API RESTful pour gérer une bibliothèque simple. L'API doit permettre de gérer les utilisateurs, les livres et les emprunts.

Fonctionnalités à implémenter

- **Utilisateurs :**
 - Créer un utilisateur
 - Récupérer un utilisateur par ID
 - Lister tous les utilisateurs
 - Modifier les informations d'un utilisateur
 - Supprimer un utilisateur
- **Livres :**
 - Créer un livre
 - Récupérer un livre par ID
 - Lister tous les livres
 - Modifier les informations d'un livre
 - Supprimer un livre
- **Emprunts :**
 - Emprunter un livre
 - Renvoyer un livre
 - Lister les emprunts d'un utilisateur
 - Lister les livres empruntés

Environnement de développement

- Langage de programmation : Python, Java, JavaScript (Node.js), PHP ou autre selon les compétences des étudiants.
- Framework web : Django (Python), Spring Boot (Java), Express.js (Node.js), Laravel (PHP) ou autres.

- Base de données : SQLite, PostgreSQL, MySQL ou une autre base de données relationnelle.
- Outils de test : Postman, cURL ou un autre outil de test HTTP.

Déroulement du TP

1. Conception de l'API :

- Définir les ressources et les opérations CRUD associées.
- Définir les formats de données (JSON ou XML).
- Créer un document de spécification de l'API (Swagger, OpenAPI).

2. Implémentation de l'API :

- Créer le modèle de données (entités) correspondant aux ressources.
- Développer les contrôleurs REST pour gérer les requêtes HTTP.
- Implémenter la logique métier pour les opérations CRUD.

3. Tests de l'API :

- Utiliser Postman ou cURL pour envoyer des requêtes HTTP.
- Vérifier les réponses et les codes de statut HTTP.
- Tester différents scénarios, y compris les erreurs.

4. Documentation de l'API :

- Générer la documentation à partir du fichier de spécification (Swagger UI, etc.).
- Ajouter des commentaires explicatifs à l'API.