

API Rest en Python avec FastAPI

L'objectif de cette formation est d'initier les participants à la création d'API Restful en utilisant FastAPI, un framework moderne et performant de Python. Les stagiaires apprendront à concevoir, développer et déployer des API robustes tout en intégrant des bonnes pratiques de développement. À l'issue de la formation, ils seront en mesure de créer des applications web évolutives.

Pré-requis : Bases en langage Python, concepts fondamentaux des API, expérience avec le développement web.



Modalité :

- Distanciel en classe virtuelle
- E-learning : à venir
- Présentiel

Communauté:

community.reconvert.net

Durée totale :

21 H (3 jours)

Authentification et autorisation

Concepts d'authentification et d'autorisation
Utilisation de JWT (JSON Web Tokens)
Mise en place de routes protégées
Gestion des rôles et permissions
Exemples d'implémentation d'authentification
Sécurisation des mots de passe
Bonnes pratiques de sécurité
Travaux pratiques : ajouter un système d'authentification

Tests et débogage

Importance des tests, Introduction à pytest
Écriture de tests unitaires pour les routes
Tests d'intégration avec FastAPI
Outils de débogage recommandés
Gestion des erreurs et des exceptions
Exemples de tests pour une API
Travaux pratiques : écrire des tests pour les routes de l'appli

PLAN DETAILLE

Introduction à FastAPI

Présentation de FastAPI et ses spécificités
Installation et configuration de l'environnement
Structure d'un projet FastAPI
Concepts des API Restful
Comparaison avec d'autres frameworks (Flask, Django)
Mise en place d'un serveur local
Exemples d'applications utilisant FastAPI
Travaux pratiques : créer un projet FastAPI en local

Création des routes

Définition des routes dans FastAPI
Méthodes HTTP : GET, POST, PUT, DELETE
Gestion des paramètres de requête
Routes dynamiques et statiques
Utilisation des dépendances
Gestion des réponses JSON
Documentation automatique avec Swagger
Travaux pratiques : créer des routes

Modélisation des données

Introduction aux modèles de données avec Pydantic
Validation des données d'entrée
Sérialisation et désérialisation des données
Gestion des erreurs de validation
Utilisation de modèles imbriqués
Intégration avec des bases de données
Exemples de modèles de données
Travaux pratiques : créer des modèles

Gestion des bases de données

Introduction à SQLAlchemy
Configuration de la base de données
Création de tables et migrations
Opérations CRUD avec SQLAlchemy
Relations entre tables
Utilisation d'ORM avec FastAPI
Connexion à différentes bases de données
Travaux pratiques : mettre en place une base de données