

## Formation Python Avancé : FastAPI

<b>Durée :</b>	3 jours
<b>Public :</b>	Développeurs Python
<b>Pré-requis :</b>	Avoir suivi le stage "Python : Initiation + Approfondissement" ou notions équivalentes
<b>Objectifs :</b>	Construire une API REST en Python avec FastAPI
<b>Sanction :</b>	Attestation de fin de stage mentionnant le résultat des acquis
<b>Taux de retour à l'emploi:</b>	Aucune donnée disponible
<b>Référence:</b>	PYT101930-F
<b>Note de satisfaction des participants:</b>	Pas de données disponibles

### Découvrir FastAPI

Présentation des Web Services (WS) : fonctionnement, intérêt, interopérabilité  
Architecture orientée services (SOA) vs microservices : composantes, technologies  
FastAPI : présentation, cas d'usage, architecture  
FastAPI vs autres frameworks (Flask, Django)  
Design et documentation : OpenApi Specification (Swagger)  
Outils de test de services web : Postman

**Atelier : Installation de l'environnement de développement (VS Code + Interpréteur Python) -  
Création d'un projet FastAPI (structure, point d'entrée, dépendances)**

### Implémenter et interroger des services web REST

Architecture REST : composantes, méthodes d'appel (GET, POST, PUT, DELETE)  
Définition de routes  
Gestion des paramètres de la requête  
Validation des entrées : typing, pydantic  
Types de réponses, format (json, xml, texte, binaire)  
Gestion des erreurs  
Traitements asynchrones  
Déploiement d'un service RESTful  
Interrogation de web services REST (Python/Javascript)  
Implémentation de tests unitaires et fonctionnels (TestClient, PyTest)  
Déploiement et configuration d'une application FastAPI sur un serveur (Uvicorn, Hypercorn)

**Atelier : Création et interrogation d'une API REST avec FastAPI**

### Sécuriser une application FastAPI

Niveaux de sécurité

Gestion de l'authentification dans un web service (JWT, OpenID Connect)

Gestion des droits (OAuth2)

Multiples configurations : CORS, HTTPS, ...

**Atelier : sécurisation globale de l'application FastAPI**