

Formation Python Avancé : Web Services + ORM

Durée :	5 jours
Public :	Développeurs Python
Pré-requis :	Avoir suivi le stage "Python : Initiation + Approfondissement" ou notions équivalentes
Objectifs :	Construire une API en Python et implémenter une couche d'accès aux données avec un ORM
Sanction :	Attestation de fin de stage mentionnant le résultat des acquis
Taux de retour à l'emploi:	Aucune donnée disponible
Référence:	PYT100613-F
Note de satisfaction des participants:	Pas de données disponibles

Comprendre le besoin

- Présentation des Web Services (WS) : fonctionnement, intérêt, interopérabilité
- Architecture orientée service (SOA) : composantes, technologies
- Technologies : Architecture REST, Websockets
- Plates-formes à services web
- Choix de l'implémentation
- Design et documentation : Swagger
- Outils de test de services web

Atelier : Installation de l'environnement de développement et d'exécution, choix d'une implémentation de WebService, description d'un service

Implémenter et interroger des services web REST

- Architecture REST : composantes, méthodes d'appel
- Flask ou Django : présentation des outils pour des services REST
- Envoi de paramètres, validations
- Type de réponse, format
- Gestion des erreurs
- Déploiement d'un service RESTful
- Interrogation de web services REST (Python/Javascript)

Atelier : Création et interrogation d'une API REST

Sécuriser un service web

- Niveaux de sécurité
- Gestion de l'authentification dans un web service
- Gestion des droits
- Cryptographie au niveau des messages
- Signatures de messages
- Standards de sécurité disponibles

Atelier : sécurisation globale de l'API

Réaliser un mapping relationnel objet (ORM)

- Pattern DAO (Data Access Object)
- Frameworks ORM : fonctionnalités, intérêt
- ORMs Python : SQLAlchemy, Django ORM, PonyORM, SQLAlchemy, Peewee, ...
- Mapping des tables et gestion des clés primaires (simples, composées)
- Mapping des types de bases, propriétés des colonnes
- Gestion de la concurrence : optimistic (versioning), pessimistic
- Gestion des relations : OneToMany/ManyToOne, OneToOne, ManyToMany
- Paramétrage des cascades
- Gestion des collections
- Mapping de l'héritage
- Stratégies de chargement : Lazy ou Eager

Atelier : Réalisation d'un schéma global de mapping d'une base de données

Ecrire des requêtes avec un ORM

- Langage de requêtes objet
- Sélections de base, filtres
- Jointures complexes
- Fonctions d'agrégation, de chaîne, ...
- Gestion des chargements Lazy/Eager

Atelier : Réalisation d'opérations CRUD (Create Read Update Delete) - requêtes complexes

Découvrir des fonctionnalités avancées

- Cycle de vie des entités et validation
- Intercepteurs, Event-listeners
- Configuration avancée : performance et fonctionnalités
- Utilisation du cache
- Serveurs Websockets en Python

Atelier : Implémentation d'intercepteurs et gestion du cache.

Atelier : Implémentation d'un salon de discussion utilisant des websockets.