

Python, développement web avec Django

4 j (28 heures)

Ref : PYT-DJ

Public

Développeurs Web

Pré-requis

Connaissances des bases du langage Python et technologies Web (HTML, JavaScript...)

Moyens pédagogiques

Formation réalisée en présentiel ou à distance selon la formule retenue
Exposés, cas pratiques, synthèse, assistance post-formation pendant trois mois Vidéoprojecteur, support de cours fourni à chaque stagiaire

Modalités de suivi et d'évaluation

Feuille de présence émargée par demi-journée par les stagiaires et le formateur
Exercices de mise en pratique ou quiz de connaissances tout au long de la formation permettant de mesurer la progression des stagiaires
Questionnaire d'évaluation de la satisfaction en fin de stage
Auto-évaluation des acquis de la formation par les stagiaires
Attestation de fin de formation

Objectifs

- Savoir interfacier Django avec une base de données
- Développer des vues dynamiques
- Développer des templates
- Gérer les formulaires Django
- Mettre en oeuvre les principales fonctions avancées

Programme détaillé

INTRODUCTION PYTHON ET DJANGO

- Programmer en Python
- Historique de Django
- Traitement d'une requête HTTP
- Structure d'un projet Django

Fichiers

settings.py

Pattern MVT (Model View Template)

Naviguer dans la documentation de Django

Gérer un projet Django en CLI

URL ET VUES

Routing des requêtes avec urls.py

Pattern matching dans les URL

Isolation des routes par application Django

Vues et réponse HTTP

Objets "requête / réponse"

Templates

CHARGER UN TEMPLATE DANS UNE VUE

Tags et filtres

Héritage de template

Ressource statique dans un template

Définir un tag / filtre personnalisé

BASE DE DONNEES ET MODELES

Découvrir l'ORM (Object Relational Mapper) Django

Définir un modèle

Définir les relations entre modèles (OneToOne, ForeignKey, ManyToMany)

Gérer les migrations de la BDD (database)

CRUD (Create, Read, Update, Delete) sur les modèles

REQUETES SQL ET QUERYSET

Fonctionnement des QuerySet

Faire des requêtes QuerySet en utilisant le REPL

Requêtes complexes et relations inversées

LIER SES MODELES AVEC LES VUES

Charger un modèle dans une vue

Paramétrer une requête à partir d'une vue

Validation d'un modèle

Signaux et fonctions d'un modèle

OPERATIONS AVANCEES SUR LES MODELES

Requêtes en fonction avec le manager personnalisé

- Ecrire ses requêtes SQL à la main
- Faire des requêtes avancées
- Gérer plusieurs BDD (databases)
- Requêtes et transactions

GERER LES FORMULAIRES

- Afficher et traiter un formulaire Django
- Connecter un formulaire à un modèle
- Vérifier et valider le contenu d'un formulaire
- Ressources dans un formulaire
- Gérer plusieurs formulaires avec un Formset

ADMINISTRER DJANGO

- Installer le site d'administrateur
- Personnaliser les listes et formulaires
- Filtrer, trier, chercher avec l'administrateur
- Ajouter des actions personnalisées
- Ajouter du code JavaScript
- Gérer les utilisateurs et les droits

STRUCTURER SES VUES EN CLASSES

- Principe de "Class View"
- Requêtes HTTP, templates, variables de contexte
- Afficher un modèle dans une Class View
- Gérer un formulaire dans une Class View
- Les mixins et les Class View

API REST AVEC DJANGO REST FRAMEWORK

- Principes d'une API REST
- Définir un endpoint REST avec DRF
- Fonctionnement d'un routeur
- Objet nested dans une API REST
- Sérialisation de modèles et rendu format JSON / XML / MessagePack
- Routing d'actions supplémentaires

NOTIONS AVANCEES REST

- Valider le contenu
- Gérer les versions d'une API
- Caching et ETag
- Authentification et permissions d'un client

TESTS AVEC DJANGO ET DRF

Mise en place de tests Python avec les TestCases
Tester avec un jeu de données en BDD
Client de test HTTP
Tester une API REST avec DRF
Tests unitaires, de transactions, avec un serveur Web

AVOIR UN SITE EN MULTILINGUE

Préparer le code pour l'internationalisation
Générer les traductions
Activer différentes langues dans Django
Traductions en JavaScript

CONFIGURER UNE APPLICATION EN PRODUCTION

Multi-configuration de base de données, flags
Provisionner une machine de développement avec Vagrant
Workflow avec Nginx et Gunicorn
Automatisation du déploiement avec Fabric et Ansible
Isoler son application avec Docker
