Dr Samba SIDIBE
Ecole Polytechnique de Thies
***ssidibe@ept.sn*, mail@sambasidibe.com**

Développement Web avec Django

Route, view et template

Draft

Plan

* Introduction
* Installation de Django
* Structure d'un projet Django
* Les routes et view
* Routing avancée
* Les objets HttpRequest et HttpResponse
* Les Templates

Introduction

Qu'est ce que Django?

* Un Framework python pour le développement d'applications web
* Il a été créé en 2003 pour le journal Lawrence
* Publie en 2005 sous license BSD
* Il a pour but de facilite le développement d'application Web
* Son nom fait hommage au Guitariste Django Reinhardt

Introduction

Pourquoi Django?

* **Rapide et simple**
  Le but principal de Django est d'aider les développeurs a produire des applications le plus rapidement possible

* **Plusieurs extensions natives**
  Des dizaines d'extensions qu'on peut utiliser pour gérer l'authentification des utilisateurs, l'administration de contenu, des flux RSS, etc.

Introduction

Pourquoi Django?

* **Grande sécurité**
  Aide a éviter de nombreuses erreurs courantes telles que l'injection SQL, les scripts intersites, la falsification de requêtes intersites et le détournement de clics

  Il fournit un moyen sécurisé de gérer les comptes d'utilisateurs et les mots de passe.

Introduction

Pourquoi Django?

* **Extrêmement évolutif**
  S'adapte rapidement et de manière flexible pour répondre aux exigences du développement web
* Il supporte les trafics les plus lourds
* Il est utilise par dans des domaines très varies voir une liste non exhaustive sur le lien https://djangosites.org/

Installation de Django

Pour installer Django il est conseillé d'utiliser pip et virtualenv

* pip : est un gestionnaire de paquets Python, facilite l'installation de librairies Python
* virtualenv : un outil permettant d'isoler un environnement Python pour un projet spécifique.

Installation de Django

```
# Créer l'environnement virtuel
virtualenv mysite -p python3
#si la commande virtualenv n'existe pas, l'installer avec la commande « pip install virtualenv »

#entrer dans le dossier de l'env virtuel
cd mysite

#activer l'env virtuel
source bin/activate

#installer Django dans l'env virtuel
pip install Django==2.2.7
```

Installation de Django

Créer un projet Django

```
# Créer un projet Django
django-admin startproject nom-projet .

# Demarrer le server
python manage.py runserver

# ou en spécifiant l'IP et ou le port
python manage.py runserver ip:port
```

Installation de Django

Créer une application Django

* Un projet est constitué d'un ou de plusieurs applications
* Les applications doivent etre indépendantes entre elles
* Une application peut etre considérée comme un module indépendant qu'on peut déplacer facilement d'un projet a l'autre

```
#Commande pour créer une application Django
python manage.py startapp dic1
```

Structure d'une application

* apps.py : configuration du projet
* views.py : contient des fonctions permettant de répondre aux requêtes Http
* models.py : modele de données
* admin.py : configure interface de gestion des modèles

Routing

* Le routing permet de faire la correspondance entre les URLs et les fonctions python qui doivent les traiter

* Les routes sont déclarées dans le fichier urls.py

Fonction pour traiter une requête

* Les fonctions pour traiter les requêtes sont en general déclarées dans views.py

* Elle prend en paramètre un Objet HttpRequest et peut prendre d'autres paramètres

* Elle doit retourner un Objet HttpResponse

Fonction pour traiter une requête

* Exemple :

```
from django.http import HttpResponse
# Create your views here.


def bonjour(request):
    return HttpResponse("Bonjour")
```

Fonction pour traiter une requête

* Ajout de la fonction dans urls.py

```
from django.contrib import admin
from django.urls import path
from dic1 import views

urlpatterns = [
    path('admin/', admin.site.urls),
    path('toto/', views.bonjour),
]
```

Fonction pour traiter une requête

Paramètres dans l'url de la requête

* L'url peut contenir des variables que Django donnera en paramètres a la fonction chargée de traiter la requête HTTP

Fonction pour traiter une requête

Paramètres dans l'url de la requête

* Exemple de routing

```
urlpatterns = [
    path('admin/', admin.site.urls),
    path('eleve/<int:num>/<str:prenom>/<str:nom>/<int:age>', views.afficheEleve),
]
```

* Fonction associée

```
def afficheEleve(request,num, prenom, nom, age):
 res=HttpResponse('Eleve numero {}, prenom {}, nom {} et age {}'.format(num,prenom,nom,age))
 return res
```

Fonction pour traiter une requête

Utilisation des expressions régulières

* La fonction re_path permet d'utiliser des expressions régulières pour faire le routing

* Les elements capturés de l'expression régulière seront ajoutés au paramètres de la fonction qui doit traiter la requête

Fonction pour traiter une requête

Utilisation des expressions régulières

* Exemple de routing avec regex

```
urlpatterns = [
  path('admin/', admin.site.urls),
  re_path(r'^eleve/(20[\d]{2})(GIT|GEM|GC)([\d]{4})/$',views.helloRegex),
]
```

* Fonction associée

```
def helloRegex(request, annee, dep, num):
 return HttpResponse("Annee={}<br/>dept={}<br/>num={}<br/>".format(annee, dep, num))
```