heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

Django
Un python sur la toile

Reynald Borer & Murielle Savary

Tables des matières

I. Introduction

- Django est un framework web écrit en Python.
- Développé en 2003 à la base pour un petit journal du Kansas : Lawrence.
- En 2005, l'équipe de développement décide de rendre le framework open source.
- Bien que ce framework bénéficie de contributeurs à travers le monde, les développeurs originaux continuent de fournir une direction pour le développement du framework.
- Origine du nom en hommage à Django Reinhardt.

2. Rappels Python

- Langage interprété, orienté objet.
- Typage dynamique fort.
- Gestion automatique de la mémoire.
- Multi-plateforme.
- Syntaxe lisible : pas de caractère de fin de ligne, l'indentation identifie les blocs, mots-clés anglais.
- Possède plusieurs types de données natifs (list, tuple, dict, str, int, long, ...).
- Terminologie : module, classe, méthode.

2. Rappels Python


```
int factorielle (int x) {
    if (x == 0)
        return 1;
    else
        return x * factorielle(x-1);
}
```

```
def factorielle(x) :
    if x == 0:
        return 1
    else :
        return x * factorielle(x-1)
```

3. Présentation de Django

A. Notion de projet

- Projet : instance d'une ou plusieurs applications avec une configuration associée.
- Application : ensemble portable de fonctionnalités Django. Contient généralement les vues et les modèles. Une application peut être utilisée dans plus d'un projet.

3. Présentation de Django

B. Modèle de conception MVC <=> MTV

- Noyau : gère l'intégration du serveur web, les urls, les erreurs, la configuration et les "middlewares".
- Modèle : traite la logique de domaine.
- Vue : ne représente pas forcément la façon dont les données sont affichées mais quelles données sont affichées. Contient les fonctions s'occupant de la logique applicative.
- Template : gère la logique de présentation.

3. Présentation de Django

B. Modèle de conception MVC <=> MTV

<table>
<tr><th>MVC</th><th>MTV</th></tr>
<tr><td colspan="2">Modèle</td></tr>
<tr><td rowspan="2">Vue</td><td>Template</td></tr>
<tr><td rowspan="2">Vue</td></tr>
<tr><td>Contrôleur</td><td>Django</td></tr>
</table>

3. Présentation de Django

C. Philosophie du framework

- Développement rapide
- Couplage faible
- Code concis
- DRY (Don't Repeat Yourself)
- Explicite est mieux qu'implicite
- SQL efficace
- Gestion des urls
- Modèles incluent toute la logique de domaine

4. Les modèles

- ORM (Object-Relational Mapper).
- Définition du modèle de la base de donnée en Python.

Modèle Django	SGBD
classe	table
objet	ligne d'une table
champ	colonne d'une table

- Création automatique des tables à l'aide des outils fournis par Django.
- Gèrent les relations un-à-un, un-à-plusieurs et plusieurs-à-plusieurs.

4. Les modèles

- Fournissent une librairie en Python permettant de manipuler des objets dans la base de données sans utiliser de SQL.
- Contiennent des informations supplémentaires destinées à l'interface d'administration.
- Possibilité de générer le modèle depuis la base de données.

4. Les modèles


```python
from django.db import models

class Personne(models.Model):
    nom    = models.CharField(maxlength=30)
    prenom = models.CharField(maxlength=30)
    # ...

class Exemple(models.Model):
    site = models.OneToOneField(Site)
    # ...
    fabriquant = models.ForeignKey(Fabriquant)
    # ...
    ingredients = models.ManyToManyField(Ingredient)
    # ...
```

5. Les vues

- Les développeurs passent une grande partie de leur temps à écrire des fonctions et des vues.
- Une fonction représentant une vue prend un objet requête en paramètre ainsi que les paramètres extraits de l'url et retourne un objet réponse.
- La vue qui sera exécutée dépend de la liste d'expressions régulières à laquelle sont comparées les requêtes entrantes.
- Certaines vues sont tellement courantes que Django les inclut en tant que vues génériques.

5. Les vues


```
from django.conf.urls.defaults import *

info_dict = { 'queryset': Article.objects.all() }

urlpatterns = patterns('django.views.generic.date_based',
    (r'^blog/$', 'archive_index', info_dict),
    (r'^blog/details/(?P<post>\w+)/$', 'object_detail', dict (info_dict, slug_field='post')),
    (r'^blog/(?P<year>\d{4})/$', 'archive_year', info_dict),
)
```

```
from django.shortcuts import render_to_response, get_object_or_404
from mysite.polls.models import Poll

def detail(request, poll_id):
    p = get_object_or_404(Poll, pk=poll_id)
    return render_to_response('polls/detail.html', {'poll': p})
```

6. Les templates

- Les templates ne contiennent aucun code exécutable (sécurité).
- N'utilisent pas seulement les formats XML et HTML, mais tous les formats textes.
- Django propose son propre langage de template, permettant de mettre en forme les données sans taper de code Python.
- Les templates sont extensibles : création simple de tags et de filtres.

6. Les templates

- Variables remplacées lors de l'évaluation :
  - {{ sportif.nom }}
- Tags (18 prédéfinis) :
  - {% if sportif.playVolley %} ... {% endif %}
- Filtres (47 prédéfinis) :
  - {{ sportif.nom|upper }}
- Héritage de templates : des balises de blocs indiquent quelles parties peuvent être redéfinies.
- Possibilités de créer ses propres tags et filtres.

7. Les plus de Django

- Les applications : Django fournit un ensemble d'applications complètes réutilisables dans les projets :
  - interface d'administration
  - système d'authentification
  - système de commentaires
  - créateur de flux RSS
  - mise en forme du texte en HTML à partir d'un langage de balises simple.
  - Autres...

7. Les plus de Django

- Les middlewares : plugings permettant d'effectuer des actions directement sur les requêtes et les réponses avant que celles-ci n'arrivent à la vue :
  - possibilité de cache
  - compression des pages
  - gestion des sessions
  - authentification par HTTP
  - système de transactions
  - possibilité de développer ses propres middlewares.

8. Comparatif avec Rails

Django	**Rails**
Documentation de référence présente et gratuite	Documentation de référence sous forme d'un livre
Pas de magie	Magique
Ne supporte pas Ajax de manière officielle	Supporte Ajax
Définition de la base en Python	La base de données doit être définie au début en SQL
Interface d'administration	-

Terminal Fichier Édition Défilement arrière Police Fenêtre Aide jeu. 05:01 (99 %)

Terminal — bash — ⌘1

```
mumu:~/Desktop migou$
```

Démonstration

9. Conclusion

- Framework très puissant
- Offre un grand nombre de fonctionnalités
- Très simple d'accès, basé sur Python
- Met à disposition de nombreux outils permettant de faciliter le développement (vue générique, application existante, ...)
- Documentation de référence très détaillée, un livre en ligne, une large communauté de développeurs

Questions ?