Python.

Веб фреймворки и темплейтные языки.

Web фреймворки

- Django
- Pylons
- Pyramid
- Web2py
- Zope
- …

Django

The Web framework for perfectionists with deadlines

Особенности:

- Слабая связанность (разделение на максимально независимые компоненты)
- Don’t repeat yourself (DRY)
- Явное лучше неявного
- Отсутствие нового языка программирования в шаблонах
- Реализация идеологии MVC

MVC

- Model - доступ к данным, обрабатывается слоем работы с базой данных
- View - часть, которая определяет, какие данные получать и как их отображать, обрабатывается представлениями и шаблонами.
- Controller - часть, которая выбирает представление в зависимости от пользовательского ввода, обрабатывается самой средой разработки, следуя созданной вами схемой URL, и вызывает соответствующую функцию Python для указанного URL

Project & Application

- Приложение — это переносимый набор некой функциональности

- Проект — это экземпляр определённого набора кода Django-приложений и конфигурация для этих приложений

Начало работы

- Установить django

- Настроить базу данных
  - ✓ Поддерживаемые СУБД: PostgreSQL, SQLite3, MySQL, Oracle
  - ✓ Настройка сервера базы данных
  - ✓ Установка библиотеки Python для поддержки необходимой базы данных

$ django-admin.py startproject TestDjangoProject

$ django-admin.py startapp mysite

Сервер разработки:

$ python manage.py runserver

http://127.0.0.1:8000/

Структура проекта

Пример

settings.py

- Файл настроек для всего приложения
- Может быть разбит на несколько файлов в папке settings
- Настройки БД, темплейтов, пути к статическому контенту, разрабатываемые приложения, директории, в которых хранятся шаблоны и др.

manage.py

- файл для запуска

- $ python manage.py runserver

models.py

- каждая модель соответствует одной таблице в базе данных
- Каждая модель представлена в виде класса Python, который является потомком класса django.db.models.Model

Добавление модели в приложение

- Зарегистрировать приложение в INSTALLED_APPS (settings.py)

- Проверить модели - python manage.py validate

- Синхронизироваться с базой данных:

  $ python manage.py syncdb

Доступ к данным

- API : python manage.py shell
- Poll.objects
- SQL INSERT && UPDATE ALL == .save()
- SQL SELECT * == .all
- SQL SELECT … == .filter(...)
- SQL DELETE == .delete()

Шаблоны

- Шаблон Django — это строка текста, которая предназначена для разделения представления документа от его данных

- Основной способ использования:

  - ✓ Создать объект Template, передав ему шаблон в виде строки.

  - ✓ Вызвать метод render() объекта Template с набором переменных(контекст). Метод возвратит полностью обработанный шаблон в виде строки, все переменные и шаблонные теги будут вычислены в соответствии с контекстом.

Пример

```
>>> from django import template

>>> t = template.Template('My name is {{ name }}.')

>>> c = template.Context({'name': 'Adrian'})

>>> print t.render(c)
My name is Adrian.

>>> c = template.Context({'name': 'Fred'})

>>> print t.render(c)
My name is Fred.
```

Теги

{% if %} {% else %} {% endif %}

{% for %} {% endfor %}

{% include %}

{% block %}

...

views.py

```
from django.http import HttpResponse


def hello(request):
    return HttpResponse("Здравствуй, Мир")
```

urls.py

- *Файл привязки URL* можно рассматривать как таблицу с содержанием сайта.

```
from django.conf.urls.defaults import *
from mysite.views import hello

urlpatterns = patterns('',
    ('^hello/$', hello),
)
```

urls.py

- первый элемент -- шаблон регулярного выражения
- второй элемент — функция представления, которая должна использоваться при совпадении данного шаблона.

- любой запрос к URL /hello/ должен быть обработан с помощью функции представления **hello**

Динамические URL

- Плохо

```
urlpatterns = patterns('',
  ('^time/$', current_datetime),
  ('^time/plus/1/$', one_hour_ahead),
  ('^time/plus/2/$', two_hours_ahead),
```

- Хорошо

```
urlpatterns = patterns('',
  (r'^time/plus/\d{1,2}/$', hours_ahead),)
```

Обработка запроса

- Приходит запрос к /hello/.

- Django просматривает файл привязки в поисках первого шаблона, который совпадёт с запрошенным URL.
- Если такой шаблон найден, Django вызывает ассоциированную с ним функцию представления.
- Функция представления возвращает HttpResponse.
- Django преобразовывает HttpResponse в соответствующий HTTP отклик, который реализует страницу.

Пример

Темплейтные языки

- Jinja 2
- Mako
- Django

- Genshi
- Cheetah

Документация

- https://www.djangoproject.com/
- http://www.djangobook.com/
- http://www.djbook.ru/