Hateful group

short but tough group

First of all, I'd like to introduce to you what our group discussed. It's about Django and turbogears framework. In the following content, our group will bring you the history, advantages and disadvantages, framework and code of both. Here we are

Chang Yusheng

Huang Yujian

Liu Xunhai

Xie Songjiang

Li Yuanyucheng

Guo Zhekai

Long Lin

Li Zhiling

I hope our works will be liked by you

01

History

What happens to the Django and
Turbogears frameworks
over time.

In this section, we will talk about the birth and development of Django and turbo gears, and which versions they have experienced. Their main characteristics are whether they have changed, which we will explain one by one in the following ppt.

History

History is the human activity related to material things. It is necessary for Web Framework

source

Django, pronounced [` d ʒ æ ŋ g goes ʊ], is written in python open source web development framework, and follow the MVC design

time

The framework was developed for the purpose of developing news-focused websites and was released in July 2005.

purpose

The main purpose of Django is to develop a "database" driven web site easily. It emphasizes code reuse, and multiple components can easily serve the entire framework as "plug-ins."

History

History is the human activity related to material things. It is necessary for Web Framework

source

TurboGears is a python-based Web development framework

purpose

It help developers integrate many major components into a Web framework that provides front-end to back-end Web integration. The code is hosted on Github under the MIT open source license

02

description of Django and turbogears

What exactly are Django and turbogears?
We'll give you a brief overview of
them in the following sections

Django, Flask, Tornado, etc., are common frameworks for learning python. Compared with other frameworks, Django framework has the advantage of being large and complete. etc..

Django framework

MVC is the well-known pattern of breaking an application into its three components :model,view, and controller. Among them:

M

manages the state of the application and restricts behavior that changes the state .

V

access the model according to the actions, and call the "view" to display the data. become the link between the"model" and the "view".

C

responsible for formatting the data and presenting it to the user.

Django framework

Django focuses more on models, templates, and Views, called the MTV pattern:

M

stands for Model, the data access layer. This layer handles all transactions related to the data: how to access it, how to validate it, what behaviors are included, and the relationships between the data.

T

stands for Template, the presentation layer. This layer handles performance-related decisions about how to display in a page or other type of document.

V

This layer contains the logic to access the model and retrieve the appropriate template. You can think of it as a bridge between the model and the template.

TurboGears framework

TurboGears is a great Web site development framework for the Python programming language.

It is made up of a number of subprojects that help us bring together many important components.

Turbogears support through ORM

SQLite
MySQL
Postgres
Firebird
MS SQL Server
Oracle

03

TurboGears and django features

in order to achieve a very efficient website structure, a clean project

Why Django is so popular, and why turbogears is far less popular than Django, we can see part of the reasons in the following features.

Django

features

- there is a sound ORM relational mapping
- it has powerful routing mapping function
- there is a sound implementation of the view template
- a sound background management system
- strong cache support

turbogears

features

•Best of Breed Modules for Python

• Interactive Programming Experience

• Integration, which can be freely combined with preference

• built in web server, ORM, AJAX capabilities, can generate HTML, json and other formats by default

• you can start developing web applications without installing the Apache web server, which can be deployed separately or with Apache, lighttpd web server

• you can start developing database websites without installing the database MySQL/PostgreSQL

• convenient deployment capability

• multiple Extension support

04

Advantages and disadvantages

Secret of success of Django framework and turbogears framework

Why do Django framework and turbo gears framework stand out from so many web frameworks? In the following part, we will answer for you

.

Advantage of Django

Secret of success of Django framework

Good documentation:
Powerful database access components:

Full features, full elements: with a large number of commonly used tools and frameworks (such as paging, authorization, authority management, etc.), suitable for rapid development of enterprise web sites.

Administrator: a complete backend data management and control platform can be implemented with a few simple configurations and lines of code.
it's easy to find code errors.

DisAdvantage of Django

deficiencies

Django comes with some lightweight applications that don't need it, and it's not portable like Flask.

Over-encapsulation: many classes and methods are encapsulated, which makes it harder to change.

compared to C and C ++, Django's performance is low

TurboGears:

A lot of advantages and a fatal disadvantage

1

ROR mode

Is there a certain amount of time to accumulate and corresponding development tools and modules

2

Support for Ajax, I18N

3

Less development effort

4

Drawbacks

SQLObject (ORM tool) is protected by the LGPL(smaller general public license). The license does not require that applications. However, some companies will ban lgpl-protected software.

05

Django VS turbogears

Who is better?

Django and TurboGears are MVC style frameworks that developers can use to quickly develop Web sites in the Python language. To choose the technology that best suits your needs, consider the following differences:

Background:

Django and TurboGears are MVC style frameworks that developers can use to quickly develop Web sites in the Python language

TurboGears has a better community drive than Django because it is built on top of existing open source components.

The TurboGears project is based on a consumer product that focuses on fat-client applications and pluggable architectures.

URLs

TurboGears' request distribution mechanism is routed through controller classes and method names.

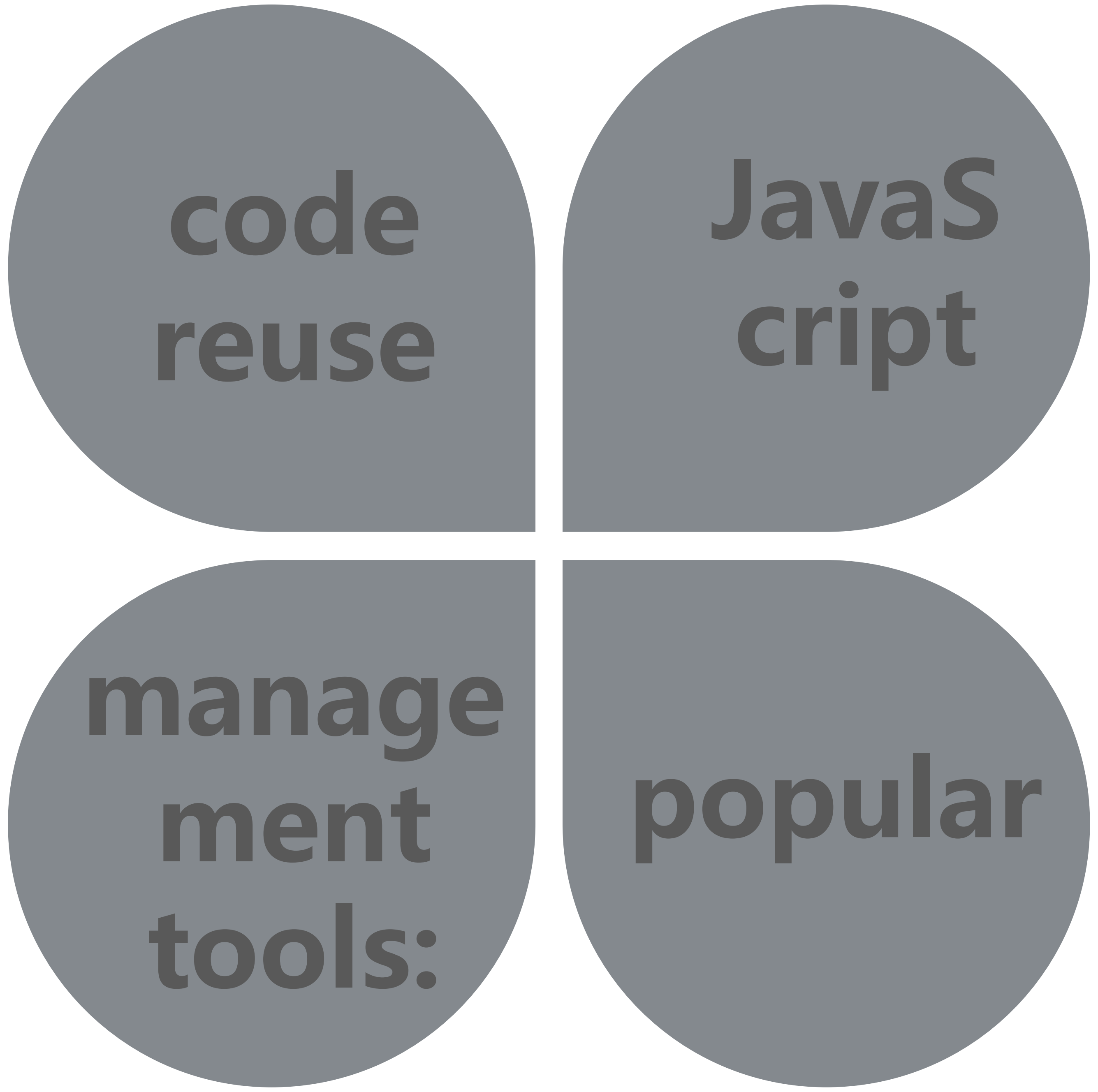
The TurboGears system is much faster to set up than Django

Django provides maximum control and flexibility. Django urls can simply be remapped to the application.

Django uses a separate regular configuration file , reducing the coupling between the URL path structure and the actual implementation.

Some other comparisons

The TurboGears team calls their project the big framework, which clearly expresses the idea that TG is a project made up of many existing components



code reuse

JavaS cript

manage ment tools:

popular

TurboGears starts by providing MochiKit, a JavaScript library while Django not

Both projects have a management interface. Django administration tools are aimed at end users so that you don't need to customize the tool every time you add new functionality.

Django

06

practical examples

turbogears

1

```
/dev$ tg-admin

TurboGears 0.9a5 command line interface

Usage: /usr/bin/tg-admin [command] [options]

Available commands:
      i18n  Manage i18n data
      info  Show version info
quickstart  Create a new TurboGears project
     shell  Start a Python prompt with your database available
       sql  Run the SQLObject manager
   toolbox  Launch the TurboGears Toolbox
    update  Update an existing turbogears project
```

2

```
/dev$ tg-admin quickstart
Enter project name: TG Commerce
Enter package name [tgcommerce]:
Selected and implied templates:
  turbogears#turbogears  web framework

Variables:
  package:  tgcommerce
  project:  TG-Commerce
Creating template turbogears
  Creating ./TG-Commerce/

    ... (output snipped) ...
```

3

```
~/dev/TG-Commerce$ python start-tgcommerce.py

... (output snipped) ...

05/Mar/2006:11:31:54 HTTP INFO Serving HTTP on http://localhost:8080/
```

4

```
sqlobject.dburi="notrans_sqlite:///path/to/devdir/TG-Commerce/tgcommerce.database"
server.environment="development"
autoreload.package="tgcommerce"
```

5

```
from sqlobject import *
from turbogears.database import PackageHub

hub = PackageHub("tgcommerce")
__connection__ = hub
```

6

```
class Category(SQLObject):
    name = StringCol(length=64)
    parent = ForeignKey('Category', default=None)
    subcategories = MultipleJoin('Category', joinColumn='parent_id')
    products = MultipleJoin('Product')

class Product(SQLObject):
    name = StringCol(length=64)
    sku = StringCol(length=64)
    price = CurrencyCol(notNone=True, default=0.0)
    category = ForeignKey('Category')
```

7

```
~/dev/TG-Commerce$ tg-admin sql sql
Using database URI sqlite:///home/ubuntu/dev/TG-Commerce/tgcommerce.db
CREATE TABLE category (
    id INTEGER PRIMARY KEY,
    name VARCHAR(64),
    parent_id INT
);

CREATE TABLE product (
    id INTEGER PRIMARY KEY,
    name VARCHAR(64),
    sku VARCHAR(64),
    price DECIMAL(10, 2) NOT NULL,
    category_id INT
```

06

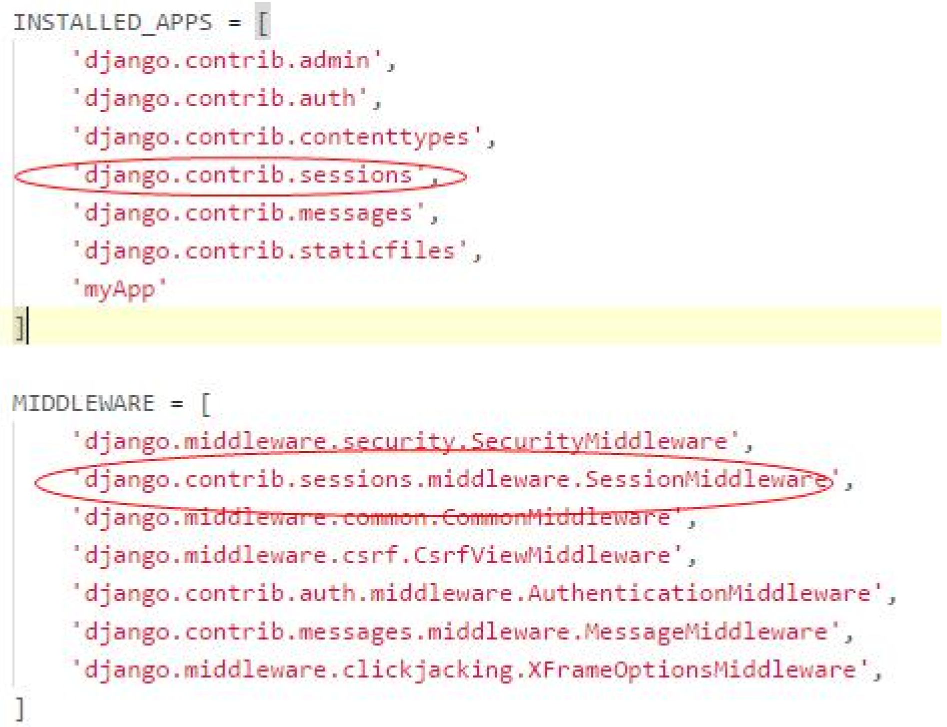
practical examples

Django

1

```
from django.conf.urls import url
from django.contrib import admin
from myApp import views
urlpatterns = [
 url(r’^admin/’, admin.site.urls),
 url(r’^login/$’, views.login),
 url(r’^index/$’, views.index),
]
```

2



```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'myApp'
]

MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
]
```

3

```
# -*- coding: utf-8 -*-
from django.shortcuts import render, redirect, HttpResponse

# Create your views here.
# from django.contrib import auth
# from django.contrib.auth.decorators import login_required

from myApp.models import Students,Grades,User
from functools import wraps
```

4

```
def check_login(f):
 @wraps(f)
 def inner(request,*arg,**kwargs):
  if request.session.get(’is_login’)==’1’:
   return f(request,*arg,**kwargs)
  else:
   return redirect(’/login/’)
 return inner
```

5

```
from django.db import models
class User(models.Model):
 username=models.CharField(max_length=16)
 password=models.CharField(max_length=32)
```

6

```
<body>

<h1>欢迎登录！</h1>
<form action=″/login/″ method=″post″>
 {% csrf_token %}
 <p>
  用户名：
  <input type=″text″ name=″username″>
 </p>
 <p>
  密码：
  <input type=″text″ name=″password″>
 </p>
 <p>
  <input type=″submit″ value=″登录">
 </p>
 <hr>
</form>
</body>
```

THANKS FOR WATCHING

第一部分:
Django的历史背景:
http://www.nowamagic.net/academy/detail/1318216
TurboGears的历史背景：
https://blog.csdn.net/iteye_16821/article/details/81616137
第二部分:
Django的简介:
https://blog.csdn.net/yangsen99/article/details/82505636
TurboGears的简介：
https://www.leiue.com/what-is-turbogears

第三部分:
Django的特点:
https://www.jianshu.com/p/e37318135a8a
TurboGears的特点：
https://www.leiue.com/what-is-turbogears
第四部分:
Django的优缺点:
https://www.jianshu.com/p/e37318135a8a
TurboGears的优缺点：
https://www.leiue.com/what-is-turbogears

第五部分:

Django和**TurboGears的比较：**
https://blog.csdn.net/iteye_16821/article/details/81616137