Django

turbogears

HISTORY

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."

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

DESCRIPTION

MVC

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.

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

MTV

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.

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

"

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

Li Yuanyucheng

Huang Yujia
n

Guo Zhekai

Liu Xunhai

Long Lin

Xie Songjiang

Li Zhiling

FEATURES

Django

features

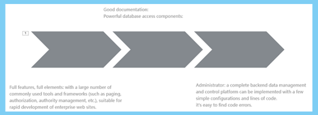
- there is a sound ORM relational mapping
- it has powerful routing mapping function
- there is a sound implementation of the view template
- a sound background management system
- strong cache support

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

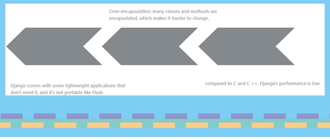
ADVANTAGES AND DISADVANTAGES



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.



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

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

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

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.

DJANGO VS TURBOGEARS

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.

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.

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

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.

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

Django