Django- Python
Quick Tutorial

From beginner To advance

Django-Python

Django is a web development framework that assists in building and maintaining quality web applications. Django helps eliminate repetitive tasks making the development process an easy and time saving experience.

Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design. Django makes it easier to build better web apps quickly and with less code.

Advantage of Django

Here are few advantages of using Django which can be listed out here

Object-Relational Mapping (ORM) Support – Django provides a bridge between the data model and the database engine, and supports a large set of database systems including MySQL, Oracle, Postgres, etc. Django also supports NoSQL database through Django-nonrel fork. For now, the only NoSQL databases supported are MongoDB and google app engine.

Multilingual Support – Django supports multilingual websites through its built-in internationalization system. So you can develop your website, which would support multiple languages.

Framework Support – Django has built-in support for Ajax, RSS, Caching and various other frameworks.
Administration GUI – Django provides a nice ready-to-use user interface for administrative activities.

Development Environment – Django comes with a lightweight web server to facilitate end-to-end application development and testing.

Django- URL Mapping

The marked line maps the URL "/home" to the hello view created in myapp/view.py file. As you can see above a mapping is composed of three elements −

The pattern − A regexp matching the URL you want to be resolved and map. Everything that can work with the python 're' module is eligible for the pattern (useful when you want to pass parameters via url).

The python path to the view − Same as when you are importing a module.

The name − In order to perform URL reversing, you'll need to use named URL patterns as done in the examples above. Once done, just start the server to access your view via :http://127.0.0.1/hello

Django- Environment

Step 3 – Database Setup

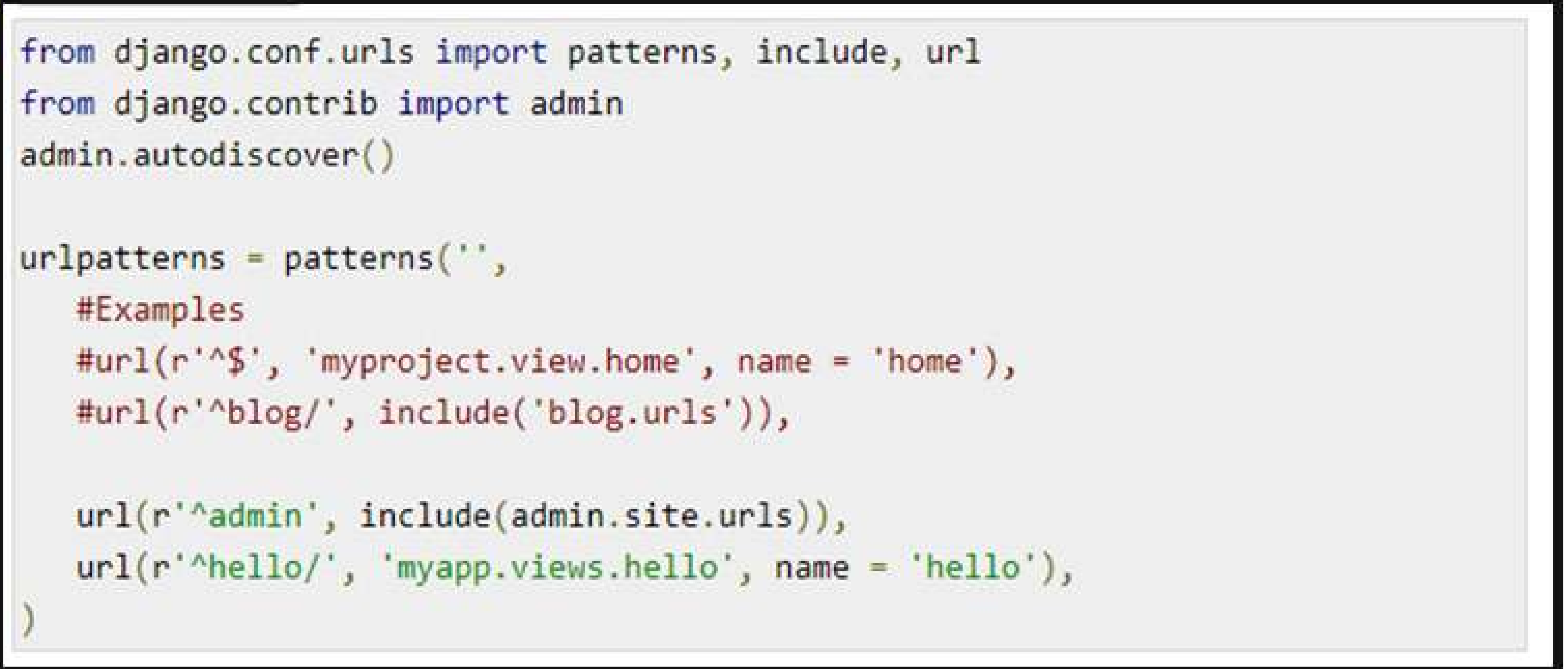
Django supports several major database engines and you can set up any of them based on your comfort.

- MySQL (http://www.mysql.com/)
- PostgreSQL (http://www.postgresql.org/)
- SQLite 3 (http://www.sqlite.org/)
- Oracle (http://www.oracle.com/)
- MongoDb (https://django-mongodb-engine.readthedocs.org)
- GoogleAppEngine Datastore (https://cloud.google.com/appengine/articles/django-nonrel)

You can refer to respective documentation to installing and configuring a database of your choice.

Django- URL Mapping

When a user makes a request for a page on your web app, Django controller takes over to look for the corresponding view via the url.py file, and then return the HTML response or a 404 not found error, if not found. In url.py, the most important thing is the **"urlpatterns"** tuple. It's where you define the mapping between URLs and views. A mapping is a tuple in URL patterns like



```python
from django.conf.urls import patterns, include, url
from django.contrib import admin
admin.autodiscover()

urlpatterns = patterns('',
   #Examples
   #url(r'^$', 'myproject.view.home', name = 'home'),
   #url(r'^blog/', include('blog.urls')),

   url(r'^admin', include(admin.site.urls)),
   url(r'^hello/', 'myapp.views.hello', name = 'hello'),
)
```

Project Structure

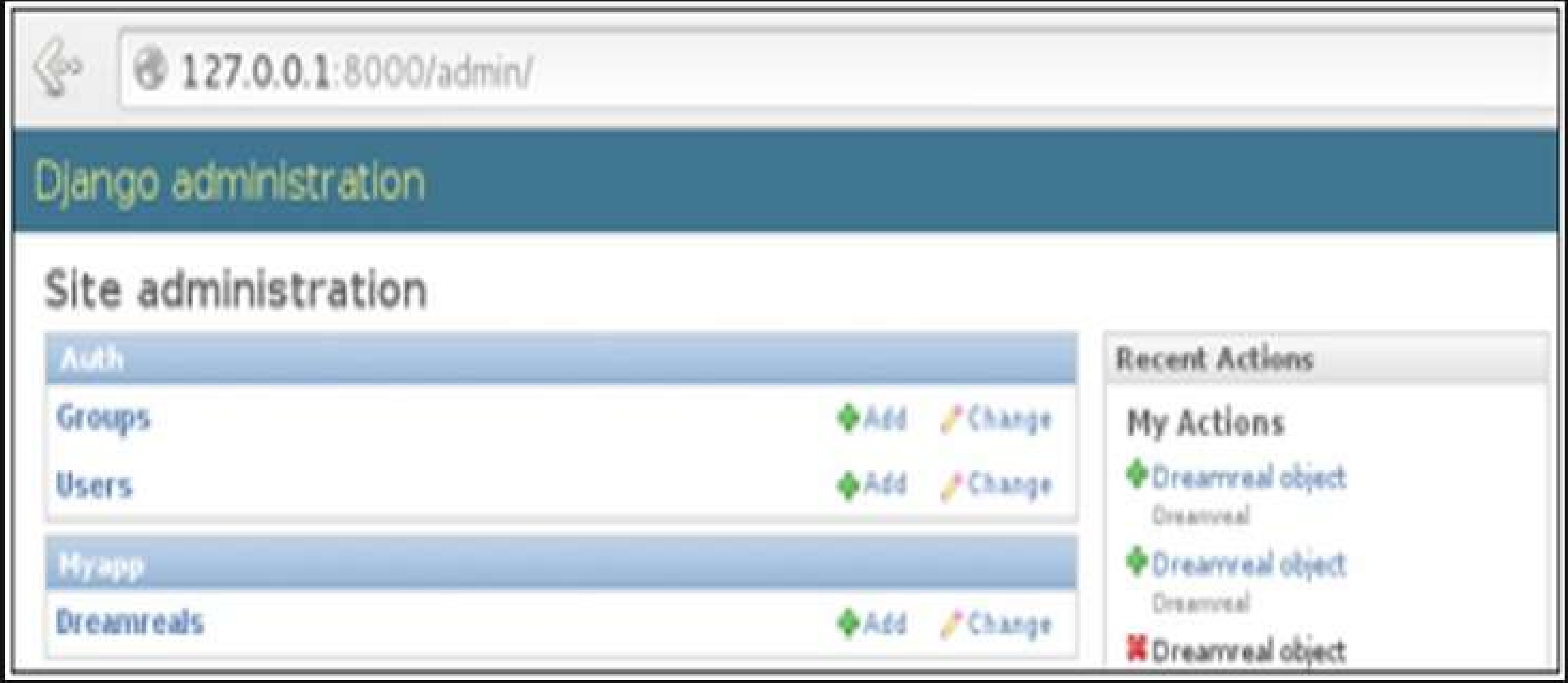
The “myproject” folder is just your project container, it actually contains two elements –
manage.py – This file is kind of your project local django-admin for interacting with your project via command line (start the development server, sync db...). To get a full list of command accessible via manage.py you can use the code –

```
$ python manage.py help
```

The “myproject” subfolder – This folder is the actual python package of your project. It contains four files –
__init__.py – Just for python, treat this folder as package.
settings.py – As the name indicates, your project settings.
urls.py – All links of your project and the function to call. A kind of ToC of your project.
wsgi.py – If you need to deploy your project over WSGI.

Organizing Your URLs

So far, we have created the URLs in “myprojects/url.py” file, however as stated earlier about Django and creating an app, the best point was to be able to reuse applications in different projects. You can easily see what the problem is, if you are saving all your URLs in the “projecturl.py” file. So best practice is to create an “url.py” per application and to include it in our main projects url.py file (we included admin URLs for admin interface before).



Creating Project

Whether you are on Windows or Linux, just get a terminal or a **cmd** prompt and navigate to the place you want your project to be created, then use this code

```
$ django-admin startproject myproject
This will create a "myproject" folder with the following structure
```

This will create a "myproject" folder with the following structure

```
myproject/
 manage.py
myproject/
__init__.py
settings.py
urls.py
wsgi.py
```

Django- Environment

Step 2 - Installing Django

Installing Django is very easy, but the steps required for its installation depends on your operating system. Since Python is a platform-independent language, Django has one package that works everywhere regardless of your operating system.

You can download the latest version of Django from the link http://www.djangoproject.com/download.

UNIX/Linux and Mac OS X Installation

You have two ways of installing Django if you are running Linux or Mac OS system –

You can use the package manager of your OS, or use easy install or pip if installed.

Install it manually using the official archive you downloaded before.

We will cover the second option as the first one depends on your OS distribution. If you have decided to follow the first option, just be careful about the version of Django you are installing.

Let's say you got your archive from the link above, it should be something like Django-x.xx.tar.gz:

Extract and install.

Django History

2003 – Started by Adrian Holovaty and Simon Willison as an

internal project at the Lawrence Journal-World newspaper.

2005 – Released July 2005 and named it Django, after the jazz

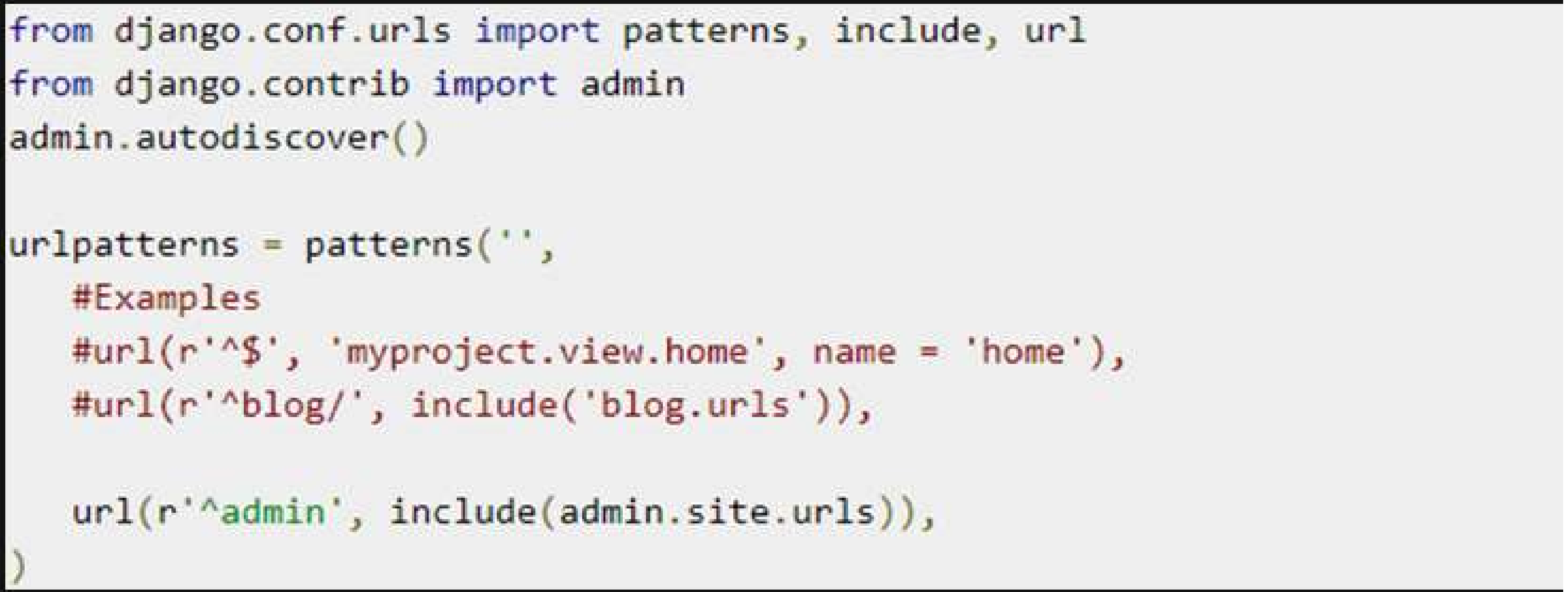
guitarist Django Reinhardt.

2005 – Mature enough to handle several high-traffic sites.

Current – Django is now an open source project with contributors across the world

Django- URL Mapping

Django has his own way for URL mapping and it's done by editing your project url.py file **(myproject/url.py)**. The url.py file looks like



```
from django.conf.urls import patterns, include, url
from django.contrib import admin
admin.autodiscover()

urlpatterns = patterns('',
   #Examples
   #url(r'^$', 'myproject.view.home', name = 'home'),
   #url(r'^blog/', include('blog.urls')),

   url(r'^admin', include(admin.site.urls)),
)
```

Django- Environment

Step 1 – Installing Python

Django is written in 100% pure Python code, so you'll need to install Python on your system. Latest Django version requires Python 2.6.5 or higher

If you're on one of the latest Linux or Mac OS X distribution, you probably already have Python installed. You can verify it by typing *python* command at a command prompt. If you see something like this, then Python is installed.

```
$ python Python 2.7.5 (default, Jun 17 2014,
18:11:42)
 [GCC 4.8.2 20140120 (Red Hat 4.8.2-16)] on
linux2
```