

Joe Jasinski
Imaginary Landscape
2016

INTRODUCTION TO DJANGO

DJANGO

OVERVIEW

- ▶ History
- ▶ Who's using it
- ▶ Features
- ▶ Architecture
- ▶ Running Django
- ▶ Resources
- ▶ Demo

About Me

How I got into Django

- side projects

 - event scheduling tool

 - sms

- Imaginary Landscape

THE WEB FRAMEWORK FOR PERFECTIONISTS WITH DEADLINES

django.org

About Django

DJANGO

- ▶ Web Framework
- ▶ Open Source
- ▶ “Batteries Included”
- ▶ Scalable
- ▶ Well Documented

<https://docs.djangoproject.com/en/1.9/misc/design-philosophies/>

DESIGN PHILOSOPHIES

- ▶ Loosely Coupled
- ▶ Less Code
- ▶ Quick Development Cycle
- ▶ DRY
- ▶ Explicit better than implicit
- ▶ Consistency

<https://docs.djangoproject.com/en/1.9/misc/design-philosophies/>

Loosely Coupled - each component is independent from each other

Less Code - get more done quickly

Quick Development Cycle -

DRY - Everything should be specified in one place; no redundancy

Explicit better than implicit -

Consistency - clean design

Django History

<https://www.quora.com/What-is-the-history-of-the-Django-web-framework>

JOE JASINSKI

CREATED

- ▶ Adrian Holovaty
- ▶ Simon Willison
- ▶ Jacob Kaplan-Moss (early contributor)

- ▶ LJWorld Newspaper
- ▶ Created in 2003
- ▶ Open Sourced in 2005

Named after Django Reinhardt

No Good python web frameworks at the time

Key Members

DSF - 501c3

DjangoCon

Who's using it?

<https://www.shoop.io/en/blog/25-of-the-most-popular-python-and-django-websites/>

Pinterest



<https://www.quora.com/What-is-the-technology-stack-behind-Pinterest-1>



<http://instagram-engineering.tumblr.com/post/13649370142/what-powers-instagram-hundreds-of-instances>



<https://www.quora.com/What-is-the-technology-stack-of-the-Onion>



<https://blog.disqus.com/scaling-django-to-8-billion-page-views>



<https://blog.mozilla.org/webdev/2013/02/22/the-restful-marketplace/>



<https://labs.spotify.com/2013/03/20/how-we-use-python-at-spotify/>



<https://code.djangoproject.com/wiki/DjangoSuccessStoryBitbucket>

WHO'S USING DJANGO

- ▶ Spotify
- ▶ NASA
- ▶ Prezi
- ▶ Evenbrite
- ▶ SpotHero
- ▶ Pitchfork
- ▶ Washington Post
- ▶ National Geographic
- ▶ Rockhall
- ▶ PBS

<https://www.eventbrite.com/engineering/our-strategy-to-migrate-to-django/>

WHO'S USING DJANGO

- ▶ <https://www.djangosites.org/>
 - ▶ 1000s of examples
 - ▶ Some with source code available

LARGE DJANGO BASED PROJECTS

- ▶ Sentry
- ▶ OpenStack
Horizon
- ▶ edX
- ▶ Oscar
- ▶ dPaste

Features

Object Relational Mapper (ORM)

```
1 from django.db import models
2
3 class Contact(models.Model):
4
5     email = models.EmailField()
6     phone = models.CharField(max_length=60, blank=True, null=True)
7     age = models.IntegerField(blank=True, null=True)
8     message = models.TextField()
9     ip = models.CharField(max_length=60, blank=True, null=True)
10    created = models.DateTimeField(auto_now_add=True, editable=False)
11
12    def __str__(self):
13        return "%s" % (self.email)
14
```

<https://docs.djangoproject.com/en/1.9/topics/db/models/>

Simple Declarative syntax for models

Built-in Customizable User Framework

- ▶ Login / Logout / Reset Password framework
- ▶ User Groups
- ▶ Customize User Model
- ▶ Permission System
- ▶ Pluggable

<https://docs.djangoproject.com/en/1.9/topics/auth/customizing/#specifying-a-custom-user-model>

Login / Logout / reset password
User model


Customizable User Authentication


Access to your account


E-mail


Password

☐ Remember me

 Sign in with Facebook

 Sign in with Twitter

 Sign in with Googleplus

 Sign in with LinkedIn

[Register for a free account](#)

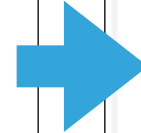
[Forgot password?](#)

<https://docs.djangoproject.com/en/1.9/topics/auth/customizing/>

Username / password - default
django social auth
django allauth

Robust Forms for Input

```
1 from django import forms
2
3 class TalkSubmissionForm(forms.Form):
4
5     title = forms.CharField(
6         max_length=50,
7         widget=forms.TextInput(
8             attrs={"class": "form-control"})
9     )
10    desc = forms.CharField(
11        label="Description",
12        widget=forms.Textarea(
13            attrs={"class": "form-control"}),
14        required=False)
```



Title:

Description:

Submit

<https://docs.djangoproject.com/en/1.9/topics/forms/>

Elegant URL Routing Based on Regex

```
1 from django.conf.urls import url
2 from . import views
3
4 url patterns = [
5     url(r'^photos/', include('photo.urls')),
6     url(r'^(?P<slug>[\w_]+)/$', 'core.page', name="page"),
7     url(r'^$', 'core.index', name="index"),
8 ]
```

<https://docs.djangoproject.com/en/1.9/topics/http/urls/>

Dynamic Administrative Interface

Django administration

Site administration

AUTHENTICATION AND AUTHORIZATION

Groups

+ Add

Change

Users

+ Add

Change

CONTACT

Contact Entries

+ Add

Change

<https://docs.djangoproject.com/ja/1.9/ref/contrib/admin/>

Simple but customizable interface

Database Migrations

- ▶ Manage Changes to your database
- ▶ Propagate model changes to database schema
- ▶ Stored in your codebase

<https://docs.djangoproject.com/en/1.9/topics/migrations/>

Multiple Database Support

```
1 DATABASES = {  
2     'default': {  
3         'NAME': 'app_data',  
4         'ENGINE': 'django.db.backends.postgresql',  
5         'USER': 'postgres_user',  
6         'PASSWORD': 's3krit'  
7     },  
8     'users': {  
9         'NAME': 'user_data',  
10        'ENGINE': 'django.db.backends.mysql',  
11        'USER': 'mysql_user',  
12        'PASSWORD': 'priv4te'  
13    }  
14 }
```

<https://docs.djangoproject.com/en/1.9/topics/db/multi-db/>

Data in separate databases (Beltone)

Supported Databases

- ▶ Postgres
- ▶ MySQL / MariaDB
- ▶ SQLite
- ▶ Oracle
- ▶ SAP SQL Anywhere
- ▶ IBM DB2
- ▶ MS SQL Server
- ▶ Firebird
- ▶ ODBC

<https://docs.djangoproject.com/en/1.9/ref/databases/>

Overridable Templates with Inheritance

<https://docs.djangoproject.com/en/1.9/ref/templates/>

Caching Framework

- ▶ Increase performance
- ▶ Reduce complex queries or lookups
- ▶ Multiple caching mechanisms available in Django

<https://docs.djangoproject.com/en/1.9/topics/cache/>

- rockhall: cache mostly static pages

Timezone Support



<https://docs.djangoproject.com/en/1.9/topics/i18n/timezones/>

Saves times as UTC in DB

Internationalization (I18N) and Localization (L10N)

- ▶ Multiple Languages
- ▶ Format Localization

<https://docs.djangoproject.com/en/1.9/topics/i18n/>

Generic Class Based Views

for view code reuse

<https://docs.djangoproject.com/en/1.9/topics/class-based-views/>

Python 2.7, 3.4, 3.5 Support

- ▶ Many projects still use 2.7
- ▶ Python 3.x more common
- ▶ Most popular Django 3rd party modules support 2.7 and 3.x

<https://docs.djangoproject.com/en/1.9/topics/python3/>

Since Django 1.5

Pluggable Template Engine

(with Jinja2 support)

<https://docs.djangoproject.com/en/1.9/topics/templates/#django.template.backends.jinja2.Jinja2>

Pluggable Storage Backend
for
Static Files and Uploaded Media

Project Templates

for standardizing project layouts

GIS Features

"GeoDjango"

<http://geodjango.org/>

Distance queries

Javascript UI Agnostic

Not tied to a javascript frontend

Unit Test and Integration Test Framework

<https://docs.djangoproject.com/en/1.9/topics/testing/>

Django Itself is Well Tested

10K+ Tests!

Commits to Django itself require **comprehensive testing**

Regular Release Support Schedule

Release Series	Release Date	End of mainstream support ¹	End of extended support ²
1.9	December 2015	August 2016	April 2017
1.10	August 2016	April 2017	December 2017
1.11 LTS	April 2017	December 2017	Until at least April 2020
2.0	December 2017	August 2018	April 2019


<https://goo.gl/NkqlU9>

<https://goo.gl/8T4G7m>

Long Term Support (LTS) Releases

- ▶ Past: Django 1.4
- ▶ Current: Django 1.8
- ▶ Future: Django 1.11
- ▶ 3+ years support
 - ▶ Back-ported bug fixes

Comprehensive Release Notes



Documentation

Django 1.9 release notes

December 1, 2015

Welcome to Django 1.9!

These release notes cover the new features, as well as some backwards incompatible changes you'll want to be aware of when upgrading from Django 1.8 or older versions. We've dropped some features that have reached the end of their deprecation cycle, and we've begun the deprecation process for some features.

Index of Django Releases

1.9.6, 1.8.13

- <https://www.djangoproject.com/weblog/2016/may/02/bugfix-releases/>
- <https://docs.djangoproject.com/en/1.9/releases/1.9.6/>
- <https://docs.djangoproject.com/en/1.8/releases/1.8.13/>
- [Bugfix Release](#)

1.9.5, 1.8.12

- <https://www.djangoproject.com/weblog/2016/apr/01/bugfix-releases/>
- <https://docs.djangoproject.com/en/1.9/releases/1.9.5/>
- <https://docs.djangoproject.com/en/1.8/releases/1.8.12/>
- [Bugfix Release](#)

1.9.4, 1.8.11

- <https://www.djangoproject.com/weblog/2016/mar/05/bugfix-releases/>
- <https://docs.djangoproject.com/en/1.9/releases/1.9.4/>
- <https://docs.djangoproject.com/en/1.8/releases/1.8.11/>
- [Bugfix Release](#)

Focus on Security

- ▶ XSS (Cross Site Scripting) protection
- ▶ CSRF (Cross Site Forgery) protection
- ▶ SQL Injection protection
- ▶ Clickjacking protection
- ▶ Common HTTPS protections
- ▶ Host Header validation

<https://docs.djangoproject.com/en/1.9/topics/security/>

XSS - inject javascript into forms

CSRF - protects against replay attacks

SQL injection - arbitrary SQL code

Clickjacking - site wrapped in iframe

Pluggable Password Hashers

```
1 PASSWORD_HASHERS = [  
2     'django.contrib.auth.hashers.PBKDF2PasswordHasher',  
3     'django.contrib.auth.hashers.PBKDF2SHA1PasswordHasher',  
4     'django.contrib.auth.hashers.BCryptSHA256PasswordHasher',  
5     'django.contrib.auth.hashers.BCryptPasswordHasher',  
6     'django.contrib.auth.hashers.SHA1PasswordHasher',  
7     'django.contrib.auth.hashers.MD5PasswordHasher',  
8     'django.contrib.auth.hashers.CryptPasswordHasher',  
9 ]
```

<https://docs.djangoproject.com/en/1.9/topics/auth/passwords/#how-django-stores-passwords>

- Why: passwords are hashed

Python Logging Built In

<https://docs.djangoproject.com/en/1.9/topics/logging/>

know what your app is doing

standardized

Light wrapper around python logging

What to log?

Where to log?

How much to log?

Architecture

MODEL VIEW TEMPLATE

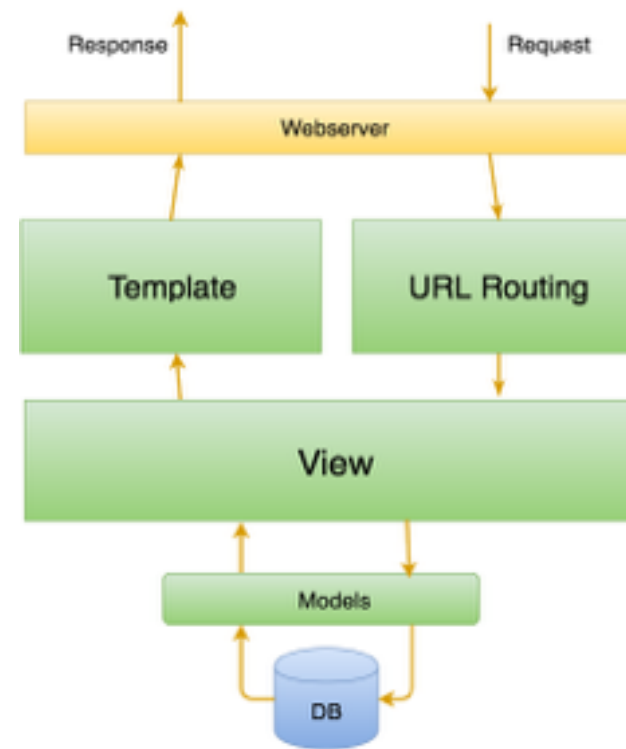


Model = Model



Controller = View

MODEL VIEW TEMPLATE



DJANGO MODEL

- ▶ Stores data persistently
- ▶ Multiple Database backend support
- ▶ Object Relational Mapper (ORM)
- ▶ Robust query syntax

DJANGO MODEL (CONT)

```
1 from django.db import models
2
3 class Contact(models.Model):
4
5     email = models.EmailField()
6     phone = models.CharField(max_length=60, blank=True, null=True)
7     age = models.IntegerField(blank=True, null=True)
8     message = models.TextField()
9     ip = models.CharField(max_length=60, blank=True, null=True)
10    created = models.DateTimeField(auto_now_add=True, editable=False)
11
12    def __str__(self):
13        return "%s" % (self.email)
14
```

DJANGO MODEL (CONT)

```
1 >>> Contact.objects.all()
2 [<Contact: jeo>, <Contact: jeo>, <Contact: jeo>, <Con
3
4 >>> Contact.objects.count()
5 6
6
7 >>> Contact.objects.all()[0]
8 <Contact: joe>
9
10 >>> Contact.objects.filter(age__gte=22)[0]
11 <Contact: joe>
12
13 >>> Contact.objects.filter(age__gte=23)
14 [<Contact: Joe>]
```

DJANGO VIEWS

- ▶ Performs business logic
- ▶ Input Validation
- ▶ Permission Logic
- ▶ Most other in-request processing
- ▶ Defines context variables to be used in the Template

Typically **1 view** per url

Takes **Request**; Returns **Response**

DJANGO VIEWS (CONT)

```
1 from django.shortcuts import render
2
3 def index(request):
4     context = {"hello": "world"}
5     return render(request, 'index.html', context)
```

DJANGO TEMPLATES

- ▶ Contains the site HTML
- ▶ Presents user interface to the user
- ▶ Utilizes the variables set by the view
- ▶ Flexible inheritance hierarchy
- ▶ Easy for designers to work with

DJANGO TEMPLATES (CONT) – BASE.HTML

```
1 <!DOCTYPE HTML>
2 <html lang="en">
3   <head>{% block extra_head %}{% endblock %}</head>
4   <body>
5     <div class="content">
6       {% block content %}
7         THIS GETS REPLACED
8       {% endblock content %}
9     </div>
10    <footer>{% now "Y" %}</footer>
11  </body>
12  {% block extra_foot %}{% endblock %}
13 </html>
14
```

DJANGO TEMPLATES (CONT)

```
1 {% extends "base.html" %}  
2  
3 {% block content %}  
4  
5 <h1>Welcome</h1>  
6 <p>Welcome to Django</p>  
7 <p>{{ hello }}</p>  
8  
9 {% endblock content %}  
10
```

Running Django

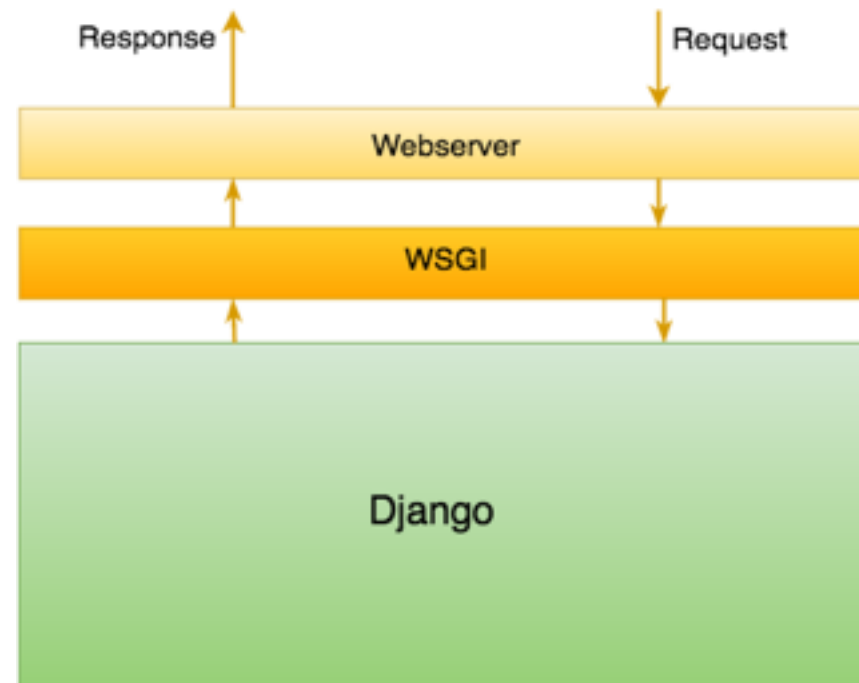
`./manage.py runserver`

- ▶ Simple development webserver
- ▶ Not suitable for production

WSGI (WEB SERVER GATEWAY INTERFACE)

- ▶ Specification
- ▶ Webserver talks to Webapp
- ▶ Layer between Webserver and Webapp

WSGI (WEB SERVER GATEWAY INTERFACE)



WSGI IMPLEMENTATIONS FOR DJANGO

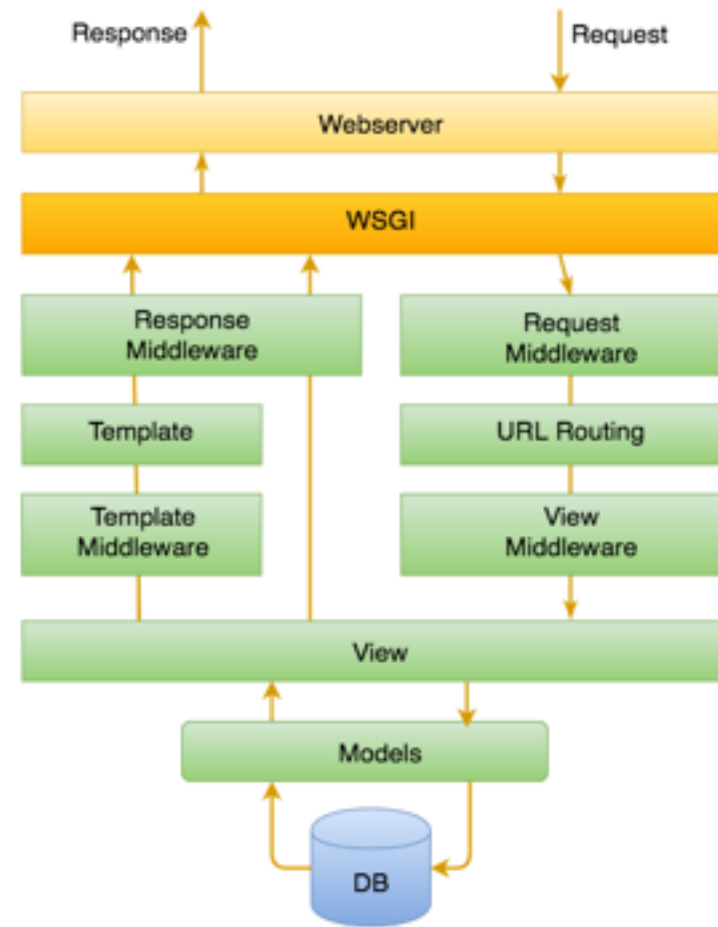
- ▶ uWSGI
- ▶ Gunicorn
- ▶ mod_wsgi (Apache)

Python software for running Django

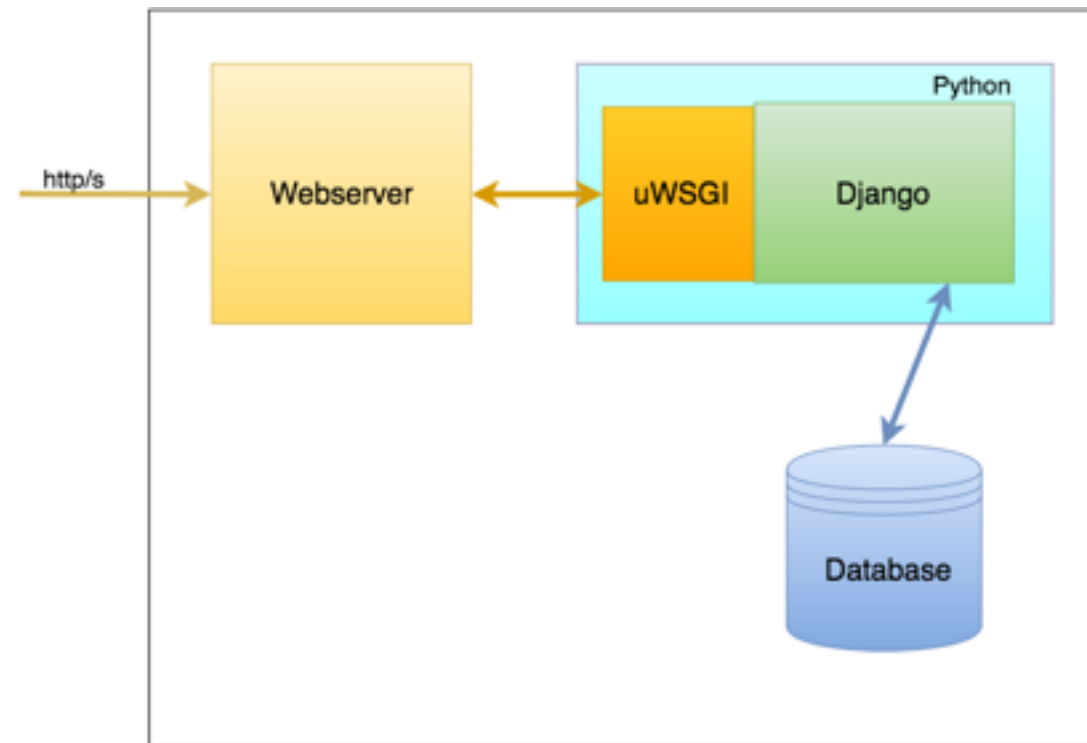
Gunicorn benefits

uWSGI benefits

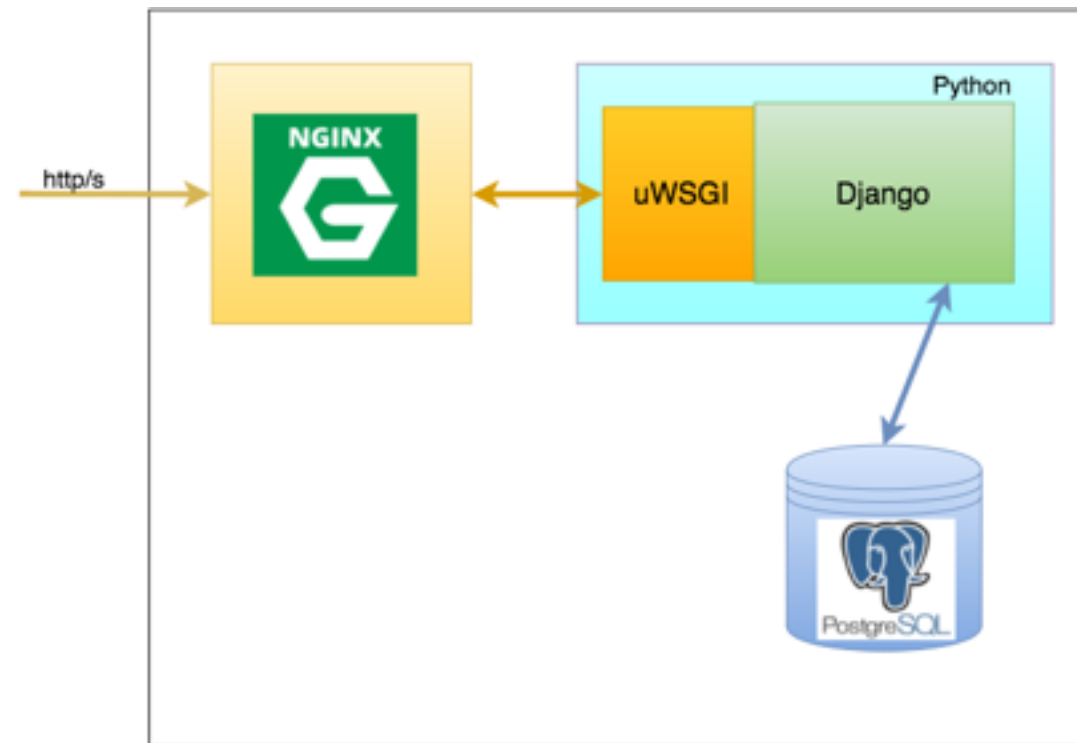
REQUEST / RESPONSE LIFECYCLE



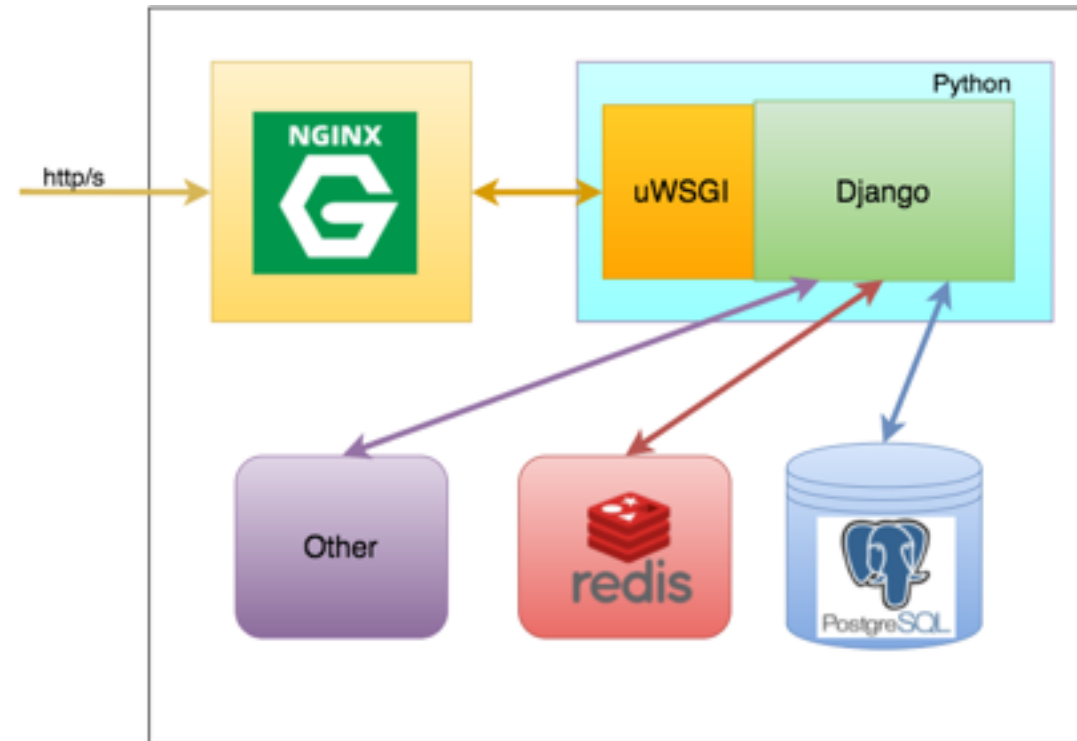
SINGLE SERVER DJANGO ARCHITECTURE



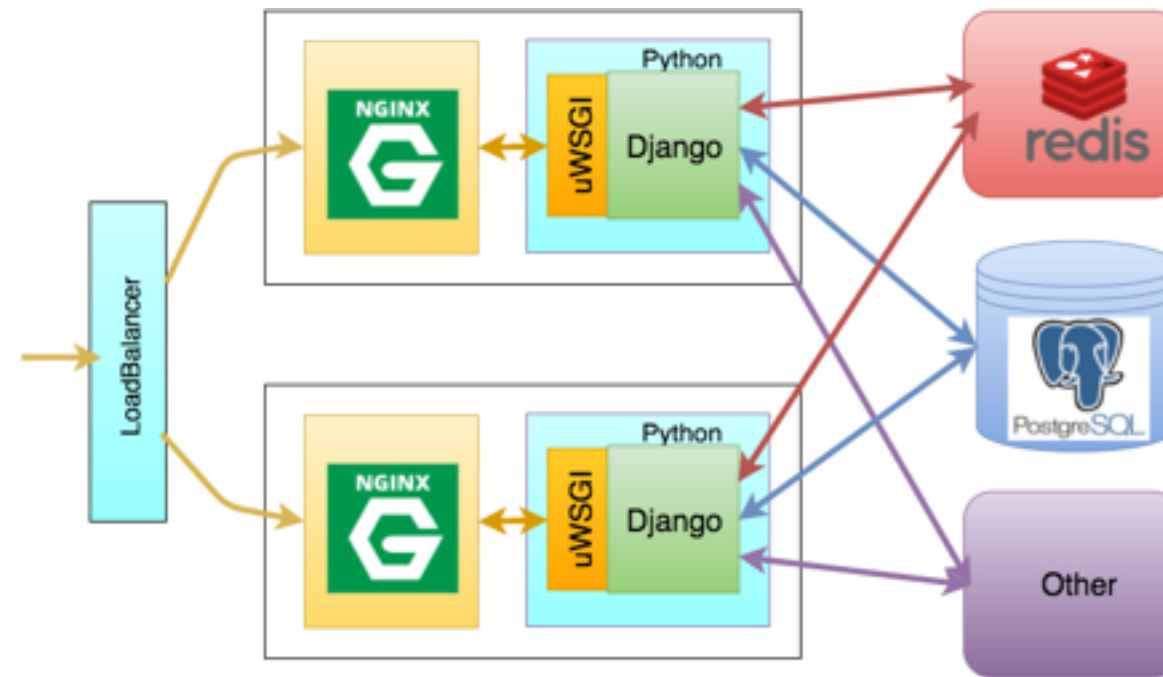
SINGLE SERVER DJANGO ARCHITECTURE



SINGLE SERVER DJANGO ARCHITECTURE



MULTI SERVER DJANGO ARCHITECTURE

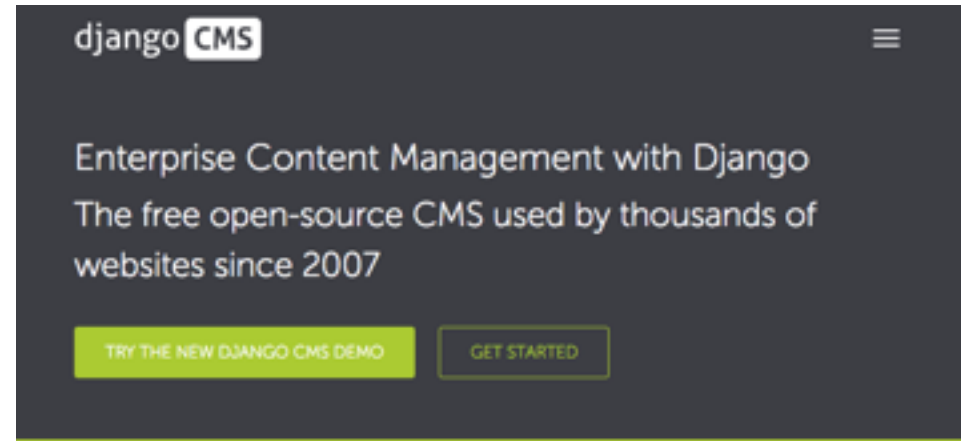


Django Modules

LARGE DJANGO BASED PROJECTS

- ▶ Django CMS /
Mezzanine / Wagtail /
FeinCMS
- ▶ Zinnia
- ▶ Filer / Filebrowser
- ▶ Haystack
- ▶ Celery
- ▶ Rosetta
- ▶ Django Rest
Framework
- ▶ Django SocialAuth
- ▶ django-debug-toolbar
- ▶ Django-extensions

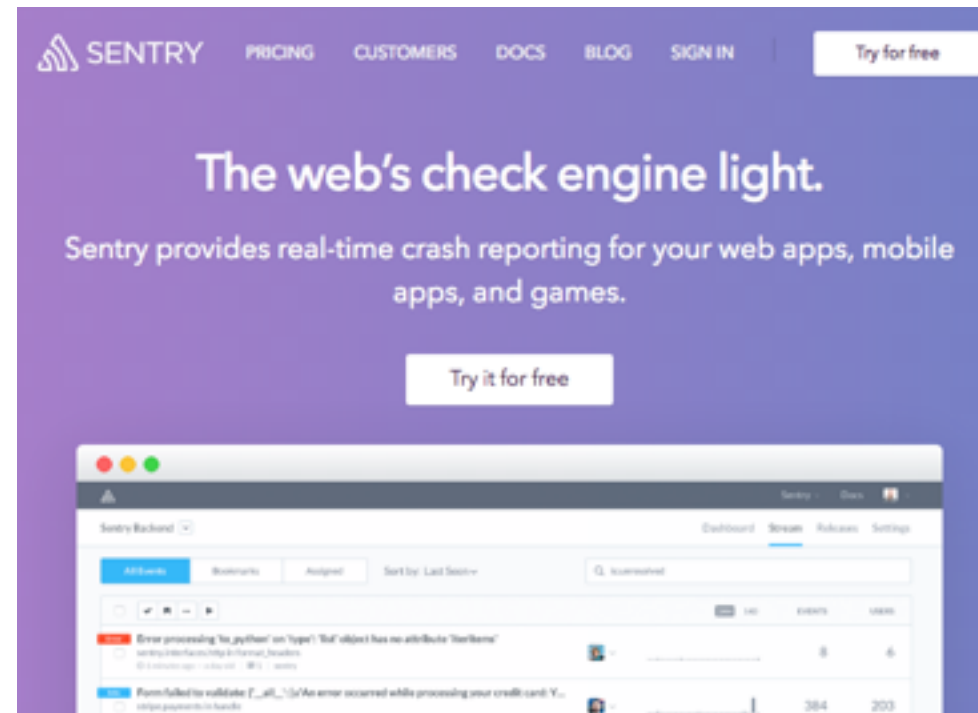
DJANGO CMS ([DJANGO-CMS.ORG](https://django-cms.org))




HAYSTACK – SEARCH FOR DJANGO (HAYSTACKSEARCH.ORG)




SENTRY – ERROR REPORTING ([GITHUB.COM/GETSENTRY](https://github.com/getsentry))



CELERY TASK QUEUE (WWW.CELERYPROJECT.ORG)

 Celery

DOCS AND SUPPORTINSTALLTUTORIALSCOMMUNITYSOURCECODE



Celery: Distributed Task Queue

Celery is an asynchronous task queue/job queue based on distributed message passing. It is focused on real-time operation, but supports scheduling as well.

The execution units, called tasks, are executed concurrently on a single or more worker servers using multiprocessing, `Eventlet`, or `gevent`. Tasks can execute asynchronously (in the background) or synchronously (wait until ready).

Celery is used in production systems to process millions of tasks a day.

Celery 3.1 Is Now Available!

on 11 Nov 2013, 7:57 p.m.

Rackspace Supports Celery

on 14 Oct 2013, 3:02 p.m.

[Read all news](#)

Current Version	Downloads
3.1.23	Last month: 98,690
Open changelog	Last week: 476
	Last day: 129

DJANGO REST FRAMEWORK ([DJANGO-REST-FRAMEWORK.ORG](https://djangorestframework.org))



MONGOENGINE (MONGOENGINE.ORG)



Not strictly django
but Django-like ORM for Mongoddb

DJANGO PACKAGES

Django Packages Home FAQ Sign in with Github

Search Go

ACTIVITIES AUTHORIZATION CAPTCHA CUSTOM MODELS DJANGO REST FR...
ADMIN INTERFACE AUTO-COMPLETE CHAT DATABASE MIGRA... DJANGO SHOP PL...
ANALYTICS AWARDS AND BADGES CMS DATA TOOLS DOCUMENT MANAG...
ANTI-SPAM BLOGS COMMENTING DEPLOYMENT E-COMMERCE
API CREATION BOOTSTRAPS CONFIGURATION DESIGN EMAIL
ASSET MANAGERS CACHING COOKIECUTTERS DEVELOPER TOOLS ERROR HANDLING
AUTHENTICATION CALENDAR COUNTRIES DJANGO-CMS FEED AGGREGATORS

Django Packages is a directory of reusable apps, sites, tools, and more for your Django projects.

3153 packages and counting!

1061 packages compatible with Python 3!

Add packages and grids

Know of any packages not listed here? Add them and file a pull request.

Latest Python 3 Package Releases

django-siteprefs v0.5.3
Reusable app for Django introducing site preferences system

django-sitemessage v0.7.2
Reusable application for Django introducing a message delivery framework

django-markdownx v1.5

Latest 5 Packages Added

fluentcms-jumbotron
Bootstrap 3 Jumbotron element for django-fluent-contents

fluentcms-contactform
A contact form plugin django-fluent-contents

fluentcms-pager
Bootstrap 3 Pager element for django-fluent-contents

3K+ django packages

DJANGO PACKAGES

[home](#) / [packages](#) / [Zinnia](#) 

Simple yet powerful and really extendable application for managing a blog within your Django Web site.

Usage	
Category	# Using This
App	43 

PyPI				
http://pypi.python.org/pypi/django-blog-zinnia				
Version	License	Released	Status	Python 3?
0.16	BSD	03/08/2016	Production/Stable	✓
0.15.2	BSD	06/22/2015	Production/Stable	✓
0.15.1	BSD	01/07/2015	Production/Stable	✓
0.15	BSD	12/18/2014	Production/Stable	✓
0.14.3	BSD	12/04/2014	Production/Stable	✓
0.14.2	BSD	09/16/2014	Production/Stable	✓

Documentation

<https://readthedocs.org/projects/django-blog-zinnia/>

Repo Activity Last fetched: 1 day, 4 hours ago [Fetch latest data](#)

Repo Hosting: Github

Repo URL	Commits	★	🔗
https://github.com/Fantomas42/django-blog-zinnia		1183	520

Comparison Grids

[Blogs](#) [Django-CMS](#)

Sites using this package

1. [Demo site of django-blog-zinnia](#)
2. [Euromedia France](#)
3. [Euromedia Group](#)
4. [Fantomas' Site](#)

Drill down and see info about a package

Community

DJANGOPROJECT.COM/COMMUNITY/

django

OVERVIEWDOWNLOADDOCUMENTATIONNEWSCOMMUNITYCODEABOUT♥ DONATE

Django Community 10769 people, 161 countries, 3154 packages and projects.

Mailing lists

Using Django

Get help with Django and follow announcements.

Enter email

You can also subscribe by sending an email to django-users+subscribe@googlegroups.com and follow the instructions that will


Contributing to Django

Contribute to the development of Django itself.

Enter email

You can also subscribe by sending an email to django-developers+subscribe@googlegroups.com and follow the instructions that will

Support Django!



Invest by Rubicon donated to the Django Software Foundation to support Django development. Donate today!

Get Involved

DJANGO USERS MAILING LIST (GOOGLE GROUP)


... for getting help with Django-related issues


Django users Shared publicly		
30 of 46422 topics (99+ unread) ☆ 0-1		
About		
	Can't get initial value into form correctly. (7) By Chris Kavanagh - 7 posts - 4 views	4:04 PM
	How made worked models.Sum() together with models.Count in Django annotations? (1) By Seš Volkyľany - 1 post - 1 view	11:42 AM
	django.contrib.apps migrations (4) By McKinley - 4 posts - 29 views	4:42 AM
	south migration: renaming foreign key and m2m tables (4) By scha...@gmail.com - 4 posts - 9 views	4:35 AM
	modelformset_factory generating "object has no attribute instance" error (1) By Farhan Khan - 1 post - 3 views	May 14
	Iterations of a list (4) By Shu Latif - 4 posts - 13 views	May 14
	A question on save_m2m() (1) By Ankush Thakur - 1 post - 9 views	May 14
	1.8 or 1.9? (6) By Ankush Thakur - 6 posts - 46 views	May 14

DJANGO DEVELOPERS MAILING LIST (GOOGLE GROUP)








... for discussions related to contributing to Django

Django developers (Contributions to Django itself) Shared publicly

30 of 8190 topics (99+ unread) 

About 

This group is for the discussion of the development of Django itself. If you want to ask questions about using Django, please post on [django-users](#).

	AJAX support in auth views (2) By Luke Hodkinson · 2 posts · 17 views	1:47 PM
	status of 1.10 release blockers By Tim Graham · 1 post · 27 views	May 14
	Fellow Report - May 14, 2016 By Tim Graham · 1 post · 14 views	May 14
	Template-based widget rendering (18) By Preston Timmons · 18 posts · 270 views	May 14
	Unicode normalization for username field (17) By Rick Leir · 17 posts · 261 views	May 14
	Process DEP for "official" non-core projects (9) By Andrew Godwin · 9 posts · 158 views	May 12
	Help Text Per Model Field Choice (1) By David Sanders · 2 posts · 65 views	May 12

STACKOVERFLOW

[django](#) × 119170

An open source web application framework, written in Python. Django is designed to reduce the effort required to create complex

56 asked today, 578 this week

[django-models](#) × 13847

The centerpiece of the Django object-relational mapping scheme is the Model.

8 asked today, 61 this week

[django-forms](#) × 76

Specific questions re Django web framework

21 asked this week, 13

[django-admin](#) × 5724

Django's built-in, automatic admin interface (django.contrib.admin) which is part of the Django Web framework for the Python

17 asked this week, 81 this month

[django-views](#) × 5563

MVC views; they control rendering (typically through templates), and the data displayed.

21 asked this week, 137 this month

[django-rest-framework](#)

A powerful and flexible API framework. Django is a high-level framework that encourages

8 asked today, 48 this

[django-urls](#) × 1519

The Django URL dispatch system

26 asked this month, 281 this year

[django-orm](#) × 1354

Django's ORM system, comprising its queryset and model systems.

13 asked this week, 34 this month

[django-haystack](#) ×

a modular search application. It aims to provide a fast and easy way for developers that does

22 asked this month, 2

[django-cms](#) × 896[django-authentication](#) × 868[django-celery](#) × 81

On first page of SO Tags
100K+ questions

CONFERENCES

- ▶ DjangoCon US
- ▶ DjangoCon EU
- ▶ DjangoCon AU
- ▶ Django Under the Hood
- ▶ A bunch of others...

INCLUSION

- ▶ Code of Conduct
 - ▶ <https://www.djangoproject.com/conduct/>
- ▶ Diversity Statement:
 - ▶ <https://www.djangoproject.com/diversity/>

CONTRIBUTING

- ▶ Source code
 - ▶ <https://github.com/django/django>
- ▶ Sprints
 - ▶ @ PyCon / DjangoCon
- ▶ <https://docs.djangoproject.com/en/1.9/internals/contributing/>
- ▶ IRC: #django on freenode

BOOKS

- ▶ Two Scoops of Django: Best Practices for Django 1.8
- ▶ Lightweight Django
- ▶ Test Driven Development with Python
- ▶ High Performance Django

RESOURCES

- ▶ Awesome Django
 - ▶ <https://github.com/rosarior/awesome-django>
- ▶ Django Girls Tutorial
 - ▶ <http://tutorial.djangogirls.org/en/>
- ▶ Tango with Django
 - ▶ <http://www.tangowithdjango.com/book17/>

Getting Started

INSTALL

- ▶ # optionally create virtualenv
- ▶ virtualenv myenv
- ▶ cd myenv
- ▶ ../bin/activate
- ▶ # install Django
- ▶ pip install django

NEW PROJECT

- ▶ `django-admin startproject myproject`
- ▶ `cd myproject`
- ▶ `./manage.py migrate`
- ▶ `./manage.py createsuperuser`
- ▶ `./manage.py runserver`