

Django

有誰在使用Django

- Pinterest
- Instagram
- Mozilla
- Disqus
- the Public Broadcasting Service

Why Django

- 歷史悠久：今年(2019)邁入第12年，經得起考驗。
- 市場佔有率最高
- 最完善的技術文件
- 豐富的套件
- 強大的社群
- 提供total solution，初學者可以透過Django內建的模塊了解網站運作。

Framework

Django



stackoverflow

126,394
questions tagged

[django](#) [about »](#)

github

	Repositories	78,271
	Code	
	Issues	202,995
	Wikis	13,343
	Users	558

特性

大而全，內建全套解決方案。

Flask



126,394
questions tagged

[django](#) [about »](#)

	Repositories	26,936
	Code	
	Issues	34,273
	Wikis	3,162
	Users	71

微框架，輕盈，簡單。

Tornado



2,542
questions tagged

[tornado](#) [about »](#)

	Repositories	3,763
	Code	
	Issues	14,981
	Wikis	1,826
	Users	262

微框架，少而精，non-blocking network IO。

Pyramid



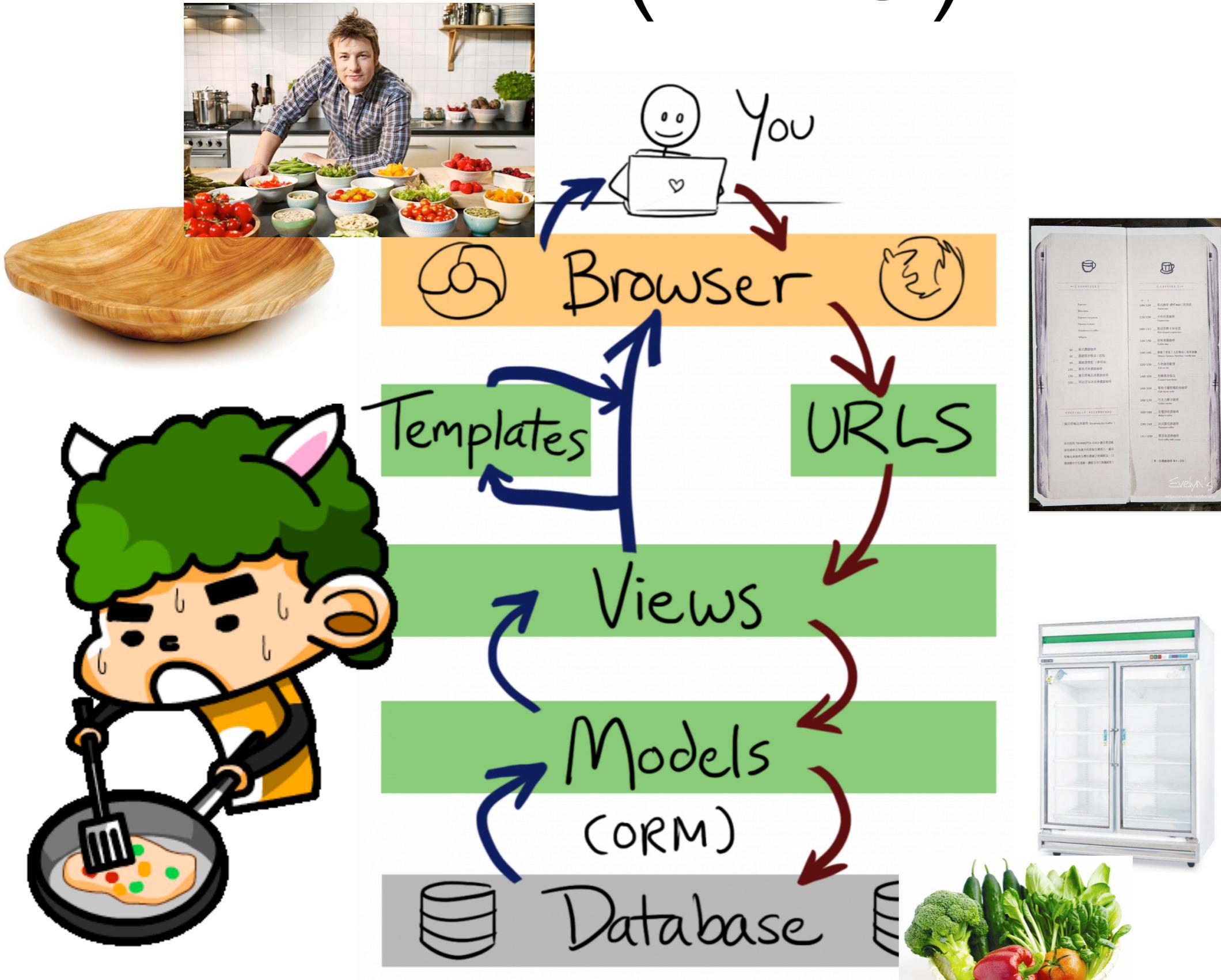
1,792
questions tagged

[pyramid](#) [about »](#)

	Repositories	2,733
	Code	
	Issues	12,311
	Wikis	1,589
	Users	129

注重靈活性，所有模塊都需另外找套件搭配運用。

MTV(MVC)



Why MVC?

```
<?php
session_start();
require_once('functions.php');
conectare();
if(isset($_POST['add']) && is_numeric($_POST['buc'])) {
    $qw=mysql_query("select * from cos where product_id='".$_POST['idprod']."' AND id_user='".$_SESSION['user']."' ");
    if(mysql_num_rows($qw)==0)
    {
        mysql_query("insert into cos ('product_id','bucati','id_user') values ('".$_POST['idprod']."','".$_POST['buc']."','".$_SESSION['user']."' )");
    }
}

} function valid_email($adresa) //verifica daca adresa de email este valida
{
    if (ereg('^[a-zA-Z0-9_.\-\+@\[a-zA-Z0-9\-\]+\.\[a-zA-Z0-9\-\.\]+\$', $adresa))
        return true;
    else
        return false;
}

if(isset($_SESSION['user']))
{
    $qwerty=mysql_query("select * from cos where id_user='".$_SESSION['user']."' ");
    $produc=mysql_num_rows($qwerty);
}
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml/DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
```

MySQL

PHP

HTML

Spaghetti Code



Why MVC?

- 分離商業邏輯與UI
- 前後端可獨立作業
- 擁有更多彈性
- 較容易維護
- 降低複雜度

models.py

```
<?php
session_start();
require_once('functions.php');
connectare();
if(isset($_POST['add']) && is_numeric($_POST['buc'])) {
    $qwe=mysql_query("select * from cos where product_id='".$_POST['idprod']."' AND id_user='".$_SESSION['user']."' ");
    if(mysql_num_rows($qwe)==0)
    {
        mysql_query("insert into cos ('product_id','bucati','id_user') values ('".$_POST['idprod']."','".$_POST['buc']."','".$_SESSION['user']."' )");
    }
}

function valid_email($adresa) //verifica daca adresa de email este valida
{
    if (ereg('^[a-zA-Z0-9_.\-\+@\[a-zA-Z0-9\-\]+\.\[a-zA-Z0-9\-\.\]+\$', $adresa))
        return true;
    else
        return false;
}

if(isset($_SESSION['user']))
{
    $qwert=mysql_query("select * from cos where id_user='".$_SESSION['user']."' ");
    $produce=mysql_num_rows($qwert);
}
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
</head>
```

MySQL

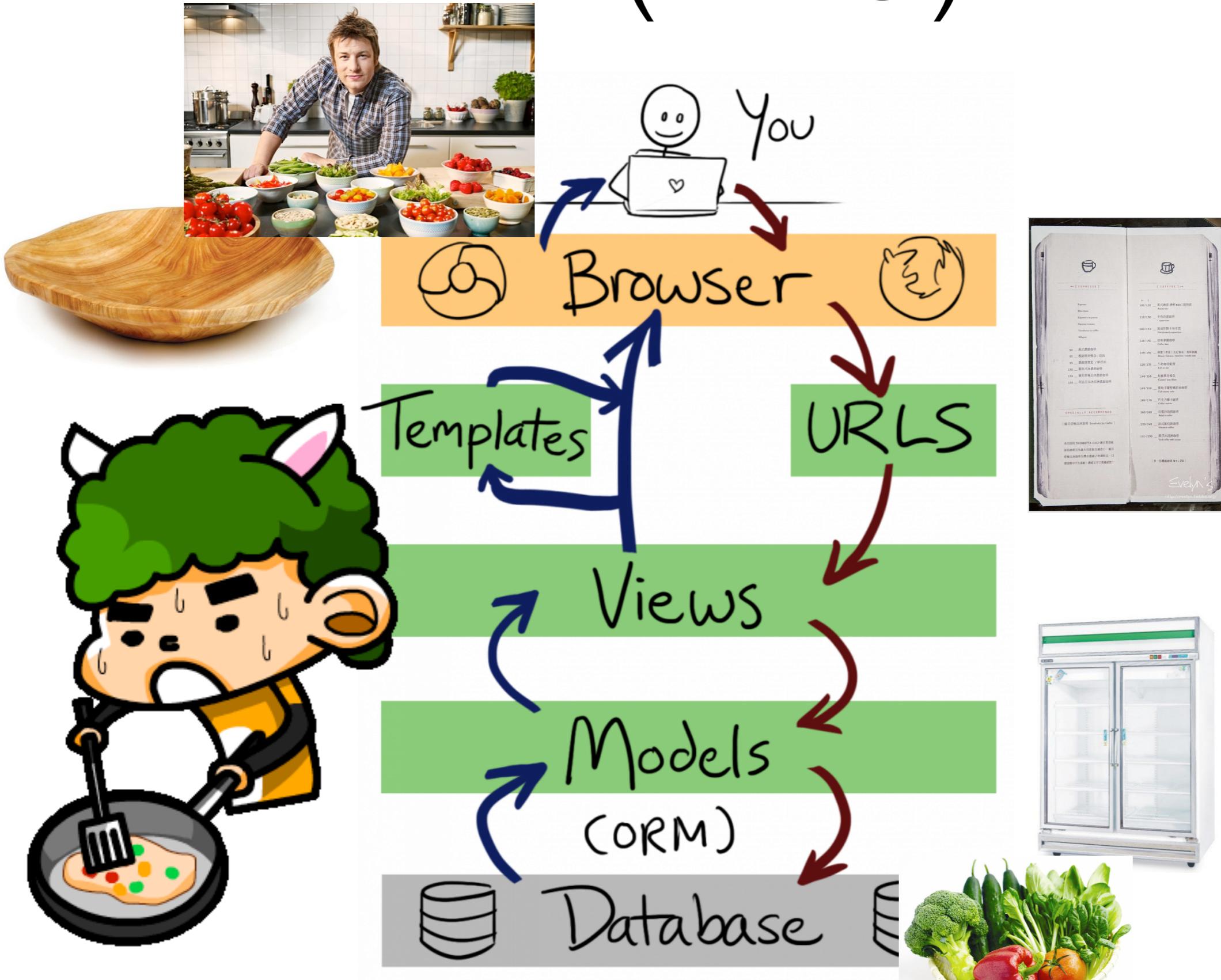
PHP

HTML

views.py

templates

MTV(MVC)



常見的類MVC架構

PHP: Zend Framework

Python: Django

Ruby: Ruby on Rails

.NET: ASP.NET MVC

...

安裝

```
pip install django  
pip freeze | grep Django
```

暖個身，

先來做個簡單的Blog吧

建置project

```
mkdir myproject
cd myproject
django-admin startproject project .
```

```
myproject
└── manage.py
└── project
    ├── settings.py
    ├── urls.py
    ├── wsgi.py
    └── __init__.py
```

設定資料庫

- myproject/project/settings.py

```
DATABASES = {  
    'default': {  
        'ENGINE': 'django.db.backends.sqlite3',  
        'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),  
    }  
}
```

- myproject/
python manage.py migrate



db.sqlite3

測試server

- myproject/

python manage.py runserver 8000

輸入網址：127.0.0.1:8000

[django](#) [View release notes for Django 2.0](#)



The install worked successfully! Congratulations!

You are seeing this page because `DEBUG=True` is in
your settings file and you have not configured any
URLs.



[Django Documentation](#)
Topics, references, & how-to's



[Tutorial: A Polling App](#)
Get started with Django



[Django Community](#)
Connect, get help, or contribute

建置app

- myproject/

python manage.py startapp blog

myproject

 └── db.sqlite3

 └── project

 |
 | └── __init__.py

 |
 | └── settings.py

 |
 | └── urls.py

 |
 | └── wsgi.py

 └── manage.py

 └── blog

 └── migrations

 |
 | └── __init__.py

 |
 | └── __init__.py

 |
 | └── admin.py

 |
 | └── models.py

 |
 | └── tests.py

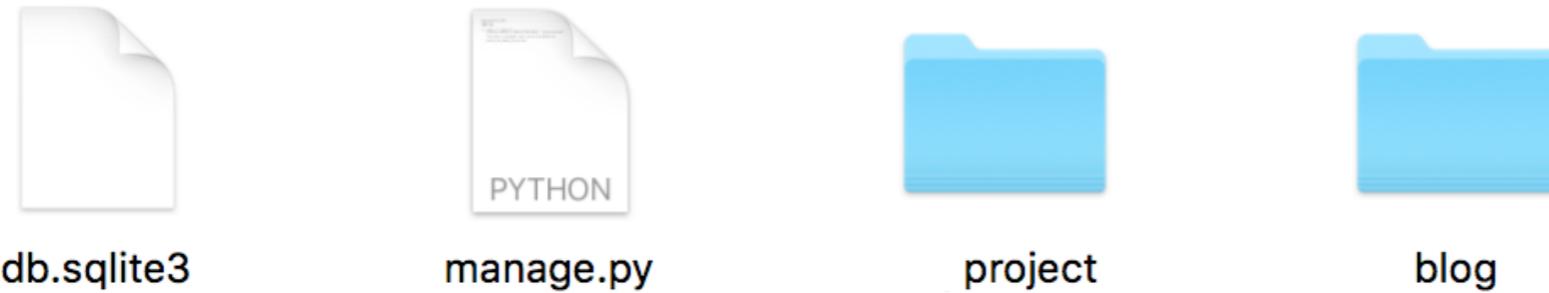
 └── views.py

建置app

- myproject/project/settings.py

```
INSTALLED_APPS = (
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'blog',
)
```

建置app



兩層架構：

- project (project)
- app (blog)

可只用project層就做完網站。

app層可以多個。

app層讓架構分明以外，可作為可拔插的元件，
供任何project使用。

後台管理介面

- myproject/

python manage.py createsuperuser

設定user, password,e-mail

輸入網址 : 127.0.0.1:8000/admin/

The screenshot shows the Django administration interface. At the top, there's a blue header bar with the text "Django administration". Below it, a white header bar says "Site administration". Underneath, there's a light blue sidebar header "Authentication and Authorization" containing two items: "Groups" and "Users". To the right of these items are two buttons: a green "Add" button with a plus sign and a blue "Change" button with a pencil icon. On the far right, there's a sidebar titled "Recent Actions" which is currently empty, followed by another sidebar titled "My Actions" which also says "None available".

ADDING MODEL

- myproject/blog/models.py

```
from django.db import models
from django.utils import timezone

class Post(models.Model):
    author = models.ForeignKey('auth.User',
        on_delete = models.CASCADE,
        null = False)
    title = models.CharField(max_length=200)
    text = models.TextField()
    created_date = models.DateTimeField(
        default=timezone.now)

    def publish(self):
        self.created_date = timezone.now()
        self.save()

    def __str__(self):
        return self.title
```

MODEL MIGRATION

- myproject/

創建完model之後，要migrate才會建立資料庫內容

step 1. 創建migrations腳本檔案：

python manage.py makemigrations

- myproject/blog/migrations新增了0001_initial.py
 - migrations檔案名稱結構為：(編號)_(執行動作).py
- 可將此步驟視為創建了一個”建立資料庫內容”的腳本

step 2. 創建資料庫：

python manage.py migrate

- 執行migrations腳本，建立資料庫內容

註冊MODEL

之後可於後台管理介面查看資料

- myproject/blog/admin.py

```
from django.contrib import admin  
from .models import Post  
  
admin.site.register(Post)
```

<http://127.0.0.1:8000/admin/>

The screenshot shows the Django administration interface. At the top, there's a blue header bar with the text "Django administration". Below it, a white page titled "Site administration". A sidebar on the left lists several models: "Authentication and Authorization" (Groups, Users), "Webapp" (Posts), and "Posts". The "Webapp" section is highlighted with a red oval around its "Posts" entry. Each entry in the sidebar has two buttons: a green "Add" button and a pink "Change" button.

於後台新增資料進資料庫

Django administration

Welcome, jojotenya. View site / Change password / Log out

Home > Blog > Posts > Django首發!

Change post

Author: ZB103

Title: Django首發!

Text:
Django到底要怎麼唸啊?
Dee – jango?
Dj – ango??

Created date: Date: 2015-11-08 Today |
Time: 03:08:32 Now |
Note: You are 8 hours ahead of server time.

Published date: Date: Today |
Time: Now |
Note: You are 8 hours ahead of server time.

URL DISPATCHER-1

- myproject/project/urls.py (project 層級的urls.py)

```
from django.contrib import admin
from django.urls import path, include

urlpatterns = [
    path('admin/', admin.site.urls),
    path('blog/', include('blog.urls')),
]
```

URL DISPATCHER-2

- myproject/blog/urls.py (app 層級的urls.py)

```
from django.urls import path
from . import views

urlpatterns = [
    path("", views.post_list, name='post_list'),
]
```

URL DISPATCHER-3

url的位置代表的是執行哪一支View

VIEW

- myproject/blog/views.py

```
from django.shortcuts import render

def post_list(request):
    return render(request, 'blog/post_list.html', {})
```

TEMPLATE

- blog/templates/blog/post_list.html (首先直接寫入資料看看)

```
<html>
  <head>
    <title>III's blog</title>
  </head>
  <body>
    <div>
      <h1><a href="">III's Blog</a></h1>
    </div>

    <div>
      <p>發佈時間: 10.11.2015, 10:11</p>
      <h2><a href="">Django第一天</a></h2>
      <p>今天來了一個代課阿豪，講的課都聽不是很懂，還扯到什麼開餐廳~!@#$%^WTF</p>
    </div>

    <div>
      <p>發佈時間: 11.11.2015, 11:11</p>
      <h2><a href="">Django第二天</a></h2>
      <p>發現阿豪用心良苦，現在Django對我來是簡直是piece of cake ^.<，準備要請客吃飯了</p>
    </div>
  </body>
</html>
```

網頁出現了，好感動

127.0.0.1:8000/blog

The screenshot shows a web browser window with the URL '127.0.0.1:8000/blog/' in the address bar. The browser's toolbar includes icons for back, forward, refresh, and home, along with a folder icon for the current page. Below the toolbar is a bookmarks bar with links to '應用程式', 'ruby', 'big data', '台股觀測站', 'Google Calendar', 'Oxford University Pre', and 'Facebook'. The main content area displays a blog post titled 'ZB103's Blog' in blue. Below the title, the text '發佈時間: 10.11.2015, 10:11' is shown. The first blog entry is titled 'Django第一天' in purple. The text of the entry reads: '今天來了一個代課阿豪，講的課都聽不是很懂，還扯到什麼開餐廳~!@#\$%^WTF'. Below this entry, the text '發佈時間: 11.11.2015, 11:11' is displayed. The second blog entry is titled 'Django第二天' in purple. The text of this entry reads: '發現阿豪用心良苦，現在Django對我來是簡直是piece of cake ^.<'. The browser has a light gray background and a white header bar.

用動態資料做做看

models.py

```
class Post(models.Model):
    author = models.ForeignKey('auth.User',
        on_delete=models.CASCADE,null=False)
    title = models.CharField(max_length=200)
    text = models.TextField()
    created_date = models.DateTimeField(
        default=timezone.now)
```

views.py

```
from django.shortcuts import render
from .models import Post

def post_list(request):
    posts = Post.objects.all().order_by('-created_date')
    return render(request, 'blog/post_list.html', {'posts': posts})
```

Django administration

Home > Blog > Posts > Django首發!

Change post

Author: ZB103 

Title: Django首發!

Text:
Django到底要怎麼唸啊 ?
Dee - jango?
Dj- ango??

Created date: Date: 2015-11-08 Today | 

Time: 03:08:32 Now | 

Note: You are 8 hours ahead of serv

Published date: Date: Today | 

Time: Now | 

Note: You are 8 hours ahead of serv

 Delete

向資料庫下Query

SQL	Django
<code>select * from blog_post</code>	<code>Post.objects.all()</code>
<code>select * from blog_post where title= 'Django首發!'</code>	單筆： <code>Post.objects.get(id=1)</code> 多筆： <code>Post.objects.filter(title= 'Django首發!')</code>
<code>select * from blog_post order by created_date desc</code>	<code>Post.objects.all().order_by ('-created_date')</code>
<code>select * from blog_post where created_date <= date('now')</code>	<code>Post.objects.filter(created_date__lt e=timezone.now())</code>
<code>insert into blog_post(author,title,text) value('III','test', 'test')</code>	<code>Post.objects.create(author=me, title='test', text='test')</code>

- 註1. Django 會自動把models裡面的物件前面加上"app名稱_"當做資料庫的table名稱，如果想要用自己設定的table名稱，請見附錄。
- 註2. Django下query採用ORM，其中"objects"為query管理器，為model物件自帶的參數。
- 註3. 右邊author=me的me為`django.contrib.auth.models.User.objects.get(username='III')`

動態資料塞入template

- myproject/blog/templates/blog/post_list.html

post_list.html

```
<html>
  <head>
    <title>III's blog</title>
  </head>
  <body>
    <div>
      <h1><a href="/blog">III's Blog</a></h1>
    </div>

    <div>
      {%
        for post in posts %}
        <p>發佈時間: {{post.created_date}}</p>
        <h2><a href=>{{post.title}}</a></h2>
        <p>{{post.text | linebreaks}}</p>
      {%
        endfor %}
    </div>

  </body>
</html>
```

- myproject/blog/views.py

views.py

```
from django.shortcuts import render
from .models import Post
```

```
def post_list(request):
    posts = Post.objects.all()
                  .order_by('-created_date')
    return render(request, 'blog/post_list.html', {'posts': posts})
```

網頁出現惹，但醜醜der...



發佈時間: Nov. 8, 2015, 3:08 a.m.

Django首發!

Django到底要怎麼唸啊？

Dee - jango?

Dj- ango??

發佈時間: Nov. 8, 2015, 5:48 a.m.

Django second post!

Django is not difficult la!

美化網頁神奇寶典

- 資策會王牌美女老師：倪老師動態網頁
聖經
- google字型



BOOTSTRAP

<http://getbootstrap.com/>

- Bootstrap CDN:

```
<!-- Latest compiled and minified CSS -->
```

```
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
```

```
<!-- Optional theme -->
```

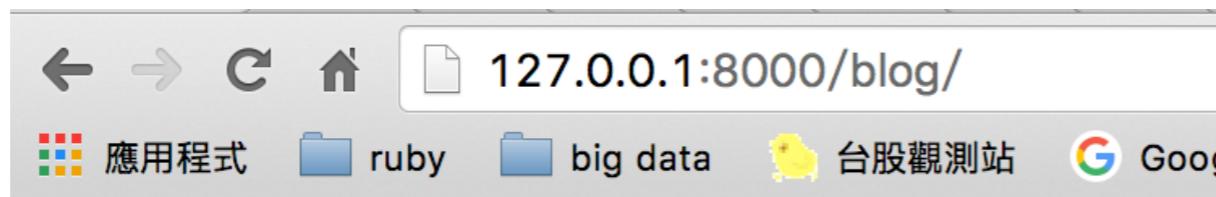
```
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap-theme.min.css">
```

```
<!-- Latest compiled and minified JavaScript -->
```

```
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js" ></script>
```

註：前兩個放在post_list.html的<head></head>裡面，最後一個要用到javascript才需要放，置於<body></body>的最後一行

BOOTSTRAP的效果



ZB103's Blog

發佈時間: Nov. 8, 2015, 3:08 a.m.

Django首發!

Django到底要怎麼唸啊？

Dee - jango?

Dj- ango??

發佈時間: Nov. 8, 2015, 5:48 a.m.

Django second post!

Django is not difficult la!

註：可上<http://getbootstrap.com/getting-started/#examples>觀看一些範例

google字型

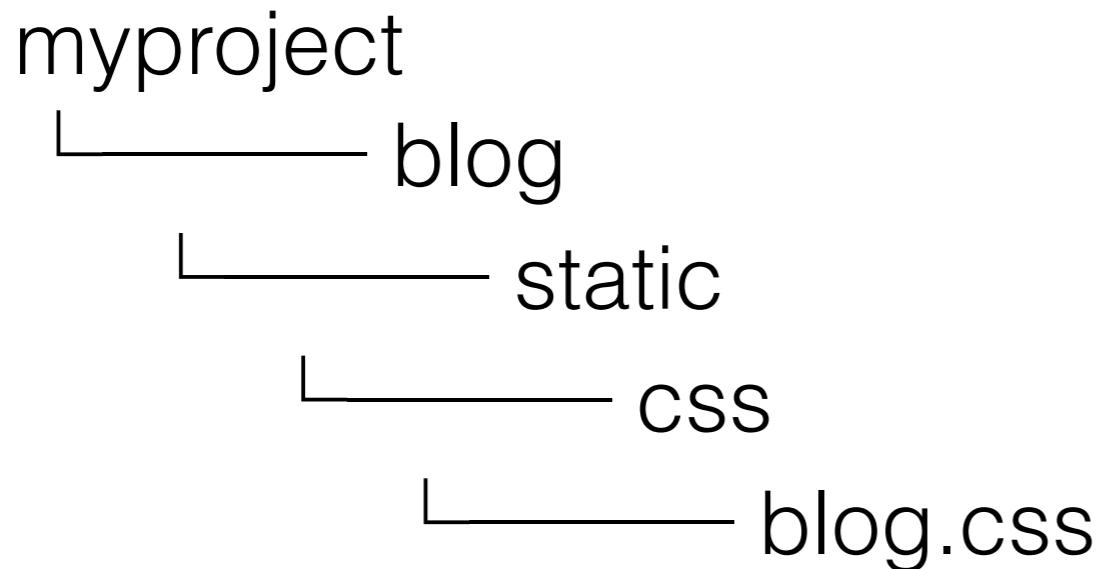
<https://www.google.com/fonts>

- google font CDN:

```
<link href="http://fonts.googleapis.com/css?family=Lobster&subset=latin,latin-ext"  
rel="stylesheet" type="text/css">
```

客製化的CSS

- 於app(blog)底下新增static資料夾，並於static底下新增css資料夾以放置css檔。



- Django會自動偵測app底下名為static的資料夾，並將資料夾路徑存為變量"static"，以供template載入使用。
- static資料夾可存放其他靜態檔，例如javascript檔、圖檔以供template存取。

套用static資料至TEMPLATE

- myproject/blog/templates/blog/post_list.html

```
{% load staticfiles %}  
<html>  
  <head>  
    <title>III's blog</title>  
    <link rel="stylesheet" href="//maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/  
bootstrap.min.css">  
    <!--<link rel="stylesheet" href ="/static/css/blog.css">-->  
    <link rel="stylesheet" href="{% static 'css/blog.css' %}">  
  </head>
```

註1. 檔案最開頭要寫入{% load staticfiles %}才可以存取static路徑變數

註2. 此方法好處為不管未來static資料夾如何變動都不需要擔心路徑的問題。

TEMPLATE

```
{% load staticfiles %}

<html>
  <head>
    <title>lii's blog</title>
    <link rel="stylesheet" href="//maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
    <link href="http://fonts.googleapis.com/css?family=Lobster&subset=latin,latin-ext" rel="stylesheet" type="text/css">
    <link rel="stylesheet" href="{% static 'css/blog.css' %}">
    <!--<link rel="stylesheet" href ="/static/css/blog.css">-->
  </head>
  <body>
    <div class="page-header">
      <h1><a href="/blog">III's Blog</a></h1>
    </div>
    <div class="content container">
      <div class="row">
        <div class="col-md-8">
          {% for post in posts %}
            <div class="post">
              <div class="date">
                <p>發佈時間: {{ post.created_date }}</p>
              </div>
              <h1><a href="/blog/{{post.id}}"/>{{ post.title }}</a></h1>
              <p>{{ post.text | linebreaks }}</p>
            </div>
          {% endfor %}
        </div>
      </div>
    </div>

  </body>
</html>
```

CSS檔範例

- myproject/blog/static/css/blog.css

```
body {  
    padding-left: 15px;  
}  
h1 a {  
    color: #FCA205;  
    font-family: 'lobster';  
}  
.page-header {  
    background-color: #3AC3E3;  
    margin-top: 0;  
    padding: 20px 20px 20px 40px;  
}  
.page-header h1, .page-header h1 a, .page-header h1 a:visited, .page-header h1 a:active {  
    color: #ffffff;  
    font-size: 36pt;  
    text-decoration: none;  
}  
.content {  
    margin-left: 40px;  
}  
h1, h2, h3, h4 {  
    font-family: 'lobster', cursive;  
}  
.date {  
    float: right;  
    color: #828282;  
}  
.post {  
    margin-bottom: 70px;  
}  
  
.post h1 a, .post h1 a:visited {  
    color: #000000;  
}
```

好像有好看一點

ZB103's Blog

Django首發！

Django到底要怎麼唸啊？

Dee - jango?

Dj- ango??

發佈時間: Nov. 8, 2015, 3:08 a.m.

Django second post!

Django is not difficult la!

發佈時間: Nov. 8, 2015, 5:48 a.m.

加入表單-1

- myproject/blog/forms.py

```
from django import forms

class PostForm(forms.Form):
    title = forms.CharField(max_length=100)
    text = forms.CharField(max_length=2000, widget = forms.Textarea())
```

加入表單-2

- myproject/blog/templates/blog/post_list.html

```
...
<div class="row">
  <div class="col-md-8">
    <h4>發佈新文章:</h4>
  </div>
  <div class="col-md-8">
    <form action="/blog/add_record" method="post" accept-charset="utf-8">
      <table>
        {{post_form.as_table}}
      </table>
      <input type="submit" name="add_record" value="提交">
    </form>
  </div>
</div>
...
```

加入表單-3

127.0.0.1:8000/blog/

發佈新文章：

Title:

Text:

提交

加入表單-4

- myproject/blog/urls.py

```
from django.urls import include, path
from . import views

urlpatterns = [
    path('add_record', views.add_record, name='add_record'),
    path("", views.post_list, name='post_list'),
]
```

加入表單-5

- myproject/blog/views.py

```
...
from .forms import PostForm
from django.contrib.auth import get_user_model
from django.shortcuts import redirect
me = get_user_model().objects.get(username='username')

...
def add_record(request):
    if request.POST:
        title = request.POST['title']
        text = request.POST['text']
        Post.objects.create(author=me, title=title, text=text)
    return redirect('/blog')
```

CSRF(Cross-Site Request Forgery)

127.0.0.1:8000/blog/

Forbidden (403)

CSRF verification failed. Request aborted.

Help

Reason given for failure:

CSRF token missing or incorrect.

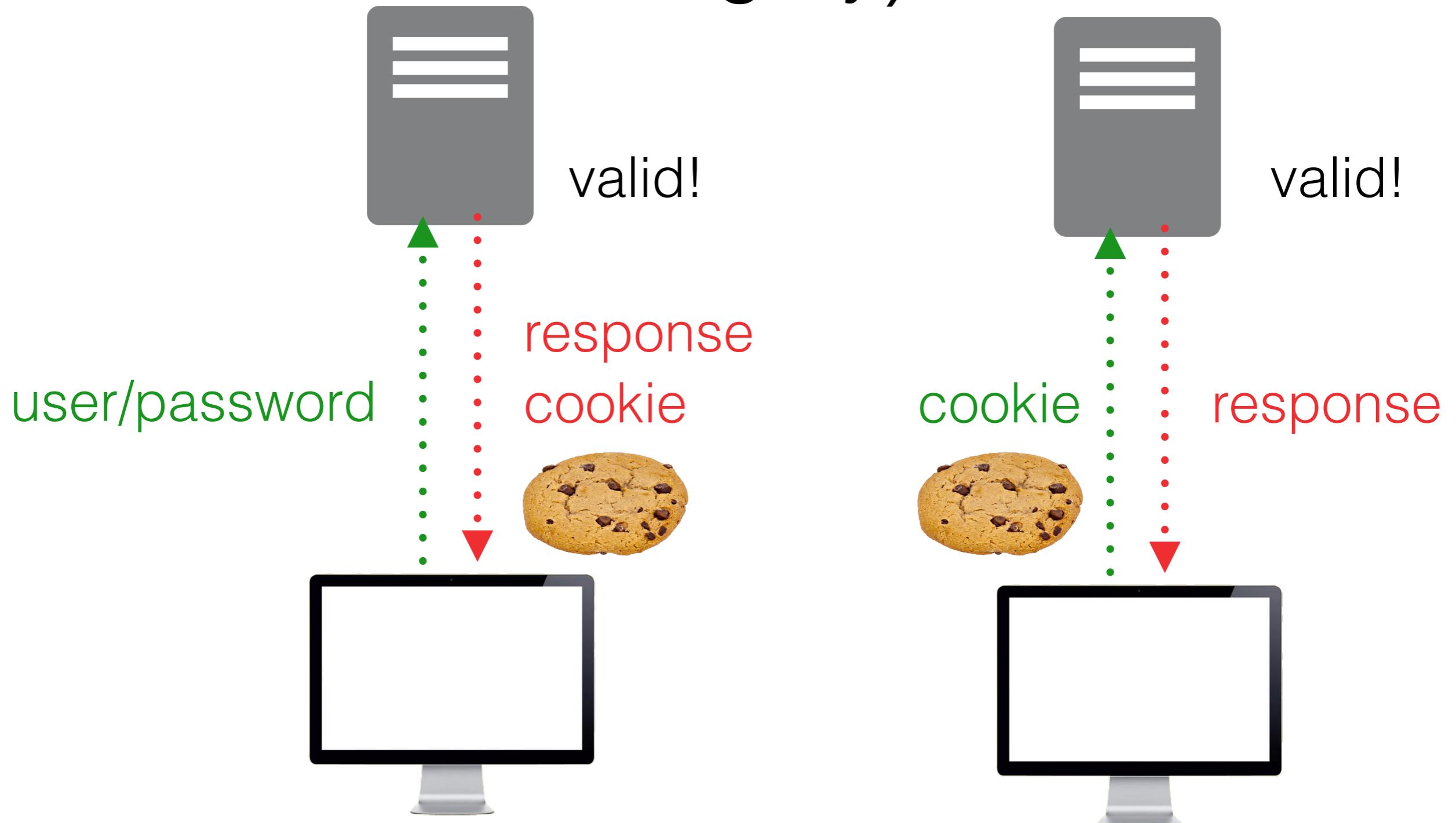
In general, this can occur when there is a genuine Cross Site Request Forgery, or when:

- Your browser is accepting cookies.
- The view function passes a `request` to the template's [render](#) method.
- In the template, there is a `{% csrf_token %}` template tag inside each POST form.
- If you are not using `CsrfViewMiddleware`, then you must use `csrf_protect` on your POST data.

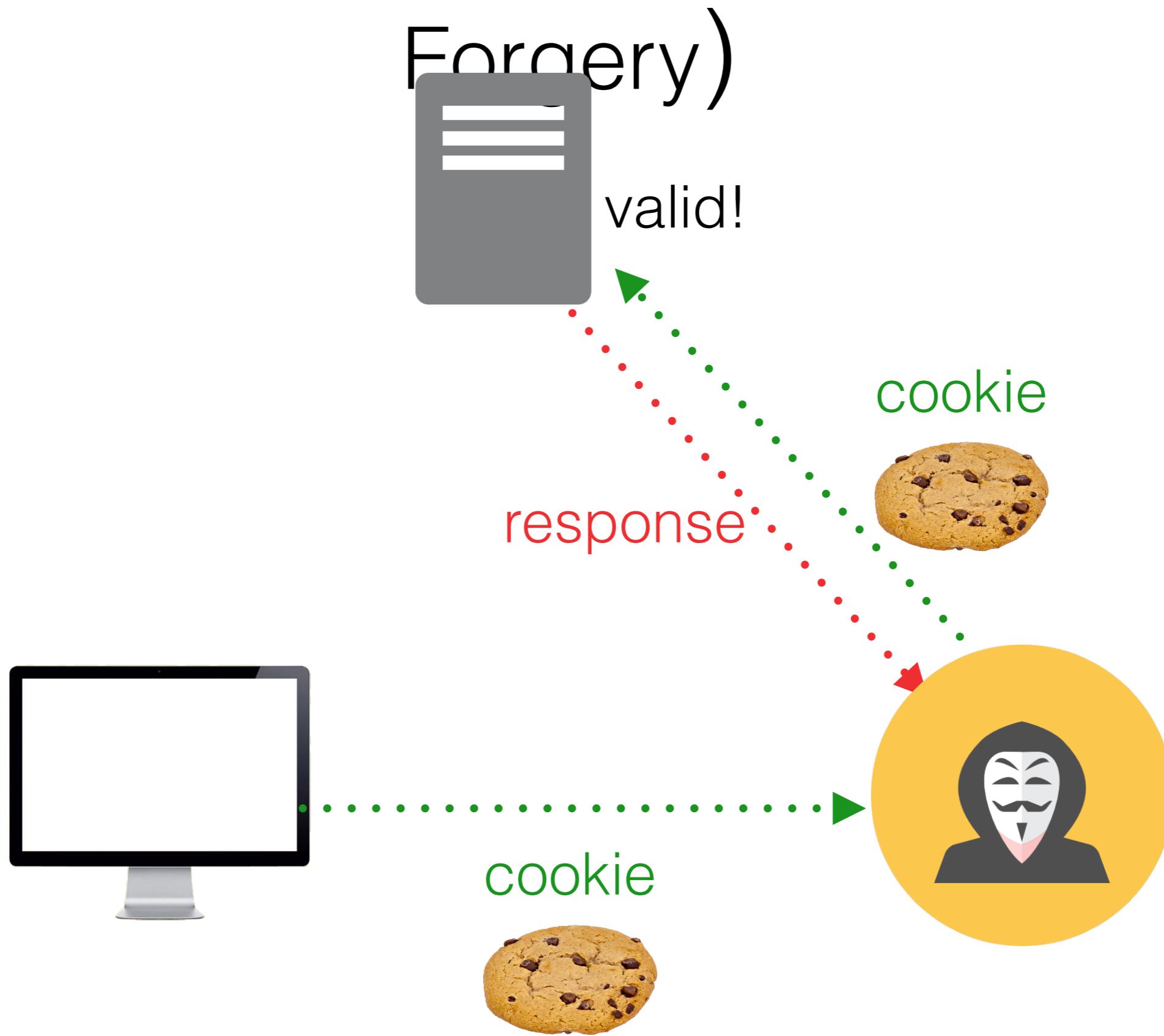
You're seeing the help section of this page because you have `DEBUG = True` in your settings.

You can customize this page using the `CSRF_FAILURE_VIEW` setting.

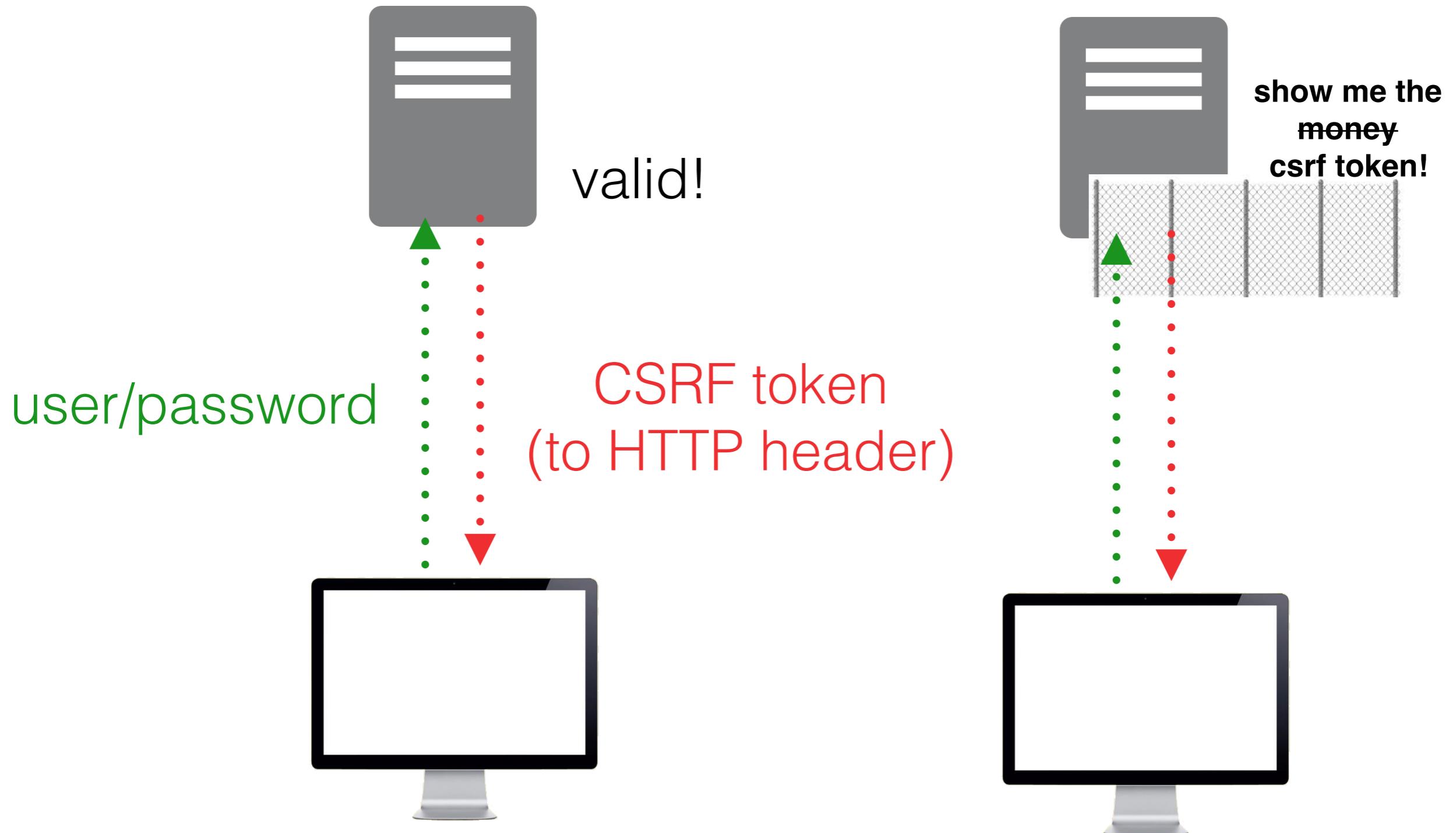
CSRF(Cross-Site Request Forgery)



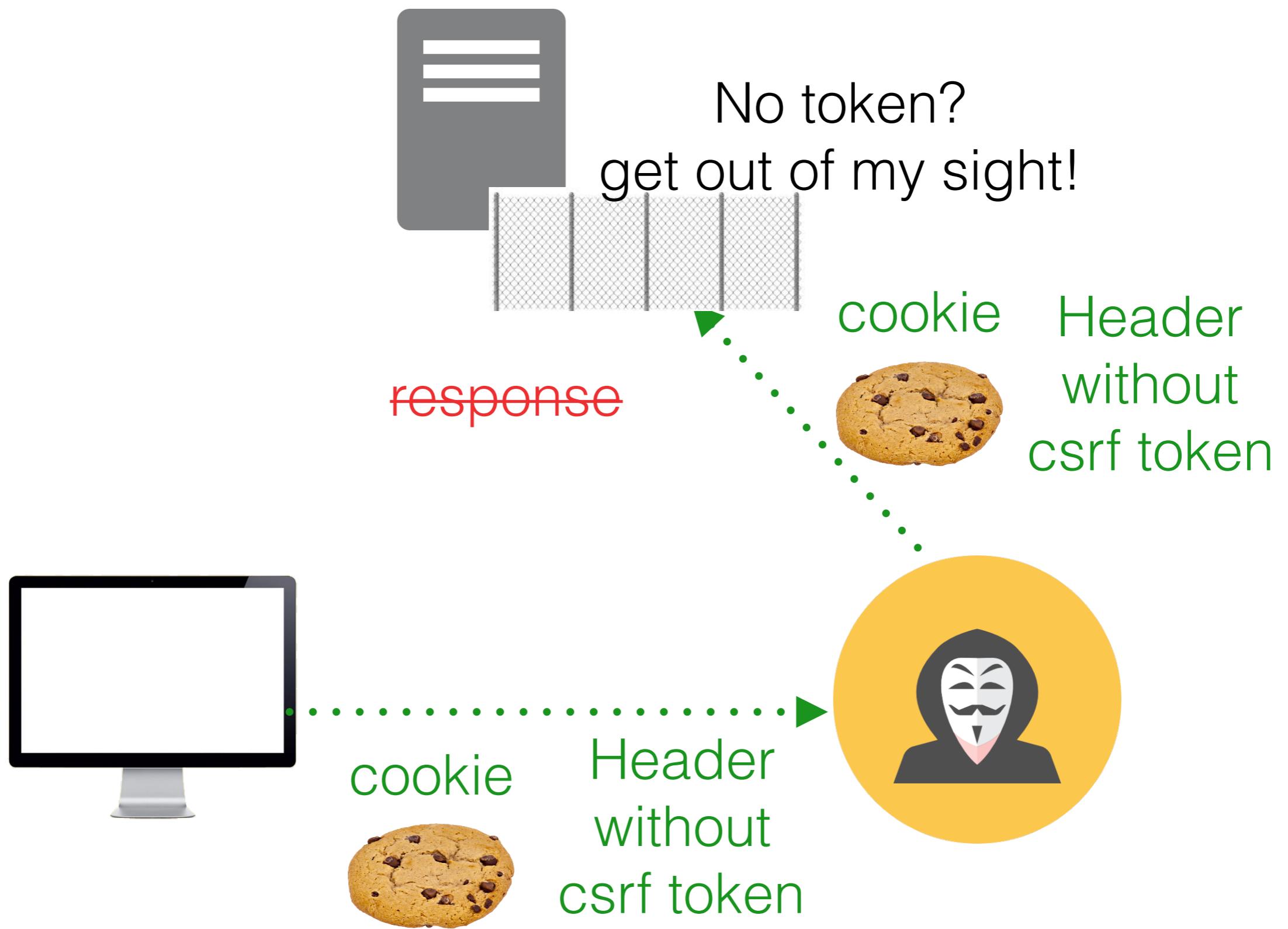
CSRF(Cross-Site Request Forgery)



Avoid CSRF



Avoid CSRF



CSRF_TOKEN

- myproject/blog/templates/blog/post_list.html

```
...
<div class="row">
  <div class="col-md-8">
    <h4>發佈新文章:</h4>
  </div>
  <div class="col-md-8">
    <form action="/blog/add_record" method="post" accept-charset="utf-8">
      {% csrf_token %}
      <table>
        {{post_form.as_table}}
      </table>
      <input type="submit" name="add_record" value="提交">
    </form>
  </div>
</div>
...
```

CSRF_EXEMPT

- myproject/blog/views.py

```
...
from .forms import PostForm
from django.contrib.auth import get_user_model
from django.shortcuts import redirect
from django.views.decorators.csrf import csrf_exempt
me = get_user_model().objects.get(username='III')

...
@csrf_exempt
def add_record(request):
    if request.POST:
        title = request.POST['title'].encode('utf-8')
        text = request.POST['text'].encode('utf-8')
        newpost = Post.objects.create(author=me, title=title, text=text)
    return redirect('/blog')
```

表單發送成功

發佈新文章:

Title: 第三篇

看看會不會成功...

Text:

提交

選擇發佈記錄:

Django首發!

Django首發!

發佈時間: Nov. 8, 2010

Django到底要怎麼唸啊 ?
Dee - jango?
Dj- ango??

Django second post!

發佈時間: Nov. 8, 2010

Django is not difficult la!

第三篇

發佈時間: Nov. 8, 2010

看看會不會成功...

用url跨頁傳遞資料-1

- myproject/blog/urls.py

```
from django.urls import path
from . import views

urlpatterns = [
    path('add_record', views.add_record, name='add_record'),
    path('<int:id>', views.post_record, name='post_record'),
    path("", views.post_list, name='post_list'),
]
```

用url跨頁傳遞資料-2

- myproject/blog/views.py

```
def post_record(request,id):  
    ...  
    post = Post.objects.get(id = id)  
    return render(request, 'blog/post_record.html', locals())  
    ...
```

用url跨頁傳遞資料-3

- myproject/blog/templates/blog/post_record.html

```
{% load staticfiles %}  
<html>  
  <head>  
    <title>III's blog</title>  
    <link rel="stylesheet" href="//maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">  
    <link href="http://fonts.googleapis.com/css?family=Lobster&subset=latin,latin-ext" rel="stylesheet" type="text/css">  
    <link rel="stylesheet" href="{% static 'css/blog.css' %}">  
  </head>  
  <body>  
    <div class="container">  
  
      <div class="blog-header">  
        <h1 class="blog-post-title">{{ post.title }}</h1>  
      </div>  
      <div class="row">  
        <div class="col-sm-8 blog-main">  
          <div class="blog-post">  
            <p>  
              {{ post.text }}  
            </p>  
            <p class="blog-post-meta">{{ post.created_date }} by <a href="#">{{ post.author }}</a></p>  
            <a href="/blog/">回首頁</a>  
          </div>  
        </div>  
      </div>  
    </div>  
  
  </body>  
</html>
```

成功換頁



Django首發！

Django到底要怎麼唸啊？Dee - jango? Dj- ango??

Nov. 8, 2015, 3:08 a.m. by [ZB103](#)

[回首頁](#)

分頁也想共用主頁版面

模版進階應用：繼承-1

建立 **base.html**

```
{% load staticfiles %}  
<html>  
  <head>  
    <title>III's blog</title>  
    <link rel="stylesheet" href="//maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">  
    <link href="http://fonts.googleapis.com/css?family=Lobster&subset=latin,latin-ext" rel="stylesheet" type="text/css">  
    <link rel="stylesheet" href="{% static 'css/blog.css' %}">  
  </head>  
  <body>  
    <div class="page-header">  
      <h1><a href="/blog">III's Blog</a></h1>  
    </div>  
    {% block content %}{% endblock %}  
  </body>  
</html>
```

模版進階應用：繼承-2

修改 `post_record.html`:

```
{% extends 'blog/base.html' %}  
{% block content %}  
<div class="container">  
  <div class="blog-header">  
    <h1 id="blog-post-title" class="blog-post-title">{{ post.title }}</h1>  
  </div>  
  <div class="row">  
    <div class="col-sm-8 blog-main">  
      <div id="blog-post" class="blog-post">  
        <p>  
          {{ post.text }}  
        </p>  
        <p class="blog-post-meta">{{ post.created_date }} by <a href="#">{{ post.author }}</a></p>  
        <a href="/blog/">回首頁</a>  
        <p id="current_page"> {{post.id}} </p>  
      </div>  
    </div>  
  </div>  
</div>  
{% endblock %}
```

模版進階應用：繼承-3

修改 post_list.html:

```
{% extends 'blog/base.html' %}  
{% block content %}  
<div class="content container">  
    <div class="row">  
        <div class="col-md-8">  
            {% for post in posts %}  
                <div class="post">  
                    <div class="date">  
                        <p>發佈時間: {{ post.created_date }}</p>  
                    </div>  
                    <!--<h1><a href="/blog/{{post.id}}"/>{{ post.title }}</a></h1>-->  
                    <h1><a href="{% url 'post_record' post.id %}">{{ post.title }}</a></h1>  
                    <p>{{ post.text|linebreaks }}</p>  
                </div>  
            {% endfor %}  
        </div>  
    </div>  
    ....(略)....  
</div>  
  
{% endblock %}
```

ZB104's Blog

Django首發！

Django到底要怎麼唸啊？Dee - jango? Dj- ango??

Nov. 8, 2015, 3:08 a.m. by ZB103

[回首頁](#)

1

[下一頁](#)

總複習

urls.py

```
from django.conf.urls import include, url
from . import views

urlpatterns = [
    url(r'^add_record', views.add_record, name='add_record'),
    url(r'^(?P<id>\d+)', views.post_record, name='post_record'),
    url(r'^$', views.post_list, name='post_list'),
]
```

models.py

```
class Post(models.Model):
    author = models.ForeignKey('auth.User')
    title = models.CharField(max_length=200)
    text = models.TextField()
    created_date = models.DateTimeField(
        default=timezone.now)
    published_date = models.DateTimeField(
        blank=True, null=True)

    def publish(self, title, text):
        self.author = me
        self.title = title
        self.text = text
        self.created_date = timezone.now()
        self.published_date = timezone.now()
        self.save()

    def __unicode__(self):
```

views.py

```
def post_list(request):
    posts = Post.objects.filter(created_date__lte=timezone.now()).order_by('published_date')
    select_forms = SelectForm()
    post_form = PostForm()
    return render(request, 'blog/post_list.html', locals())

def post_record(request, id):
    post = Post.objects.get(id = id)
    return render(request, 'blog/post_record.html', locals())

@csrf_exempt
def add_record(request):
    if request.POST:
        title = request.POST['title'].encode('utf-8')
        text = request.POST['text'].encode('utf-8')
        newpost = Post.objects.create(author=me, title=title, text=text)
    return redirect('/blog')
```

template

```
<html>
    <head>
        <title>ZB103's blog</title>
        <link rel="stylesheet" href="//maxcdn.bootstrapcdn.com/bootstrap/3.3.5/css/bootstrap.min.css">
            <link rel="stylesheet" href="//maxcdn.bootstrapcdn.com/bootstrap/3.3.5/css/bootstrap-theme.min.css">
                <link href="http://fonts.googleapis.com/css?family=Lobster&subset=latin,latin-ext" rel="stylesheet" type="text/css">
                    <link rel="stylesheet" href="{% static "css/blog.css" %}">
    </head>
    <body>
        <div class="page-header">
            <h1><a href="">ZB103's Blog</a></h1>
        </div>
        <div class="content container">
            <div class="row">
                <div class="col-md-8">
```

Deploy-1

- <https://ngrok.com>

The screenshot shows the official ngrok website. At the top, there's a navigation bar with links for "HOW IT WORKS", "PRICING", "DOWNLOAD", "DOCS", "ENTERPRISE SOLUTIONS", "LOGIN", and a prominent orange "SIGN UP →" button. Below the navigation is a large image of a computer screen. On the left side of the screen is a browser window showing a placeholder page for "katesapp.ngrok.io" with the message "Welcome to Kate's Site! It's currently under development...". On the right side of the screen is a terminal window showing the command `$./ngrok http 3000` and its output: "ngrok by @inconshreveable". Below the terminal, there's a table of session details:

Session Status	online
Account	Kate Libby (Plan: Pro)
Web Interface	http://127.0.0.1:4040
Forwarding	http://katesapp.ngrok.io -> localhost:3000
Forwarding	https://katesapp.ngrok.io -> localhost:3000

To the right of the screenshot, there's a section with the heading "Public URLs for demoing from your own machine." followed by a descriptive text: "Spend more time programming. One command for an instant, secure URL to your localhost server through any NAT or firewall." Below this is an orange "Get started for free →" button. At the bottom of the page, there are several social media and technology logos: slack, GitHub, SendGrid, twilio, ATlassian, and Amazon Web Services.

As well as [Amazon Web Services](#) and many more.

Deploy-2

- <https://ngrok.com>

1 Download ngrok

ngrok is easy to install. Download a single binary with zero run-time dependencies.

 [Download for Mac OS X](#)

[Windows](#) [Linux](#) [Mac \(32-Bit\)](#) [Windows \(32-Bit\)](#)

[Linux \(ARM\)](#) [Linux \(32-Bit\)](#) [FreeBSD \(64-Bit\)](#)

[FreeBSD \(32-Bit\)](#)

2 Unzip to install

On Linux or OSX you can unzip ngrok from a terminal with the following command. On Windows, just double click ngrok.zip.

`$ unzip /path/to/ngrok.zip`

Most people keep ngrok in their user folder or set an alias for easy access.

3 Connect your account

Running this command will add your account's authtoken to your ngrok.yml file. This will give you more features and all open tunnels will be listed here in the dashboard.

`$./ngrok authtoken 3e..._6`

4 Fire it up

Read [the documentation](#) on how to use ngrok. Try it out by running it from the command line:

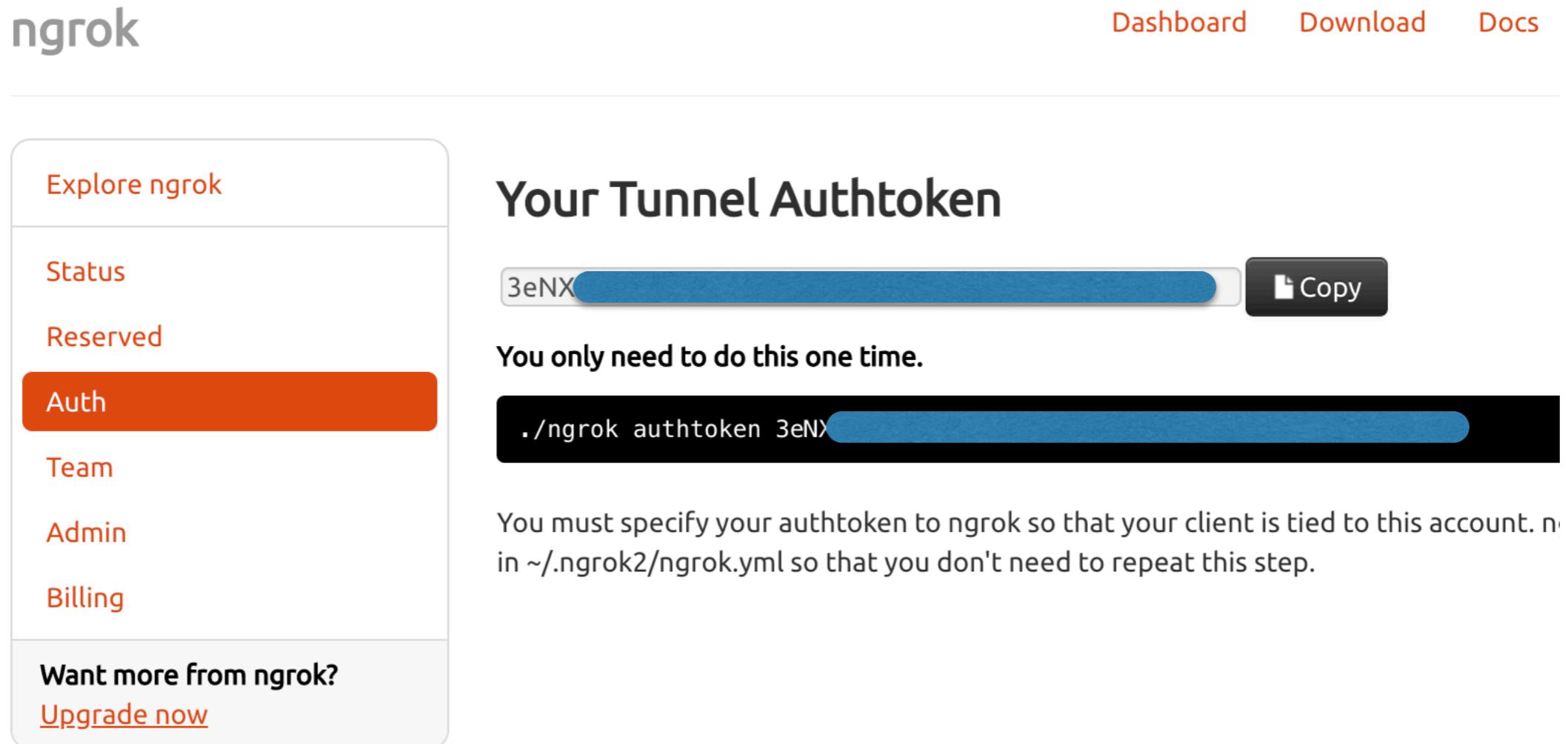
`$./ngrok help`

To start a HTTP tunnel on port 80, run this next:

`$./ngrok http 80`

Deploy-3

- <https://ngrok.com>



The screenshot shows the ngrok web interface. The top navigation bar includes links for 'Dashboard', 'Download', and 'Docs'. On the left, a sidebar menu lists 'Explore ngrok', 'Status' (which is 'Reserved'), 'Auth' (which is highlighted in orange), 'Team', 'Admin', and 'Billing'. Below the sidebar, a call-to-action button says 'Want more from ngrok? [Upgrade now](#)'. The main content area is titled 'Your Tunnel Authtoken' and displays a single-line auth token: '3eNX'. A 'Copy' button is available to copy this token to the clipboard. A note below the token states: 'You only need to do this one time.' followed by a command line example: '. /ngrok authtoken 3eNX'. A detailed explanatory text at the bottom right reads: 'You must specify your authtoken to ngrok so that your client is tied to this account. You can do this by specifying the authtoken parameter or by saving it in ~/.ngrok2/ngrok.yml so that you don't need to repeat this step.'

Deploy-4

- in your download path

lazy guys:

move ngrok file to your project directory (myproject)

make it globally:

- linux:

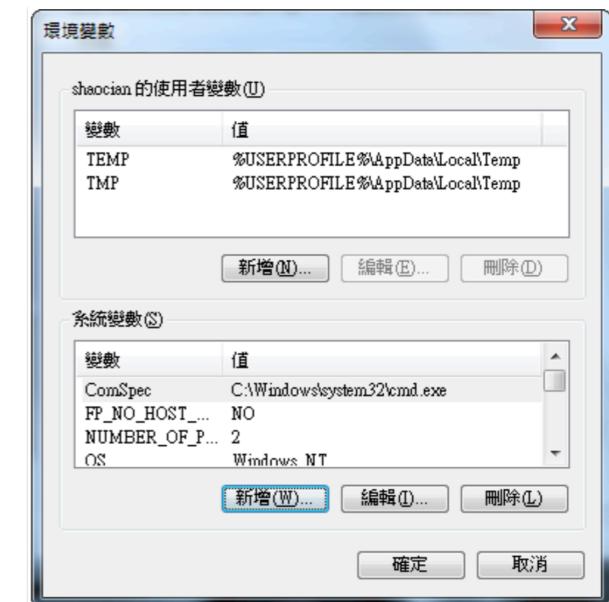
sudo mv ngrok /usr/local/bin

- windows:

1. move ngrok file to C:\Windows\system32

2. set Environment Variables (press “新增”)

add your path C:\Windows\system32\ngrok.exe



Deploy-5

- myproject/ (when you don't set ngrok as global var)

ngrok http 8000 (or whatever port your Django server adopts)

```
ngrok by @inconshreveable                                     (Ctrl+C to quit)

Session Status          online
Account                jojotenya (Plan: Free)
Version                2.3.30
Region                 United States (us)
Web Interface          http://127.0.0.1:4040
Forwarding             http://e2b25fdb.ngrok.io -> http
                        https://e2b25fdb.ngrok.io -> ht

Connections            ttl     opn     rt1     rt5
                        8       0       0.00   0.00

HTTP Requests
-----
GET /favicon.ico      404 Not Found
GET /static/css/blog.css 200 OK
GET /blog/             200 OK
GET /blog              301 Moved Permanently
GET /                  404 Not Found
GET /                  404 Not Found
GET /favicon.ico      400 Bad Request
GET /                  400 Bad Request
```

Deploy-6

- myproject/project/settings.py

```
DEBUG = False  
ALLOWED_HOSTS = ['e2b25fdb.ngrok.io']  
STATIC_ROOT = os.path.join(BASE_DIR, 'static')
```

- myproject/
python manage.py collectstatic
python manage.py runserver

Deploy-7



https://uwsgi.readthedocs.io/en/latest/tutorials/Django_and_nginx.html

Demo

- 一週新聞 (https://github.com/jojotenya/news_dashboard_demo.git)
- 即時人臉辨識 (https://github.com/jojotenya/realtime_face_detection.git)

Other Tools:

- Python: Dash (五顆星，文件清楚)
- Python: Bokeh (less recommend)
- Django + D3.js (更多的客製化)
- R: shinyapps (五顆星，速度快！)
- Tableau

附錄：Foreignkey on_delete參數

- **CASCADE**:Cascade deletes. Django emulates the behavior of the SQL constraint ON DELETE CASCADE and also deletes the object containing the ForeignKey.
- **PROTECT**:Prevent deletion of the referenced object by raising ProtectedError, a subclass of django.db.IntegrityError.
- **SET_NULL**:Set the ForeignKey null; this is only possible if null is True.
- **SET_DEFAULT**:Set the ForeignKey to its default value; a default for the ForeignKey must be set.
- **SET()**:Set the ForeignKey to the value passed to SET(), or if a callable is passed in, the result of calling it. In most cases, passing a callable will be necessary to avoid executing queries at the time your models.py is imported
- **DO_NOTHING**:Take no action. If your database backend enforces referential integrity this will cause an IntegrityError unless you manually add an SQL ON DELETE constraint to the database field.

Queryset-Methods Cheat Sheet

Methods	說明	example
get	取一特定的記錄	Post.objects.get(id=1)
create	新增記錄	Post.objects.create(author=me, title='test', text='test')
update	更新記錄	Post.objects.filter(id=1).update(title='test')
get_or_create	檢查是否存在，存在則取值，不存在則新增	created = Person.objects.get_or_create(first_name='John')
update_or_create	檢查是否存在，存在則更新，不存在則新增	created = Person.objects.update_or_create(first_name='John')
delete	刪除記錄	Post.objects.filter(id=1).delete()
first	取第一個值	p = Post.objects.order_by('pub_date').first()
last	取最後一個值	p = Post.objects.order_by('pub_date').last()
count	計算符合的個數	Post.objects.filter(title__contains='Django').count()

Queryset-Field lookups Cheat Sheet

Field lookups	說明	example
exact	與指定內容完全相等	Post.objects.get(id__exact =14)
iexact	與指定內容相等但不分大小寫	Post.objects.get(title__iexact ='My first blog')
contains	包含指定的值	Post.objects.get(title__contains ='Django')
icontains	同上但不分大小寫	Post.objects.get(title__icontains ='django')
in	在指定的list當中	Post.objects.filter(id__in =[1, 3, 4])
gt	大於指定的內容	Post.objects.filter(id__gt =4)
gte	大於等於指定的內容	Post.objects.filter(id__gte =4)
lt	小於指定的內容	Post.objects.filter(id__lt =4)
lte	小於等於指定的內容	Post.objects.filter(id__lte =4)
startswith	以指定的字串作為開頭	Post.objects.filter(title__startswith ='django')
endswith	以指定的字串做結尾	Post.objects.filter(title__endswith ='django')
isnull	SELECT ... WHERE ... IS NULL;	Post.objects.filter(publish_date__isnull =True)
regex	用正則表達式來查找內容	Post.objects.get(title__regex =r'^^(An ? The) +')

附錄：Queryset-CREATE & UPDATE

	方法一、	方法二、
Create	<pre>Post.objects.create(author=me , title='test', text='test')</pre>	<pre>p = Post() p.author = me p.title = 'test' p.text = 'test' p.save()</pre>
Update	<pre>Post.objects.filter(id=1).update (title='test')</pre>	<pre>p = Post.objects.filter(id=1) p.author = me p.title = 'test' p.text = 'test' p.save()</pre>

附錄：Model小補充

`__str__`方法決定呼叫物件時的代表名稱：

models.py

```
from django.db import models  
from django.utils import timezone
```

```
class Post(models.Model):
```

```
    ...
```

```
def __str__(self):  
    return self.title
```

python manage.py shell

```
>>>p = Post.objects.get(id=1)  
>>>print p  
Django首發!
```

127.0.0.1/admin/blog/post/

Select post to change

Action:		-----	Go	0 of 3 selected
<input type="checkbox"/>	Post			
<input type="checkbox"/>	第三篇			
<input type="checkbox"/>	Django second post!			
<input type="checkbox"/>	Django首發!			

附錄：已經有資料在資料庫怎麼辦

- myproject/

執行 **python manage.py inspectdb** 觀看目前綁定的資料庫的結構

```
class Termdate(models.Model):
    term = models.CharField(max_length=20, blank=True, null=True)
    newsdate = models.DateField(blank=True, null=True)
    freq = models.IntegerField(blank=True, null=True)
    corpus = models.IntegerField(blank=True, null=True)

    class Meta:
        managed = False
        db_table = 'termdate'

class Termfreq(models.Model):
    term = models.CharField(primary_key=True, max_length=20)
    freq = models.IntegerField(blank=True, null=True)

    class Meta:
        managed = False
        db_table = 'termfreq'
```

執行 **python manage.py inspectdb > models.py** 將結果存入models.py
執行 python manage.py migrate 完成同步

附錄：連接各種資料庫

- myproject/project/settings.py

```
DATABASES = {  
    'default': {  
        'ENGINE': 'django.db.backends.postgresql_psycopg2',  
        'NAME': 'dbname',  
        'USER': 'root',  
        'PASSWORD': '',  
        'HOST': '',  
        'PORT': ''  
    }  
}
```

postgredb

```
DATABASES = {  
    'default': {  
        'ENGINE': 'sqlserver_ado',  
        'NAME': 'dbname',  
        'USER': 'root',  
        'PASSWORD': '',  
        'HOST': 'dbserver\\ss2008',  
        'PORT': ''  
    }  
}
```

sqlserver

```
DATABASES = {  
    'default': {  
        'ENGINE': 'django.db.backends.mysql',  
        'NAME': 'dbname',  
        'USER': 'root',  
        'PASSWORD': '',  
        'HOST': 'localhost',  
        'PORT': '3306'  
    }  
}
```

mysql

註：sql server 詳見<http://django-mssql.readthedocs.org/en/latest/>

附錄：request 介紹 <HttpRequests類別的實作物件>

自帶參數(常用)：

- request.path: 請求此view的路徑。
- request.method: 此次請求的HTTP方法。
 - ex. request.method == 'POST' 會回傳boolean。
- request.POST/request.GET: 回傳 POST/GET 方法傳遞的dictionary-like物件。
- request.COOKIES: 此次請求所帶的所有cookies。
- request.META: 此次請求所有HTTP headers，為一dictionary，可用字典方法 request.META.items() 取得所有key-value。

middle-ware設置的參數：

- request.session: 一個dictionary-like物件，用來存取session。
 - ex. request.session['id'] = 1，可存入一個key='id', value=1的session物件。
- request.user: 如果網站有設置使用者，並且該使用者登入的情況之下，可由此取得 AUTH_USER_MODEL物件，無登入情況之下會獲得AnonymousUser物件。

自帶方法(常用)：

- request.get_full_path(): GET方法有帶查詢字符串時的完整路徑。
 - ex. /blog/?page=1
- request.get_host(): 得到主機(網域)名稱。

jojotenya@gmail.com