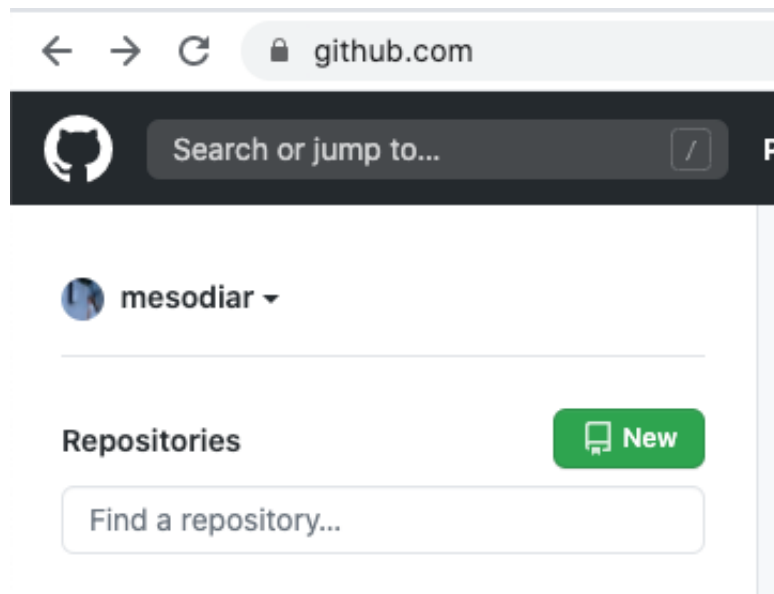


สารบัญ

1. สร้างที่เก็บโค้ดใน Github Repository ของโปรเจกต์กัน 🐙
 2. ติดตั้ง Django 😊
 3. เริ่มต้นสร้างโปรเจกต์ 🎉
-

1. สร้าง Github Repository 🐙

เข้าไปที่ www.github.com และกดปุ่มเพื่อสร้าง repository



ขั้นตอนนี้จะกดสร้าง README.md และ gitignore เป็น Python ด้วย

github.com/new


Search or jump to...

Pull requests Issues Marketplace Explore

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner ^{*} Repository name ^{*}

 mesodiar / medium-test ✓

Great repository names are short and memorable. Need inspiration? How about [laughing-disco](#)?

Description (optional)

☒ **Public**
Anyone on the internet can see this repository. You choose who can commit.

☐ **Private**
You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

☐ **Initialize this repository with a README**
This will let you immediately clone the repository to your computer.

Add .gitignore: **None** Add a license: **None** ⓘ

Create repository

github.com/mesodiar/medium-test

Search or jump to...

Pull requests Issues Marketplace Explore

mesodiar / medium-test

Issues Pull requests Actions Projects Wiki Security Insights Settings

master 1 branch 0 tags

mesodiar Initial commit

.gitignore Initial commit

Help people interested in this repository understand your project by adding a README

Go to file Add file **Code**

Clone with SSH ⓘ [Use HTTPS](#)
Use a password protected SSH key.
`git@github.com:mesodiar/medium-test`

Open with GitHub Desktop

Download ZIP

About
No description, v provided.

Releases
No releases published
[Create a new release](#)

Packages

เรา clone project

```
git clone git@github.com:mesodiar/medium-test.git
```

เราจะพบว่า มี folder ชื่อ `medium-test` เกิดขึ้น และมีไฟล์ `README.md` ซ้ำงใน

```
→ medium-test git:(master) ls  
README.md
```

2. ติดตั้ง Django 🤖

เริ่มต้นร้านค้า

```
## Mac
pipenv install django

## Windows
python -m pipenv install django
```

```
✓ Successfully created virtual environment!
Virtualenv location: /Users/mesodiar/.local/share/virtualenvs/medium-test-X6TvBSrh
Creating a Pipfile for this project...
Installing django...
Adding django to Pipfile's [packages]...
✓ Installation Succeeded
Pipfile.lock not found, creating...
Locking [dev-packages] dependencies...
Locking [packages] dependencies...
Building requirements...
Resolving dependencies...
✓ Success!
Updated Pipfile.lock (a6086c)!
Installing dependencies from Pipfile.lock (a6086c)...
🐍 ████████████████████████████████████████████ 0/0 -
To activate this project's virtualenv, run pipenv shell.
Alternatively, run a command inside the virtualenv with pipenv run.
```

จะได้ไฟล์ Pipfile กับ Pipfile.lock

```
## Mac
pipenv shell

## Windows
python -m pipenv shell
```

คราวนี้เราจะไป เริ่ม project กัน

```
django-admin startproject <project name>
```

จะได้ structure ไฟล์ ตามนี้ (ใน Windows ลอง tree ได้)

```
.
├── Pipfile
├── Pipfile.lock
└── medium
    ├── manage.py
    └── medium
        ├── __init__.py
        ├── asgi.py
        ├── settings.py
        ├── urls.py
        └── wsgi.py
```

```
python manage.py migrate
python manage.py runserver
```

```
python manage.py createsuperuser
```

3. เริ่มต้นสร้างโปรเจค 🎉

```
python manage.py startapp stories
```

ไฟล์ models.py

```
from django.conf import settings
from django.db import models

class Stories:
    author = models.ForeignKey(settings.AUTH_USER_MODEL,
on_delete=models.CASCADE)
    title = models.CharField(max_length=100)
    content = models.TextField()
    created_date = models.DateTimeField(auto_now=True)
    published_date = models.DateTimeField(blank=True, null=True)

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

**** ถ้าไม่ใส่ (models.Model) จะ เกิด error**
for model in model_or_iterable:
TypeError: 'type' object is not iterable
แล้วจะ migrate ไม่ได้

ไปแก้เพิ่ม 'story', ที่ installed_apps

ไม่จั่งจะรัน

```
python manage.py makemigrations
```

อันนี้ medium/urls.py

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

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

ไฟล์ stories/urls.py

```
from django.urls import path

from . import views

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



← → ↻ ⓘ localhost:8000/stories/

This is your story

```
from django.http import HttpResponse

def index(request):
    return HttpResponse("This is your story index")
```

คราวนี้เราจะแสดงหน้า story detail กัน แต่เราต้องมีข้อมูลของ story ก่อน



Django administration

WELCOME, MILS. VIEW SITE / C

Home › Stories › Storys › Add story

Add story

Author:

mils  


Title:


My first story

Content:

This is my first story content

Published date:

Date: 2020-07-17  Today

Time: 03:56:39  Now

Note: You are 7 hours ahead of server time.

Save and add another

Save and continue

```
from django.urls import path

from . import views

urlpatterns = [
    path('', views.index, name='index'),
    path('<int:story_id>/', views.detail, name='detail'), ## add this line
]
```

```
from django.http import HttpResponse

def index(request):
    return HttpResponse("This is your story index")

## add this function
def detail(request, story_id):
    return HttpResponse(f'This is story detail of {story_id}')
```

← → ↺ ⓘ localhost:8000/stories/1/

This is story detail of 1

ไม่ว่าเราจะ <http://localhost:8000/stories/2> หรือ 3 หรือ 4

เราก็จะเหมือนส่งผ่าน url -> view

มันยังเป็นการแสดงผลแบบhardcode อยู่ ไม่ dynamic

คราวนี้เราจะแสดงผลจากข้อมูลจริงๆที่มีอยู่

ตอนนี้เราจะสอนเรื่อง การ query ใน shell ได้

```
>>> from stories.models import Story
>>> Story.objects.all()
<QuerySet [<Story: My first story>]>
>>> █
```

เราจะลอง Story.objects.get(id=1) และลอง Story.objects.get(id=2) ดู

จะเห็นว่าตอนนี้เรามีแค่ Story ที่ id 1 เท่านั้น

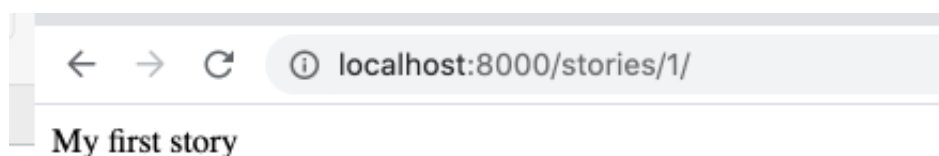
```
>>> Story.objects.get(id=1)
<Story: My first story>
>>> Story.objects.get(id=2)
Traceback (most recent call last):
  File "<console>", line 1, in <module>
  File "/Users/mesodiar/.local/share/virtualenvs/medium-but-high-Ez05_pPb/lib/python3.7/site-packages/django/db/models/manager.py", line 82, in manager_method
    return getattr(self.get_queryset(), name)(*args, **kwargs)
  File "/Users/mesodiar/.local/share/virtualenvs/medium-but-high-Ez05_pPb/lib/python3.7/site-packages/django/db/models/query.py", line 417, in get
    self.model._meta.object_name
stories.models.Story.DoesNotExist: Story matching query does not exist.
>>> █
```

ถ้าเราลองกลับไปแก้

```
def detail(request, story_id):
    story = Story.objects.get(id=story_id)

    return HttpResponse(story)
```

เราจะเหมือนกับว่าเอา object ส่งไป render หน้าบ้าน



นั่นเพราะเรามีstr ใน models.py มันจึงแสดงผลแค่ title แต่เราจะแสดงผล content ด้วย

เราจะเปลี่ยน HttpResponse เป็น render()

Render() —> It's a very common idiom to load a template, fill a context and return an [HttpResponse](#) object with the result of the rendered template. Django provides a shortcut. Here's the full `index()` view, rewritten:

ในขณะเดียวกัน ไปสร้าง template ด้วย

stories/templates/stories/detail.html

```
<h1>This is from template</h1>
```

```
views.py > ...
1 | from django.shortcuts import render
2 | from django.http import HttpResponse
3 |
4 | from .models import Story
5 |
6 |
7 | def index(request):
8 |     return HttpResponse("This is your story index")
9 |
10 | def detail(request, story_id):
11 |     story = Story.objects.get(id=story_id)
12 |
13 |     return render(request, 'stories/detail.html', story)
```

จะพบว่า

← → ↻ ⓘ localhost:8000/stories/1/

TypeError at /stories/1/
context must be a dict rather than Story.

Request Method: GET

Request URL: http://localhost:8000/stories/1/

Django Version: 2.0.8

เราต้อง ส่ง Story เป็น context ในรูปแบบของ dict

```
def detail(request, story_id):
    story = Story.objects.get(id=story_id)

    context = {'story': story}

    return render(request, 'stories/detail.html', context)
```


This is from template

```
<h1>{{ story.title }} by {{ story.author }}</h1>
<p>{{ story.content }}</p>
```

My first story by mils

This is my first story content

เสร็จแล้ว

ลอง complete ที่ index.html

```
def index(request):
    stories = Story.objects.all()
    context = {'stories': stories}
    return render(request, 'stories/index.html', context)
```

```
<h1>This is your story index</h1>
<ul>
{% for story in stories %}
    <li><a href="/stories/{{ story.id }}">{{ story }}</a></li>
{% endfor %}
</ul>

<h1>This is your story index</h1>
<ul>
{% for story in stories %}
    <li><a href="{% url 'detail' story.id %}">{{ story }}</a></li>
{% endfor %}
</ul>
```

คราวนี้เราก็เข้าใจ template tag ละ

ตอนนี้ ต้องสรุป

ภาพรวมของ MVT

ใส่ Bootstrap

```
<link
    rel="stylesheet"

href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css"
    integrity="sha384-
9aIt2nRpC12Uk9gS9baDl411NQApFmC26EwAOH8WgZl5MYYYxFfc+NcPb1dKGj7Sk"
    crossorigin="anonymous"
/>
<script
    src="https://code.jquery.com/jquery-3.5.1.slim.min.js"
    integrity="sha384-
DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj"
    crossorigin="anonymous"
></script>
<script

src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js"
    integrity="sha384-
Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo"
    crossorigin="anonymous"
></script>
<script

src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.min.js"
    integrity="sha384-
OgVRvuATP1z7JjHLkuOU7Xw704+h835Lr+6QL9UvYjZE3Ipu6Tp75j7Bh/kR0JKI"
    crossorigin="anonymous"
></script>
```

This is from template

My first story by mils

This is my first story content

คราวนี้เราก็สร้าง ใช้ css ใน project ของเราเองบ้าง

โดยการสร้างไฟล์ css ที่ stories/static/css/story.css

แล้วเราก็ใส่

```
{% load static %}  
  
<link rel="stylesheet" href="{% static 'css/story.css' %}" />
```

คราวนี้เราจะยังใช้ไม่ได้เพราะ django เรายังไม่รู้ว่าต้องไปหาไฟล์ css ที่ไหน

ถ้าเราลอง inspect ดู

<http://localhost:8000/static/css/story.css>

```
STATICFILES_DIRS = [  
    os.path.join(BASE_DIR, 'static')  
]
```

คราวนี้ถึงไม่ 404 แล้ว



localhost:8000/static/css/story.css

```
h1 {  
  color: #15c4b5;  
}
```



localhost:8000/stories/1/

This is from template

My first story by mils

This is my first story content

เข้า [FontCDN: A search tool for Google web fonts](https://fonts.googleapis.com/css?family=PT+Serif:700) แล้วเลือกฟอนชื่อ PT serif

```
<link href='https://fonts.googleapis.com/css?family=PT+Serif:700'  
rel='stylesheet' type='text/css'>  
<link href='https://fonts.googleapis.com/css?family=Abril+Fatface'  
rel='stylesheet' type='text/css'>
```

```
h1 {  
  color: #15c4b5;  
  font-family: "PT Serif", serif;  
}
```

คราวนี้เราก็ใช้ฟอนได้แล้ว

อยากเพิ่ม image

```
image = models.ImageField(null=True, blank=True)
```

```
python manage.py makemigrations
```

มันต้องลง Package Pillow ด้วย

```
pipenv install Pillow
```

```
python manage.py makemigrations  
python manage.py migrate
```

เข้า unsplash ไป download ภาพ แล้ว upload เข้าหลังบ้าน

```
MEDIA_URL = '/media/'  
MEDIA_ROOT = os.path.join(BASE_DIR, 'media')
```

```
from django.conf import settings  
from django.conf.urls.static import static  
from django.contrib import admin  
from django.urls import path, include  
  
urlpatterns = [  
    path('admin/', admin.site.urls),  
    path('stories/', include('stories.urls'))  
] + static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

สร้าง folder /media/ ด้วย

แล้วย้ายรูปที่ upload ไปไว้ใน /media/ รูป.jpg

แก้ text ใน content

```
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas porttitor  
congue massa. Fusce posuere, magna sed pulvinar ultricies, purus lectus  
malesuada libero, sit amet commodo magna eros quis urna.  
Nunc viverra imperdiet enim. Fusce est. Vivamus a tellus.  
Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac  
turpis egestas. Proin pharetra nonummy pede. Mauris et orci.
```

Extend
base.html

```
{% block content %}
{% endblock %}
```

detail.html

```
{% extends 'stories/base.html' %}
```

คราวนี้เราเป็น admin เราเลยมีสิทธิ์เพิ่ม story ได้ แต่จุดมุ่งหมายคือเราจะให้คนทั่วไปเขาสามารถสร้าง story ได้เอง เพราะหน้า admin จะให้แค่ คนที่เป็น admin เข้าได้เท่านั้น

```
def new_story(request):
    form = StoryForm()
    return render(request, 'stories/new_story.html', {'form': form})
```

เราจึงต้องทำ form.py

```
from django import forms

from .models import Story

class StoryForm(forms.ModelForm):
    class Meta:
        model = Story
        fields = ('author', 'title', 'content', 'image')
```

```
{% extends 'stories/base.html' %}

{% block content %}
    <h2>New Story</h2>
    <form method="POST" class="post-form">{% csrf_token %}
        {{ form.as_p }}
        <button type="submit" class="save btn btn-dark">Save</button>
    </form>
{% endblock %}
```

```
def new_story(request):
    if request.method == "POST":
        form = StoryForm(request.POST, request.FILES)
        if form.is_valid():
            form.save()

            return redirect('index')
    else:
        form = StoryForm()
    return render(request, 'stories/new_story.html', {'form': form})
```

เพิ่ม Navbar

คราวนี้เพิ่ม fontawesome แล้วเราก็สมัครโดยการใส่เมล