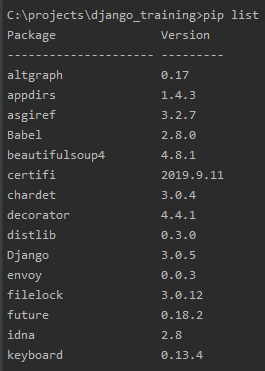
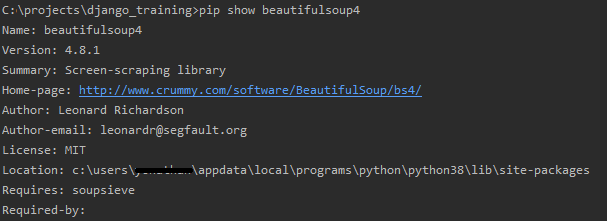
# How I learn Django

## Refrech python :

1. List package : pip list



1. Get package detail : pip show beautifulsoup4



1. Install virtual environement : pip install virtualenv

* Create a virtual environment for a project :

cd project\_folder

virtualenv venv

* Begin to use env in windows : env\Scripts\activate
* For stop : deactivate

## Create an install file : requirements.txt

'''

pip freeze > requirements.txt

'''

## Install all package :

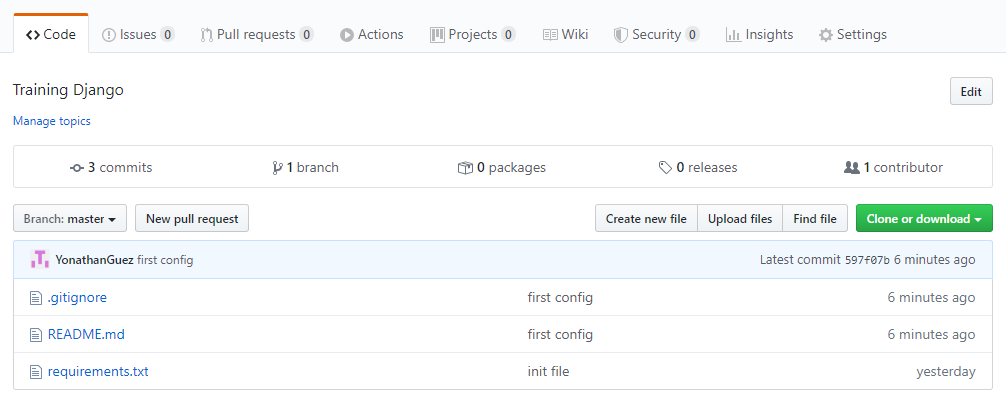
'''

pip install -r requirements.txt

'''

## Configuration CI with Github :

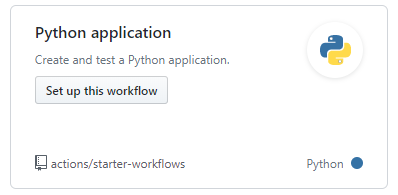
1. After your first push with requirements.txt



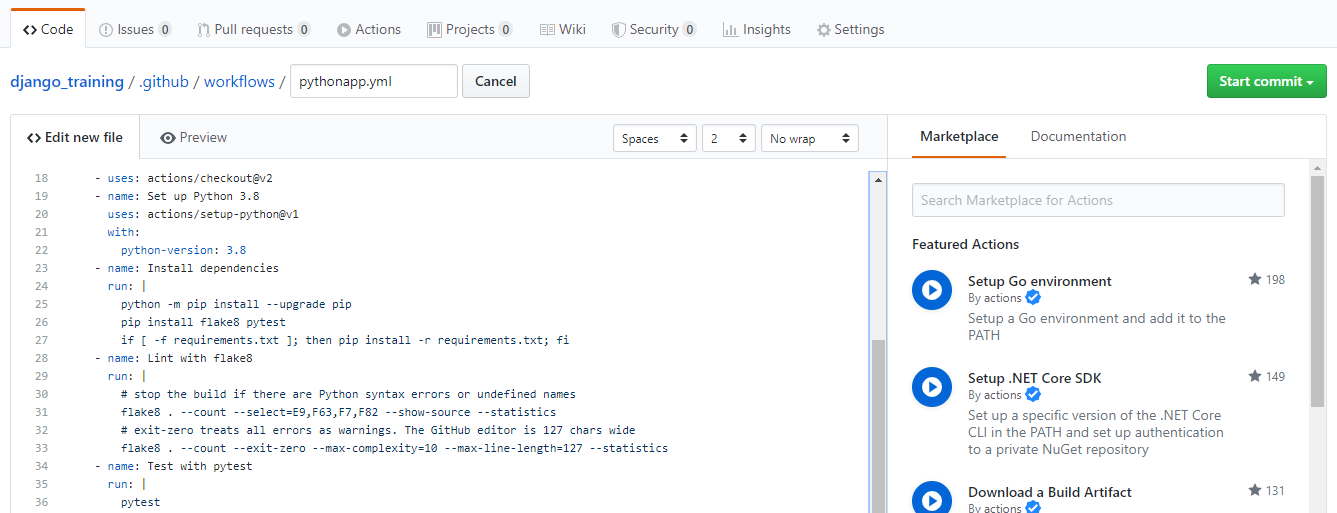
1. Go to Action :



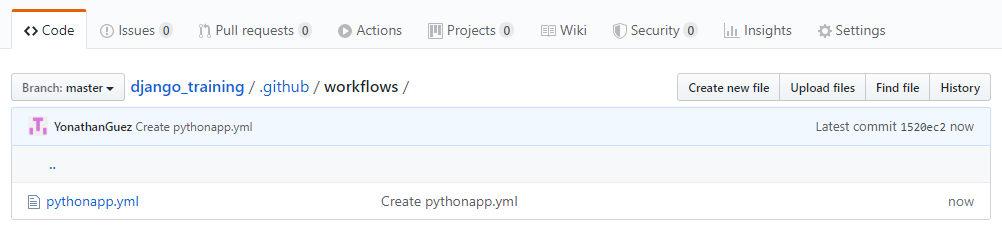
Scroll down :



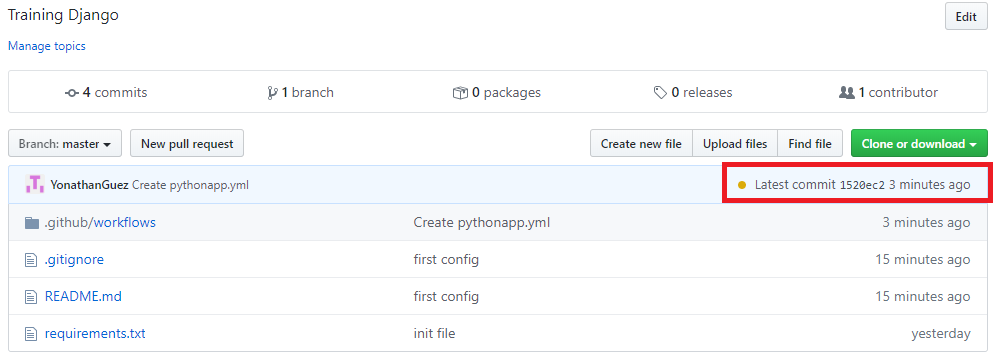
1. Set up this Workflow :



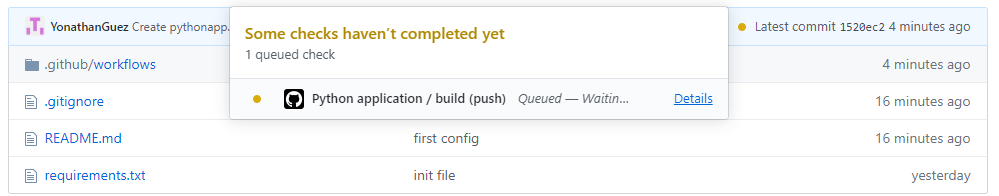
1. Start commit :



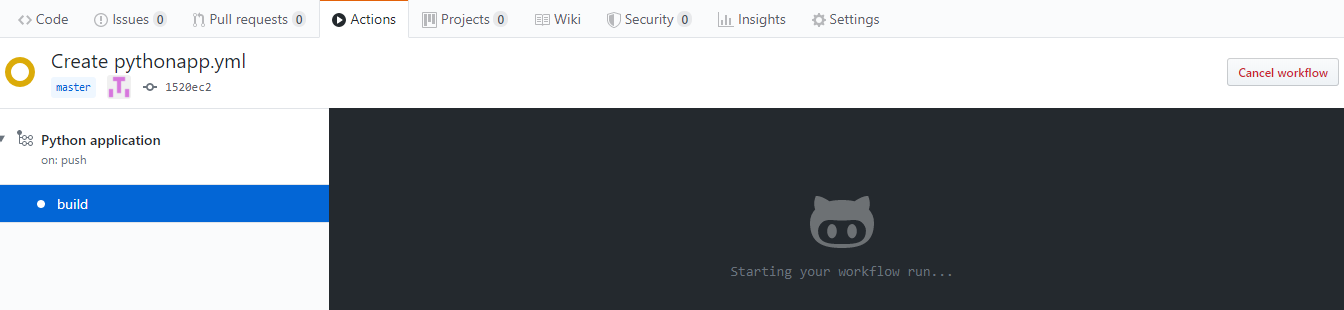
1. Go Back to the root :



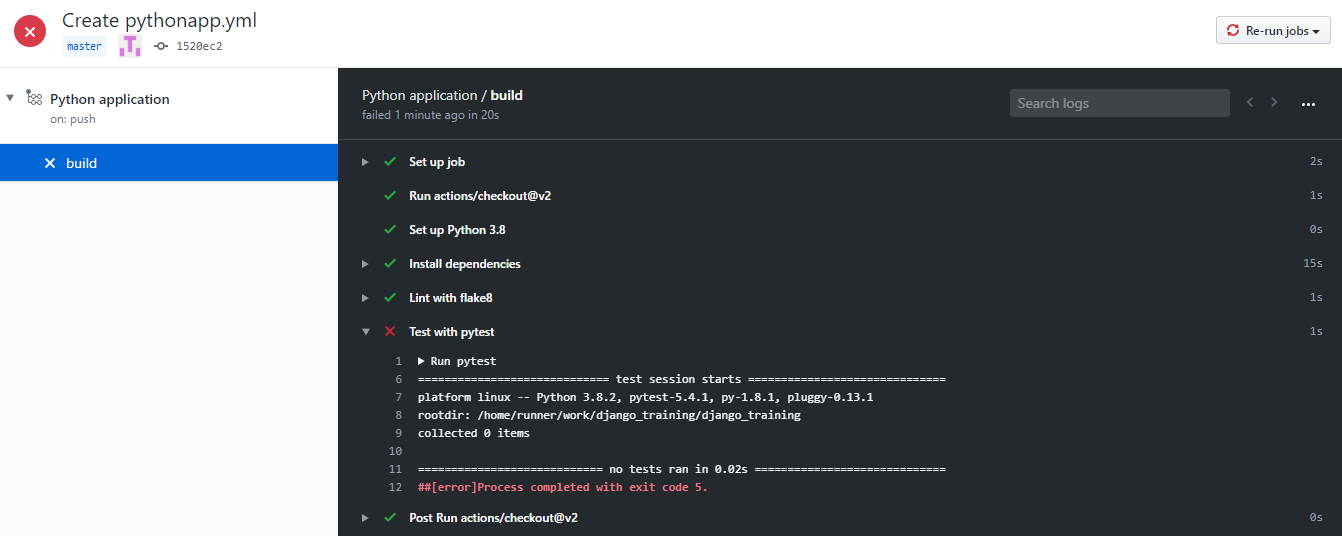
1. Click on Yellow button :



1. Click Detail :



1. Wait for the result :

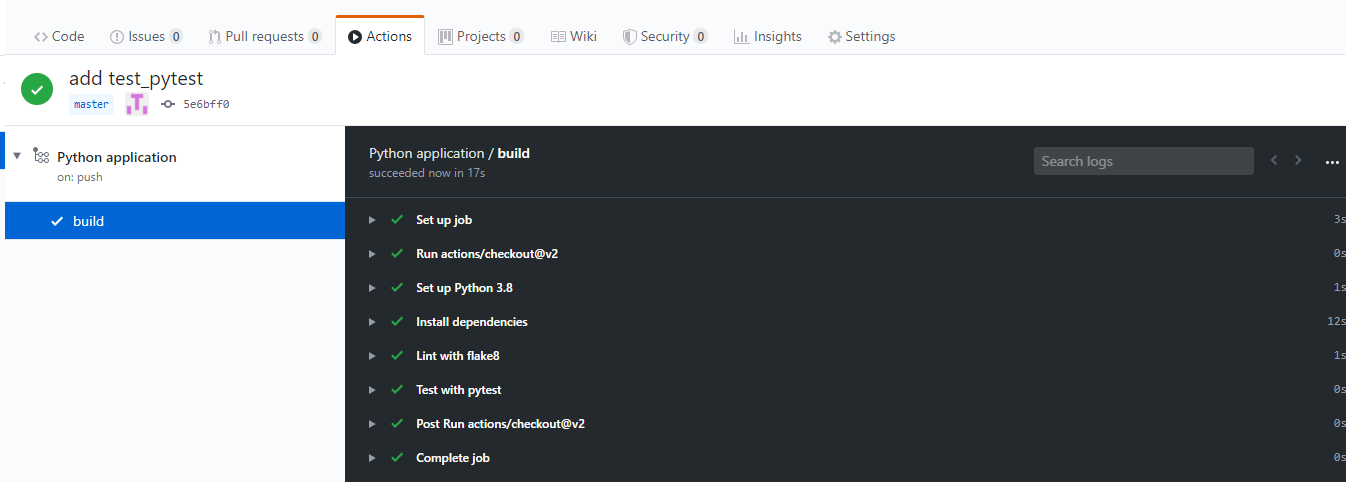


1. We have one error in the CI :

We will create test\_pytest.py to the root :

def helloTest():  
 return "hello test"  
  
def test\_pytest():  
 assert helloTest() == "hello test"

1. After the Push my CI run the Yml :



All it s green so it run my test\_pytest.py

## 1) generate first admin

'''

django-admin startproject mysite

'''

That will create :

'''

mysite/

manage.py

mysite/

\_\_init\_\_.py

settings.py

urls.py

asgi.py

wsgi.py

'''

## 2) Check if it’s work

Enter in mysite: run the server

'''

py manage.py runserver

'''

## 3) Give the IP available

All available public IPs:

Listen on all available public IPs use: (0 is a shortcut for 0.0.0.0. )

'''

py manage.py runserver 0:8000

'''

## 4) Creation application and Work with rooting in Django:

* we need to be at the same level of manage.py before to run this:

```

python manage.py startapp polls

```

that will create :

```

polls/

\_\_init\_\_.py

admin.py

apps.py

migrations/

\_\_init\_\_.py

models.py

tests.py

views.py

```

* We will add a file urls.py :

```

from django.urls import path

from . import views

urlpatterns = [

path('', views.index, name='index'),

]

```

* and put something in views.py :

```

from django.http import HttpResponse

def index(request):

return HttpResponse("Hello, world. You're at the polls index.")

```

that will read our function index when we enter in /polls/

* For rooting our new app

And in test\_site/urls.py we will import include and this line:

```

path('polls/', include('polls.urls')),

```

relaunch our server

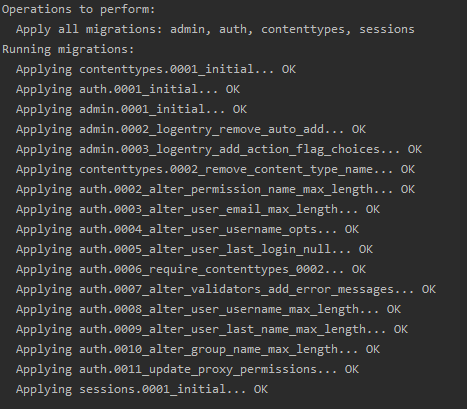
Now we can check the URL: <http://localhost:8000/polls/>

## 5) Configuration DataBase :

By default we have SQLIGHT we can add also an other database we need to change settings.py for that

1. By default we haven t tables install in SQLIGHT for that we need to run :

python manage.py migrate



So that will set into  [INSTALLED\_APPS](https://docs.djangoproject.com/fr/3.0/ref/settings/#std:setting-INSTALLED_APPS):

'django.contrib.admin',  
'django.contrib.auth',  
'django.contrib.contenttypes',  
'django.contrib.sessions',  
'django.contrib.messages',  
'django.contrib.staticfiles',

1. We want to create database now :

* We will build a model into polls/models.py :

**from** **django.db** **import** models

**class** **Question**(models.Model):

question\_text = models.CharField(max\_length=200)

pub\_date = models.DateTimeField('date published')

**class** **Choice**(models.Model):

question = models.ForeignKey(Question, on\_delete=models.CASCADE)

choice\_text = models.CharField(max\_length=200)

votes = models.IntegerField(default=0

* For activate model in settings.py  we need to add a line into INSTALLED\_APPS:

'polls.apps.PollsConfig',

* And run : python manage.py makemigrations polls

**Makemigrations indicate model change**

* Now django know we change the model as we can see un polls/migrations/0001\_initial.py for change new or old migration  :

python manage.py sqlmigrate polls 0001

* Add the migration set : python manage.py migrate