**Django**

Are you looking for a web framework that is open-source and uses model-view-template(MVT) architecture to create a database driven website?

Introducing Django , built using python with a principle known as Don’t Repeat yourself to suit your need. It provides an abstraction with reusable and pluggable components.

Django consists of

- Object-Relational Mapper(ORM) : mediates between data models.

- Relational Database Model : a system for processing web requests.

- View : A template system dealing with functional logic

- Controller : a regular-expression based URL dispatcher.

**Let’s Focus on the ORM and model in detail.**

**Why Django :**

- Ridiculously Fast

- Fully loaded

- Reassuringly Secure

- Exceedingly scalable

- Incredibly Versatile

Django is powered with inclusive battery – the Django ORM

**Django Installation**

Since Django is python based module, install using the pip installer

$ pip install django

To verify the installation, get the installed version using

$ python3 -c “import django; print(django.get\_version())”

1.11.7

**Database support**

Django comes with a default SQLite. However, packages are available for other database soures.

- PostgreSQL

- MySQL

- Oracle

**Your First Django Project:**

**Project Creation:** The Django environment is ready, and you can build your first web application now. It starts by creating a new Django project.

From command line, cd to a directory where the project code will be created.

Use the django-admin command to create a project myweb.

$ django-admin startproject myweb

**What Entails a Django project:**

Django creates a project structure like:

Container for the project

myweb/

A command-line utility to interact with the project myweb

manage.py

Python package for the project. Import it to use any code inside

myweb/

myweb directory is considered a Python package due to this empty file

\_\_init\_\_.py

Setting/Configuration for myweb project

settings.py

URL declarations similar to Table of contents for myweb project

urls.py

An entry-point for WSGI-compatible web servers to serve myweb project.

wsgi.py

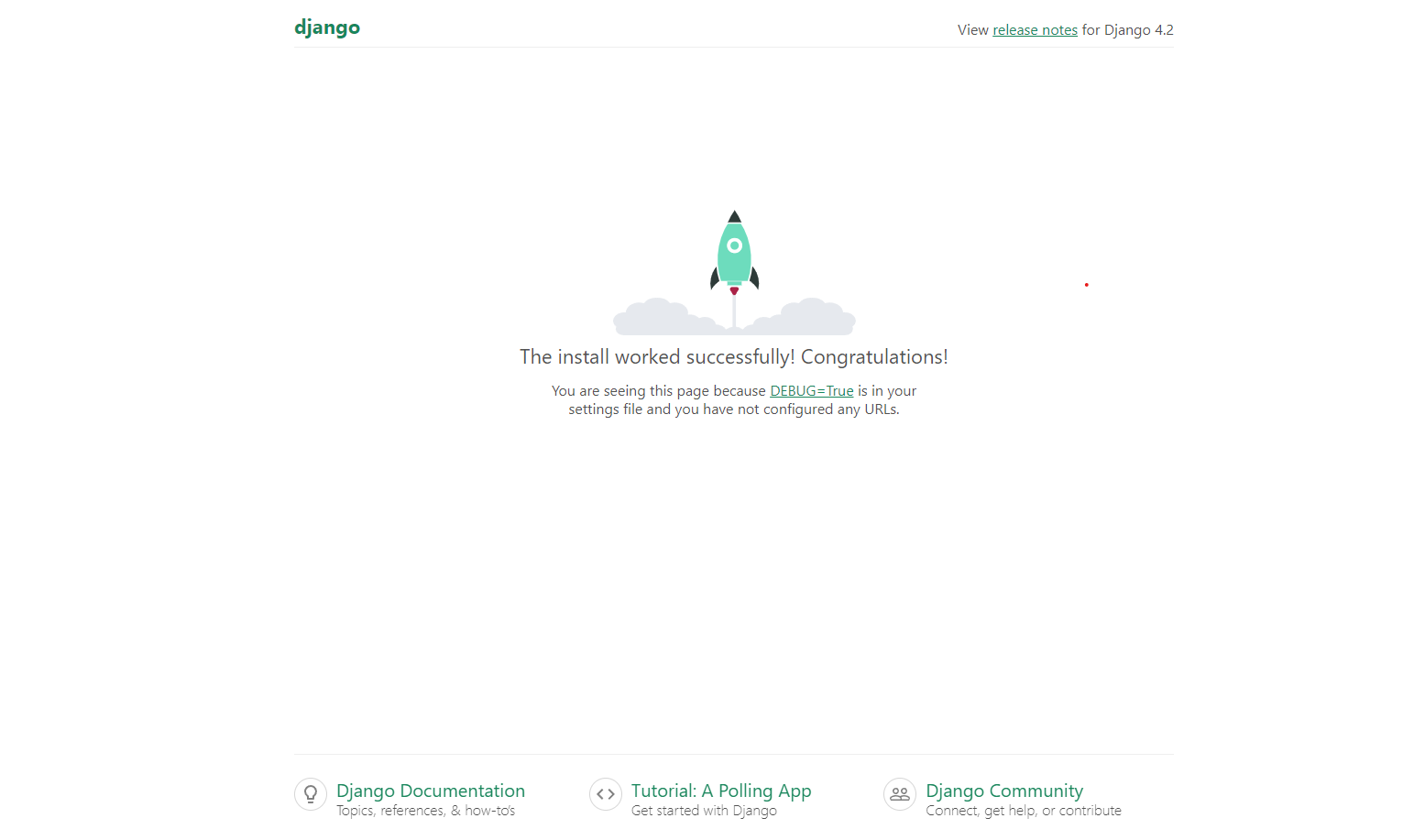
**Up and Running your Project:**

Lets start the project to see the Django web interface. On the command line, change to myweb root directory and execute the command

$ python manage.py runserver

You’ll see a output on the command line.

Open the web browser and provide the URL <http://127.0.0.1:8000/> to launch the Django project page.



**Create an App:**

To work with the project, you need a application or app that performs different actions.

To create an app poll. Go to the container directory myweb and execute the command.

$ python manage.py startapp poll

The poll app is created with all necessary components.

**What an App Entails:**

A Django app has the following:

admin.py reads model metadata and provides an interface to manage app content

views.py web based requests and response is configured in this file

apps.py application configuration details for the app are included. Eg: custom app name.

tests.py app unit test automation classes are included in this

models.py A class for each model is defined with the model structure layout

migrations/ contains migrated model details with the corresponding database table structure.