directory set up:

mkdir flask-by-example && cd flask-by-example

Initialize a new git repo within your working directory:

git init

Set up a virtual and activate venv:

python -m venv ./env

cd env\Scripts  
activate  
cd ../..

Add the following files to your “flask-by-example” folder:

echo >> app.py

This will give you the following structure:

├── .gitignore

├── app.py

├── README.md

└── requirements.txt

Next:

pip install Flask

Check it is there:

pip show flask  
  
# Output will be:  
Name: Flask

Version: 1.1.2

Summary: A simple framework for building complex web applications.

Home-page: https://palletsprojects.com/p/flask/

Author: Armin Ronacher

Author-email: armin.ronacher@active-4.com

License: BSD-3-Clause

Location: d:\softunistudents\public\_lectures\flask\_apps\flask-by-example\env\lib\site-packages

Requires: click, Werkzeug, Jinja2, itsdangerous

Required-by:

Open up app.py in PyCharm add the following code:

from flask import Flask

app = Flask(\_\_name\_\_)

@app.route('/')

def hello():

return "Hello World!"

if \_\_name\_\_ == '\_\_main\_\_':

app.run()

Let’s create a db and table:  
  
>>> import sqlite3

>>> conn = sqlite3.connect('posts.db')

>>> conn.execute('CREATE TABLE post (title TEXT, text TEXT)')

<sqlite3.Cursor object at 0x011A1DE0>

>>> conn.close()

>>> exit()

We will use super simple db and we will write row slq queries for this example. Tha aim will be to have two endpoints – for fetching all posts from db and for creating a post via request arguments.  
  
The final code will look like this:  
  
**from** flask **import** Flask, request  
**import** sqlite3 **as** sql  
  
app = Flask(\_\_name\_\_)  
  
  
@app.route(**'/posts/'**)  
**def** posts():  
 con = sql.connect(**"posts.db"**)  
 con.row\_factory = sql.Row  
  
 cur = con.cursor()  
 cur.execute(**"select \* from post"**)  
 res = {}  
 rows = cur.fetchall()  
  
 index = 0  
 **for** entry **in** rows:  
 res[index] = dict(entry)  
 index += 1  
 **return** {**"rows"**: res}  
  
  
@app.route(**'/posts/create/'**)  
**def** create():  
 title = request.args.get(**'title'**)  
 text = request.args.get(**'text'**)  
 **with** sql.connect(**"posts.db"**) **as** con:  
 cur = con.cursor()  
 cur.execute(**"INSERT INTO post (title,text) VALUES(?, ?)"**, (title, text))  
  
 con.commit()  
 **return** {**"title"**: title, **"text"**: text}  
  
  
**if** \_\_name\_\_ == **'\_\_main\_\_'**:  
 app.run()

Call this url with different values for title and text:  
<http://127.0.0.1:5000/posts/create/?title=sometitle&text=here%20some%20text>

Then call this url and chech that they are fetched:  
  
<http://127.0.0.1:5000/posts/>

You will have something similar to:  
  
{"rows":{"0":{"text":null,"title":null},"1":{"text":"heloooo","title":"Mytitle"},"2":{"text":"heloooo","title":"Mytitle"},"3":{"text":"testtext","title":"test"},"4":{"text":"here some text","title":"sometitle"}}}

And we are done 😊

Github