**Práctica Python Flask Básico**

1. **Instalación**

Instalar Flask usando pip3:

sudo yum install epel-release

sudo yum install python3-pip

pip3 --version

sudo pip3 install Flask

pip3 freeze | grep Flask

1. **Probar con aplicación básica**

[vagrant@servidor ~]$ mkdir testFlask

[vagrant@servidor ~]$ cd testFlask /

[vagrant@servidor testFlask]$ vim hello\_world.py

Cree el archivo hello\_world.py con el siguiente contenido:

# hello\_world.py

from flask import Flask

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

@app.route('/')

def hello\_world():

return 'Hello World!'

Corra la aplicación:

[vagrant@servidor testFlask]$ export FLASK\_APP=hello\_world.py

[vagrant@servidor testFlask]$ python3 -m flask run --host=0.0.0.0

\* Serving Flask app 'hello\_world.py'

\* Debug mode: off

WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.

\* Running on all addresses (0.0.0.0)

\* Running on http://127.0.0.1:5000

\* Running on http://10.0.2.15:5000

Press CTRL+C to quit

En otra terminal, use la ip de la máquina donde instaló el servicio para probar:

[vagrant@servidor ~]$ curl 192.168.50.3:5000

Hello World!

1. **Crear Proyecto Flask más completo**

Clone el siguiente repositorio que contiene un proyecto de Python Flask más completo:

<https://github.com/omondragon/pythonFlaskExample1>

Analice la estructura del proyecto y explique el contenido de cada uno de sus archivos:

[vagrant@servidorm ~]$ cd pythonFlaskExample1/

[vagrant@servidor pythonFlaskExample1]$ cd my-project/

[vagrant@servidor my-project]$ tree

.

├── app

│   ├── data.py

│   ├── \_\_init\_\_.py

│   ├── templates

│   │   ├── about.html

│   │   ├── base.html

│   │   └── index.html

│   └── views.py

├── application.wsgi

├── config.py

└── run.py

Detenga el firewall y corra la aplicación

[vagrant@servidor my-project]$ sudo systemctl stop firewalld

[vagrant@servidor my-project]$ export FLASK\_APP=run.py

[vagrant@servidor my-project]$ python3 -m flask run --host=0.0.0.0

\* Serving Flask app 'run.py'

\* Debug mode: off

WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.

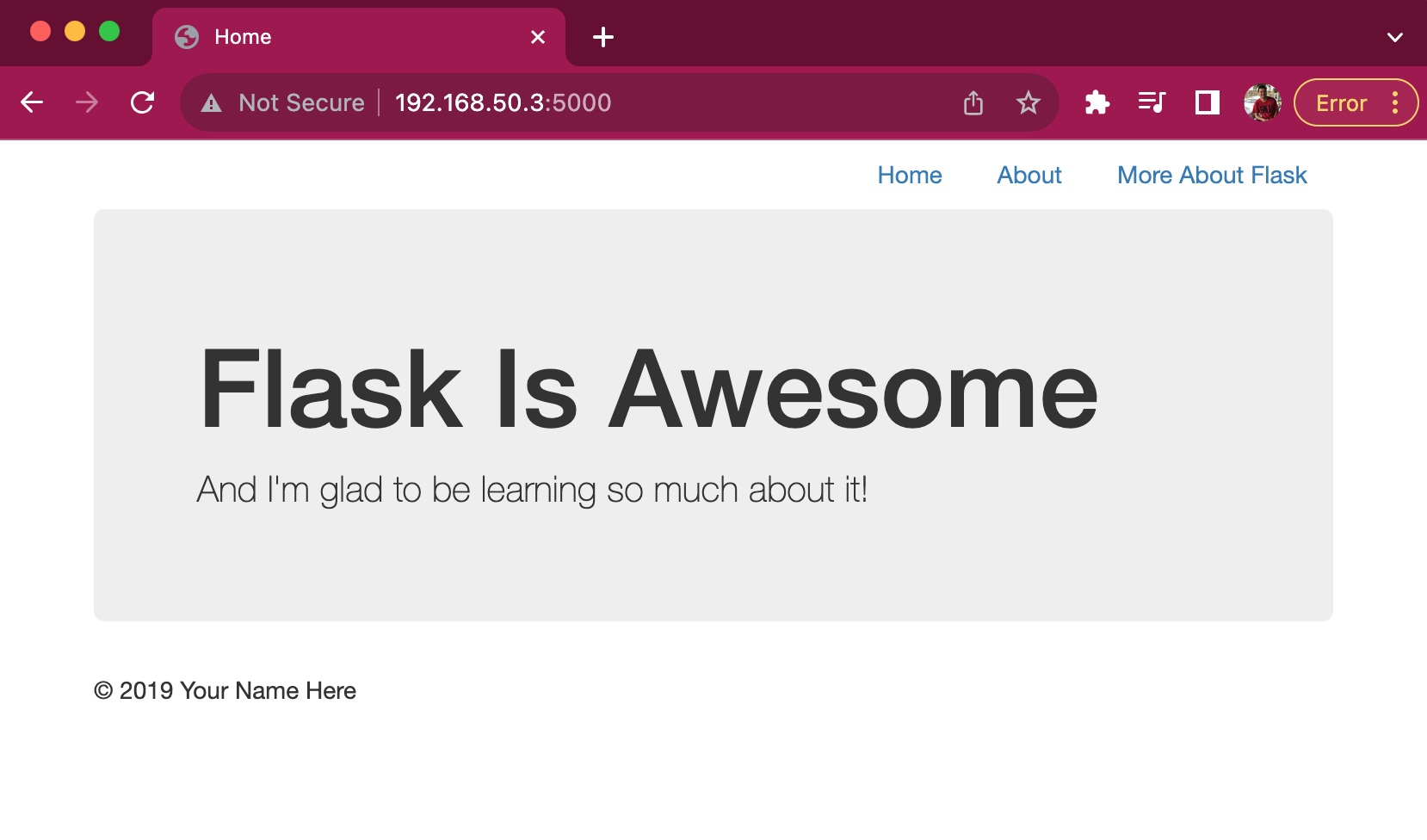
\* Running on all addresses (0.0.0.0)

\* Running on http://127.0.0.1:5000

\* Running on http://10.0.2.15:5000

Press CTRL+C to quit

Pruebe en un navegador



1. **Ejercicio**
2. Despliegue el sitio en Apache

Para esto debe instalar el modulo de wsgi

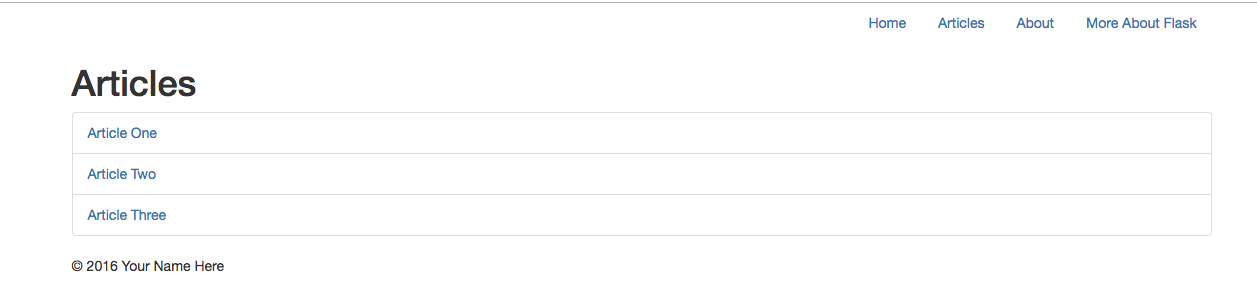
sudo yum install mod\_wsgi

httpd -M | grep wsgi

Se sugiere seguir el tutorial en:

<https://www.youtube.com/watch?v=cxIlQYyrpoU>

1. Agregue un view que muestre una lista de archivos



La lista de artículos debe cargarse desde el archivo **data.py**

1. Cree una vista que muestre la identificación del artículo una vez seleccionado de las listas de artículos.

Por ejemplo, si el usuario hace clic en el hipervínculo del artículo 2, debe mostrar:



**Referencias**

* Flask: <http://flask.pocoo.org/>
* Getting Started With Flask: [https://scotch.io/tutorials/getting-started-with-flask-a-python-microframework#toc-file-structure](https://scotch.io/tutorials/getting-started-with-flask-a-python-microframework)
* Apache Configuration: <http://flask.pocoo.org/docs/1.0/deploying/mod_wsgi/>
* FLASK HELLO WORLD APP WITH APACHE WSGI: <https://www.bogotobogo.com/python/Flask/Python_Flask_HelloWorld_App_with_Apache_WSGI_Ubuntu14.php>
* Use mod\_wsgi to Run Python as a Web Application on CentOS 7: <https://www.1and1.com/cloud-community/learn/application/python/use-mod-wsgi-to-run-python-as-a-web-application-on-centos-7/>
* Sitio Flask Basico: <https://www.youtube.com/watch?v=yr-bbnl4d40>
* Proyecto Flask – Parte 1: <https://www.youtube.com/watch?v=mi_30rP2_nE>
* Proyecto Flask – Parte 2: <https://www.youtube.com/watch?v=I4vOESVdmXg>
* Desplegar Sitio Flask en Apache: <https://www.youtube.com/watch?v=cxIlQYyrpoU>