



EBook Gratis

APRENDIZAJE cherry.py

Free unaffiliated eBook created from
Stack Overflow contributors.

#cherry.py

Tabla de contenido

Acerca de.....	1
Capítulo 1: Empezando con cherrypy.....	2
Observaciones.....	2
Examples.....	2
Instrucciones de instalación.....	2
Precondiciones.....	2
Instalar.....	2
Hola mundo en CherryPy.....	3
Carga de archivos con CherryPy.....	3
server.py.....	3
webpage.html.....	4
cli.py.....	5
subir.txt.....	5
Subir desde el navegador.....	5
Subir desde la línea de comando.....	5
Creditos.....	6

Acerca de

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [cherry.py](#)

It is an unofficial and free cherry.py ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official cherry.py.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Capítulo 1: Empezando con cherrypy

Observaciones

Esta sección proporciona una descripción general de qué es cherrypy y por qué un desarrollador puede querer usarlo.

También debe mencionar cualquier tema grande dentro de cherrypy, y vincular a los temas relacionados. Dado que la Documentación para cherrypy es nueva, es posible que deba crear versiones iniciales de esos temas relacionados.

Examples

Instrucciones de instalación

Precondiciones

- Estas instrucciones suponen que tiene cualquier tipo de Linux, Unix, Mac con bash o Git-bash Windows.
- Windows: descargue e instale [Git-bash para Windows](#) , luego ejecute 'bash' desde la línea de comandos.
- Otros shells que no sean bash también están bien, simplemente reemplace el comando de `activate` continuación con `activate.csh` o Google: " [virtualenv activar su nombre de shell](#) ".

Antes de comenzar, compruebe que Python, virtualenv y pip están instalados:

- `$ python --version`
- `$ virtualenv --version`
- `$ pip --version`

Instalar

Cree un directorio con su web / aplicación, cree un entorno e instale el paquete CherryPy.

- `$ mkdir /develop/myapp/`
- `$ cd /develop/myapp/`
- `$ virtualenv venv`
- `$ source venv/bin/activate`
 - **En Windows en Git-bash:** `$ source venv/Scripts/activate`
- `(venv) $ pip install cherrypy`
- `(venv) $ python`

```
Python 3.5.2 ...
>>> import cherrypy
>>> cherrypy
```

```
<module 'cherrypy' from '... venv/site-packages/cherrypy/__init__.py'>
```

¡Felicidades! Ahora estás listo para tu primera aplicación CherryPy.

Hola mundo en CherryPy

Si tiene un virtualenv y CherryPy ya está instalado, cree un archivo `hello.py` :

```
#!/usr/bin/env python
# -*- coding: UTF-8 -*-

import cherrypy

class HelloWorld(object):
    @cherrypy.expose
    def index(self):
        return 'Hello World!'

    @cherrypy.expose
    def greet(self, name):
        return 'Hello {}'.format(name)

cherrypy.quickstart(HelloWorld())
```

Luego ejecute el archivo: `$ hello.py` o `$ python hello.py` . Deberías ver una salida similar a esta:

```
user@computer /develop/myapp $ python hello.py
[06/Nov/2016:05:58:44] ENGINE Listening for SIGTERM.
[06/Nov/2016:05:58:44] ENGINE Bus STARTING
[06/Nov/2016:05:58:44] ENGINE Set handler for console events.
CherryPy Checker:
The Application mounted at '' has an empty config.

[06/Nov/2016:05:58:44] ENGINE Started monitor thread '_TimeoutMonitor'.
[06/Nov/2016:05:58:44] ENGINE Started monitor thread 'Autoreloader'.
[06/Nov/2016:05:58:45] ENGINE Serving on http://127.0.0.1:8080
[06/Nov/2016:05:58:45] ENGINE Bus STARTED
```

- Para ver '¡Hola mundo!' dirija su navegador a <http://localhost:8080/>
- Para ver un saludo, vaya a <http://localhost:8080/greet?Name=John>

Carga de archivos con CherryPy

Este ejemplo consta de tres partes:

- `server.py` - Aplicación CherryPy que puede recibir y guardar un archivo.
- `webpage.html` : ejemplo de cómo cargar un archivo a `server.py` desde una página web.
- `cli.py` : ejemplo de cómo cargar un archivo en `server.py` desde una herramienta de línea de comandos.
- **Bonus** - `upload.txt` - archivo que `upload.txt` .

server.py

```
#!/usr/bin/env python
# -*- coding: UTF-8 -*-

import os
import cherrypy

config = {
    'global' : {
        'server.socket_host' : '127.0.0.1',
        'server.socket_port' : 8080
    }
}

class App:

    @cherrypy.expose
    def upload(self, ufile):
        # Either save the file to the directory where server.py is
        # or save the file to a given path:
        # upload_path = '/path/to/project/data/'
        upload_path = os.path.dirname(__file__)

        # Save the file to a predefined filename
        # or use the filename sent by the client:
        # upload_filename = ufile.filename
        upload_filename = 'saved.txt'

        upload_file = os.path.normpath(
            os.path.join(upload_path, upload_filename))
        size = 0
        with open(upload_file, 'wb') as out:
            while True:
                data = ufile.file.read(8192)
                if not data:
                    break
                out.write(data)
                size += len(data)
        out = ''
        File received.
        Filename: {}
        Length: {}
        Mime-type: {}
        ''' .format(ufile.filename, size, ufile.content_type, data)
        return out

if __name__ == '__main__':
    cherrypy.quickstart(App(), '/', config)
```

webpage.html

```
<form method="post" action="http://127.0.0.1:8080/upload" enctype="multipart/form-data">
```

```
<input type="file" name="ufile" />
<input type="submit" />
</form>
```

cli.py

Este ejemplo requiere un paquete de [solicitudes de Python](#) , sin embargo, el archivo se puede enviar al servidor en Python simple.

```
#!/usr/bin/env python
# -*- coding: UTF-8 -*-

import requests

url = 'http://127.0.0.1:8080/upload'
files = {'ufile': open('file.txt', 'rb')}

r = requests.post(url, files=files)

print(r)
print(r.text)
```

subir.txt

```
Hello! This file was uploaded to CherryPy.
```

Subir desde el navegador

- Ejecutar `$ server.py`
- Abra `webpage.html` en su navegador web.
- Después de seleccionar el archivo de su unidad y enviarlo, se guardará como `saved.txt` .

Subir desde la línea de comando

- Abra una consola y ejecute `$ server.py`
- Abra otra consola y ejecuta `$ cli.py`
 - Nota: el archivo de prueba `upload.txt` debe estar en el mismo directorio con `cli.py`
- El archivo `upload.txt` debe cargarse y guardarse como `saved.txt` .

Lea Empezando con [cherrypy en línea](https://riptutorial.com/es/cherrypy/topic/7696/empezando-con-cherrypy): <https://riptutorial.com/es/cherrypy/topic/7696/empezando-con-cherrypy>

Creditos

S. No	Capítulos	Contributors
1	Empezando con cherrypy	Community , dwich