



Instalación de Django en PythonAnywhere

Antes de comenzar esta tarea, ya debería haberse registrado en una cuenta de PythonAnywhere [🔗](#) e iniciar sesión en su cuenta. Debería poder completar todos los ejercicios de este curso utilizando una cuenta gratuita de PythonAnywhere.

Puede ver un tutorial en video [🔗](#) de esta tarea.

Nota: Si está enviando estas tareas al calificador automático, debe completar cada tarea y luego enviarla y obtener el crédito completo antes de pasar a la siguiente tarea. Debido a que las tareas se basan unas en otras, la aplicación que creó en el último paso del tutorial ya no aprobará los autoevaluadores anteriores.

Configuración de su entorno

Una vez que haya creado su cuenta PYAW, inicie un `bash` shell y configure un entorno virtual con Python 3.x y Django 3.

```
mkvirtualenv django3 --python=/usr/bin/python3.8
pip install django==3.2.5 ## this may take a couple of minutes
```

Tenga en cuenta que si sale y reinicia un nuevo shell en PythonAnywhere, necesita el siguiente comando para volver a su entorno virtual en el nuevo shell `bash`.

```
workon django3
```

Asegurémonos de que su django se instaló correctamente con el siguiente comando:

```
python -m django --version
# This should show something like 3.2.5
```

Instalación del código de muestra para DJ4E

También obtengamos una copia del código de muestra para DJ4E para que pueda ver el código de muestra a medida que avanza el curso e instalar algunas bibliotecas de software Django adicionales importantes usando `pip`.

```
cd ~
git clone https://github.com/csev/dj4e-samples
cd dj4e-samples
pip install -r requirements.txt
python3 manage.py check
```

Esta es la salida normal de ejecutar `check` :
DJ4E (https://www.dj4e.com)

```
When you want to use social login, please see dj4e-samples/github_settings-dist.py
Using registration/login.html as the login template
System check identified no issues (0 silenced).
```

Si ve un error de sintaxis

Si `check` identifica errores, no continúe con el resto de la tarea una vez que pueda ejecutarla `check` y no haya errores.

Si ves este error:

```
python manage.py check
File "manage.py", line 17
    ) from exc
      ^
SyntaxError: invalid syntax
```

No edite su archivo; el problema nunca está en ese archivo `manage.py`

Hay varias razones posibles para esto:

- Puede significar que no está ejecutando en el entorno virtual (`workon django3`) y está ejecutando por error Python 2 en lugar de Python 3. No hay *ningún error* : es una sintaxis válida de `manage.py` Python 3 que confunde a Python 2.
- Puede significar que ha editado su `manage.py` archivo y ha introducido un error de sintaxis. Eche un vistazo a esta captura de pantalla de un `manage.py` en funcionamiento [🔗](#) y compruebe si hay alguna diferencia. Mientras esté allí, puede cambiar la línea 1 de 'python' a 'python3' y la indicación de error de sintaxis desaparecerá. A menudo, los errores están en la sangría de las líneas 16 y 17. Verifique y haga coincidir exactamente la sangría en la captura de pantalla.

Cuando ejecutar 'verificar' funciona

Una vez realizadas las `check` obras:

```
python3 manage.py makemigrations
```

Esta es la salida normal de `makemigrations` :

```
When you want to use social login, please see dj4e-samples/github_settings-dist.py
Using registration/login.html as the login template
No changes detected
```

Entonces corre:

```
python3 manage.py migrate
```

Si está haciendo esto por primera vez, debería ejecutar algunas migraciones y crear un archivo `db.sqlite3` .
DJ4E (<https://www.dj4e.com>)

La `dj4e-samples` carpeta es material de referencia que puede utilizar a lo largo del curso. De vez en cuando podemos hacer cambios a esto y pedirle que haga una `git pull` para obtener la última versión del código.

Construyendo su aplicación

Ahora que tenemos tu Django configurado y has recuperado el código de muestra para DJ4E, construyamos tu primera aplicación en el shell PYAW:

```
cd ~  
mkdir django_projects  
cd django_projects  
django-admin startproject mysite
```

Debe editar el archivo `~/django_projects/mysite/mysite/settings.py` y cambiar la línea de hosts permitidos (alrededor de la línea 28) para que sea:

```
ALLOWED_HOSTS = [ '*' ]
```

Deje el valor **DEBUG** establecido en *True* : no estamos realmente "en producción" y si establece esto en *False* , no verá mensajes de error cuando cometa errores.

Edición de archivos en PythonAnywhere

Hay tres formas de editar archivos en su entorno de PythonAnywhere, desde la más fácil hasta la más genial. Solo tiene que editar el archivo de una de estas maneras.

(1) Vaya al panel principal de PythonAnywhere, busque archivos, navegue a la carpeta correcta y edite el archivo

```
/home/drchuck/django_projects/mysite/mysite/settings.py
```

(2) O en la línea de comando:

```
cd ~/django_projects/mysite/mysite/  
nano settings.py
```

Save the File by pressing 'CTRL-X', 'Y', and Enter

(3) No intente esta forma más difícil y genial de editar archivos en Linux sin un ayudante si es la primera vez que usa el `vi` editor de texto.

```
cd ~/django_projects/mysite/mysite/  
vi settings.py
```

Una vez que haya abierto `vi`, mueva el cursor hacia abajo hasta la `ALLOWED_HOSTS` línea, coloque el cursor entre las llaves y presione la `i` tecla para ir al modo 'INSERTAR', luego escriba su nuevo texto y presione la `esc` tecla cuando

haya terminado. Para guardar el archivo, escriba `:wq` seguido de `enter`. Si te pierdes pulsa `escape` `:q!` `enter` para salir del archivo sin guardar.

Si ya conoce algún *otro* editor de texto de línea de comandos en Linux, puede usarlo para editar archivos. En general, encontrará que a menudo es más rápido y más fácil realizar pequeñas modificaciones en los archivos en la línea de comandos en lugar de una interfaz de usuario de pantalla completa. Y una vez que comience a implementar aplicaciones reales en entornos de producción como Google, Amazon, Microsoft, etc., todo lo que tendrá será una línea de comandos.

Ejecutando su aplicación

Ahora que hemos creado su primera aplicación, debemos decirle a PythonAnywhere dónde buscar para ejecutar su aplicación como un servidor web para que pueda probarla.


En la interfaz web de PYAW, navegue hasta la `Web` pestaña para crear una nueva aplicación web. No necesita actualizar su cuenta; le brindan una aplicación como *drchuck.pythonanywhere.com*; use esta aplicación gratuita para el curso.

Al crear la nueva aplicación, *no* cree una "aplicación Django"; en su lugar, seleccione `manual configuration Python 3.8`. Una vez que se crea la aplicación web, debe realizar algunos cambios en la configuración de la aplicación web y su aplicación.

```
Source code: /home/drchuck/django_projects/mysite
Working directory: /home/drchuck/django_projects/mysite

Virtualenv: /home/drchuck/.virtualenvs/django3
```

Reemplace `drchuck` con su cuenta en PythonAnywhere.

Tenga en cuenta que una vez que haya configurado su Virtualenv, tiene un enlace muy conveniente titulado "Iniciar una consola en este virtualenv": esta es una excelente manera de abrir consolas para que nunca tenga que escribir "workon django3" y se asegura El entorno está instalado y configurado correctamente. Imagen de muestra 

Luego edite el *archivo de configuración WSGI* y coloque el siguiente código en él. **Asegúrese de eliminar el contenido existente** del *archivo de configuración de WSGI* y reemplácelo por completo con el texto a continuación. Esto es ligeramente diferente de la muestra en el tutorial de PythonAnywhere.

```
import os
import sys
```

DJ4E(https://www.dj4e.com)

```
path = os.path.expanduser('~/.django_projects/mysite')
if path not in sys.path:
    sys.path.insert(0, path)
os.environ['DJANGO_SETTINGS_MODULE'] = 'mysite.settings'
from django.core.wsgi import get_wsgi_application
from django.contrib.staticfiles.handlers import StaticFilesHandler
application = StaticFilesHandler(get_wsgi_application())
```

Una vez completada la configuración anterior, regrese a la parte superior de la pestaña Web de PYAW, Reload su aplicación web, espere unos segundos y verifique que esté funcionando:

```
http://(your-account).pythonanywhere.com/
```

Aquí hay una muestra [🔗](#) de cómo debería verse la página resultante.

Solo como nota, *nunca* ejecuta el `runserver` comando en PythonAnywhere.

```
python3 manage.py runserver
```

Si intenta hacerlo `runserver` en PythonAnywhere, verá un mensaje de error como este

```
00:26 ~/.django_projects/mysite (master)$ python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...
System check identified no issues (0 silenced).
September 14, 2021 - 00:26:16
Django version 2.2.7, using settings 'mysite.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
Error: That port is already in use.
00:26 ~/.django_projects/mysite (master)$
```

Esto *nunca* funcionará en PythonAnywhere. Ejecuta / reinicia su servidor en PythonAnywhere usando el botón "recargar" en su web pestaña. Entonces, si está leyendo las instrucciones de Django que dicen hacer un `runserver`, en su lugar haga un `check` y luego vuelva a cargar la aplicación en la interfaz de usuario web de PythonAnywhere.

Agregar su aplicación de encuestas

En este punto, puede agregar la aplicación de encuestas del primer tutorial de Django. Estamos saltando a la mitad de este tutorial porque la primera parte del tutorial es instalar y configurar Django en general.

El primer paso es hacer la `polls` solicitud:

```
cd ~/.django_projects/mysite
python3 manage.py startapp polls
```

Continúe siguiendo los pasos descritos en el tutorial de Django [🔗](#) . hasta llegar a la parte donde el tutorial te dice que **DJ4E (https://www.dj4e.com)** ejecute este comando:

```
python3 manage.py runserver      # <-- Never run this on pythonanywhere
```

No ejecute el comando en PythonAnywhere . `runserver` En su lugar, ejecute el siguiente comando:

```
python3 manage.py check
```

Realiza `check` una verificación de errores de sintaxis y lógica en su aplicación Django. Es más fácil corregir errores en la línea de comandos.

Y cuando no haya errores, haya terminado con el tutorial de Django, regrese a estas instrucciones y navegue a la web pestaña en Python en cualquier lugar y `Reload` su aplicación y luego pruebe su aplicación navegando a:

```
(your-account).pythonanywhere.com/polls
```

Deberías ver una línea que se parece a:

```
Hello, world. You're at the polls index.
```

En el futuro, cada vez que hagamos cambios en nuestra aplicación, deberíamos ejecutar

```
python3 manage.py check
```

en el shell, y cuando no muestre errores, navegue hasta `Web` , presione `Reload` y luego vaya a su sitio web para probar sus cambios. Este patrón de cambio, recarga y prueba se convertirá en una segunda naturaleza después de un tiempo.

Posibles errores

Si su aplicación pasó un `check` error pero no se puede cargar o volver a cargar, es posible que reciba un mensaje de error similar a este [🔗](#) .

Si recibe un error, deberá revisar los registros de errores en la `Web` pestaña de PythonAnywhere:

Log files:

The first place to look if something goes wrong.

Access log: drchuck.pythonanywhere.com.access.log

Error log: drchuck.pythonanywhere.com.error.log

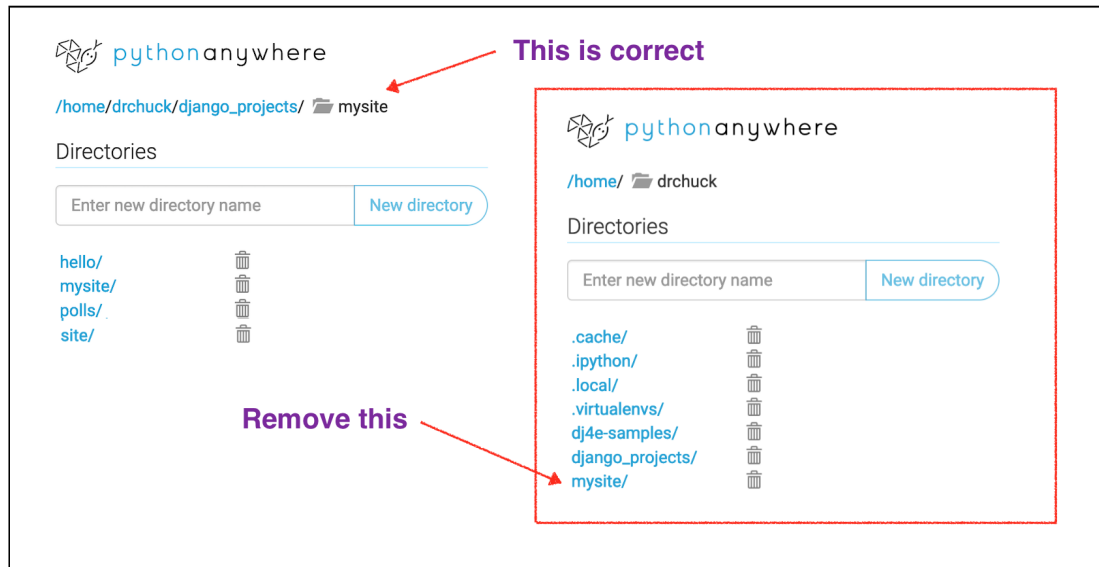
Server log: drchuck.pythonanywhere.com.server.log

Log files are periodically rotated. You can find old logs here: </var/log>

Primero verifique el error registro y luego verifique el server registro. Asegúrese de desplazarse por los registros hasta el final para encontrar el error más reciente.

¿Tiene dos carpetas mysite?

Por alguna razón, los estudiantes que terminan esta tarea a menudo terminan haciendo su mysite carpeta dos veces. Terminan con una carpeta en su directorio de inicio y en su django_projects subcarpeta.



Es una buena idea eliminar la carpeta adicional en su directorio de inicio después de asegurarse de que el código correcto esté en su `django_projects/mysite` carpeta. Es realmente frustrante tener dos carpetas y hacer un montón de trabajo en una de las carpetas que en realidad no afecta la aplicación en ejecución.

Así que también podrías limpiar esto de inmediato si lo ves.

Más información sobre su archivo `manage.py`

Es posible que obtenga muchos errores cuando ejecute `manage.py` : los errores nunca están en el archivo en sí y nunca necesitará editarlo `manage.py` para solucionar un problema con su programa.

A veces si corres

```
python manage.py ....
```

A veces incluso dice que hay un error de sintaxis `manage.py` así:

```
$ python manage.py check
File "manage.py", line 16
    ) from exc
    ^
SyntaxError: invalid syntax
```

Esto se debe a que está ejecutando Python 2.x y no Python 3.x. El archivo no tiene una sintaxis válida para python2. Si está ejecutando python2, probablemente significa que no está en el entorno virtual correcto. Si comprueba la versión de python:

```
$ python --version
Python 2.7.12
```

Y es 2.x, tiene problemas más grandes que deben solucionarse primero.

Si abre `manage.py` en el editor de archivos, mostrará un error de sintaxis en la línea 16; esto se debe a que está viendo el archivo como Python 2 (suena familiar).

Si esto le molesta, puede cambiar la primera línea del archivo (no cambie nada más) para que sea:

```
#!/usr/bin/env python3
```

Esto se llama "Hashbang" y es un comentario con formato especial que indica qué tipo de código contiene el archivo.

Y si está leyendo esto después de cometer un error y editar su `manage.py` - aquí hay una copia nueva del archivo que puede usar:

Copia nueva de `manage.py` (`dj4e_install/manage.py.txt`)

Empezar de nuevo

Si ha seguido las instrucciones y simplemente no funciona y desea comenzar de nuevo desde el principio de esta tarea, estos son los pasos para aclarar las cosas:

- Elimine todas las consolas en ejecución en la pestaña Consolas de www.pythonanywhere.com
- Abra una nueva consola **bash** desde la pestaña Consolas. No ejecute el comando **workon** y ejecute los siguientes comandos:

```
cd ~
rm -rf .virtualenvs/django3
rm -rf dj4e-samples
rm -rf django_projects
```

- Luego vaya a la pestaña Web en www.pythonanywhere.com y vuelva a cargar su aplicación. Puede darte errores, esto está bien. Solo queremos asegurarnos de que no queden procesos pendientes de archivos antiguos.

Luego cierre su consola y elimínela en la pestaña Consolas y vaya al principio de este folleto y comience de nuevo.

No eliminamos ninguno de los cambios de configuración en la pestaña Web; por lo tanto, cuando vuelva a crear todos los archivos, es posible que partes de la pestaña Web comiencen a funcionar cuando vuelva a cargar su aplicación.

