

Como instalar Django en tu hosting

Aunque esta guía es válida para cualquier instalación de django, está pensada para usuarios de guebs.com, especialmente en lo que se refiere a las rutas de archivos y aplicaciones

Para seguir esta guía será necesario que dispongas de acceso ssh a tu cuenta de hosting. Sigue esta guía sobre [Como acceder vía SSH a tu cuenta](#)

1 - Crear la aplicación desde el panel de Hosting

Para instalar Django lo primero que debes hacer es crear la aplicación desde el panel de hosting del plan de alojamiento.

El nombre que pongas como nombre de aplicación será el nombre de tu aplicación django, supondremos que has elegido **miapp** como nombre de aplicación.

Crear nueva aplicación python

Nombre de la aplicación:

 Se usará como nombre de directorio de la aplicación, nada de caracteres raros.

Dominio:

Stack:

 Se permite un único stack por cuenta de hosting, el asignado a esta es **python-3.4**

La elección de nombre **miapp** implica que el directorio de tu aplicación Django será **python/miapp**

2 - Instalar paquete Django

Para instalar la última versión de Django ejecuta el siguiente comando vía SSH:

```
<br />
pip install django --user<br />
```

Para comprobar que se ha instalado correctamente ejecuta el siguiente comando:

```
<br />
python -m django --version<br />
```

3 - Crear el proyecto Django

Tal y como hemos adelantado, el directorio de tu aplicación deberá ser **python/miapp**. Por tanto, para crear la aplicación debes ejecutar el siguiente vía SSH:

```
<br />
cd python<br />
rm -fr miapp<br />
django-admin startproject miapp<br />
```

4 - Configurar el archivo WSGI

Suponiendo que el nombre usuario de tu hosting es **usuario_cpanel** y dado que la aplicación es **miapp**, crea el archivo **miapp/passenger_wsgi.py** con el siguiente contenido:

```
import sys

## Necesario con ambos casos.
sys.path.insert(0, "/home/usuario_cpanel/python/miapp")

import os
os.environ['DJANGO_SETTINGS_MODULE'] = 'miapp.settings'

from django.core.wsgi import get_wsgi_application
```

```
application = get_wsgi_application()
```

5 - Configurar Django

Ahora debemos editar el archivo **python/miapp/miapp/settings.py** y dejar las opciones de rutas estáticas de la siguiente manera:

```
STATIC_URL = '/'  
STATIC_ROOT= '/home/usuario_cpanel/python/miapp/public/'  
ALLOWED_HOSTS = 'www.midominio123.com'
```

En el servicio de guebs.com los archivos estáticos están mapeados contra el directorio public que debes colocar en el directorio **public** de tu aplicación, que en este ejemplo sería: **/home/usuario_cpanel/python/miapp/public**

Una vez establecidas las directivas anteriores, sólo debes usar la herramienta de copiado de archivos estáticos al directorio establecido:

```
<br />  
cd miapp<br />  
python manage.py collectstatic<br />
```

Que copiará de manera automática los archivos a tu public. Cada vez que realices un deployment de tu aplicación es recomendable que lo reejecutes de nuevo.

6 - El primer «hola mundo»

Creamos el archivo **python/miapp/miapp/views.py** con una primera vista básica:

```
from django.http import HttpResponse  
  
def index(request):  
    return HttpResponse("Hola Mundo.")
```

Y posteriormente mapeamos la URL a través del **python/miapp/miapp/urls.py**:

```
from django.conf.urls import include,url  
from django.contrib import admin  
from . import views  
  
urlpatterns = [  
    url(r'^admin/', admin.site.urls),
```

```
    url(r'^$', views.index, name='index'),  
]
```

Hecho esto, al llamar a tu dominio encontrarás tu primer 'hola mundo' en Django.

7 - Configuración de la base de datos

Para este ejemplo configuraremos MySQL, aunque en guebs.com disponemos de PostgreSQL también.

Primero, instalamos el paquete **mysqlclient** de Python:

```
<br />  
pip install mysqlclient &#8211;user<br />
```

Luego configuramos los datos de conexión a MySQL en el archivo **settings.py** de la aplicación:

```
DATABASES = {  
    'default': {  
        'ENGINE': 'django.db.backends.mysql',  
        'NAME': 'nombre_basedatos',  
        'USER': 'usuario_base_datos',  
        'PASSWORD': 'password',  
        'HOST': 'localhost',  
        'PORT': '3306',  
    }  
}
```

Por último, sólo debemos realizar el migrate:

```
<br />  
python manage.py migrate<br />
```

Si prefieres usar PostgreSQL en lugar de MySQL sigue las indicaciones de la guía Como instalar Django en tu hosting usando PostgreSQL.

8 - Configuración del acceso administrador

inicial

Debemos ejecutar el siguiente comando:

```
<br />
python manage.py createsuperuser<br />
```

A partir de este momento podremos loguearnos en la zona de administración del proyecto, que suele estar por defecto en `tudominio.com/admin`

9 - Creación de la primera aplicación dentro del proyecto

Para crear una aplicación dentro de un proyecto existente usaremos la opción `startapp`. Por ejemplo con el siguiente comando:

```
<br />
python manage.py startapp nuevo<br />
```

Que creará una carpeta con la aplicación en la raíz del proyecto. Igual que en la aplicación principal, crearemos un `views.py` y un `urls.py` con el contenido de una vista por defecto

python/miapp/nuevo/views.py

```
from django.shortcuts import render

# Create your views here.
from django.http import HttpResponse

def index(request):
    return HttpResponse("Hello, world. Esto es nuevo.")
from django.conf.urls import url
```

python/miapp/nuevo/urls.py

```
from django.conf.urls import url, include
from django.contrib import admin

from . import views
```

```
urlpatterns = [  
    url(r'^$', views.index, name='index'),  
]
```

Y finalmente mapeamos la url que queremos a la nueva aplicación, en el archivo urls.py de la aplicación principal:

python/miapp/miapp/urls.py

```
urlpatterns = [  
    url(r'^admin/', admin.site.urls),  
    url(r'^nuevo/', include('nuevo.urls')),  
    url(r'^$', views.index, name='index'),  
]
```

10 - Reiniciar Aplicacion Python

Una vez modificado todo, necesitamos reiniciar el servidor para que nos muestre los cambios realizador.

en el directorio python/miapp/

```
<br />  
mkdir tmp<br />  
touch tmp/restart.txt<br />
```

Pasados unos segundos la aplicación se habrá reiniciado y veremos nuestro hola mundo en django.