

HINTS

CONFIGURACIONES RELEVANTES

`settings.py` Django, por defecto, mantiene todas las configuraciones del proyecto en un solo archivo monolítico. Esto es lo que se crea cuando se genera un nuevo proyecto utilizando las herramientas de línea de comandos de Django. Es simple.

Y ese es el problema. El enfoque monolítico no encaja bien con la forma en que desarrollamos. Una configuración pertinente puede basarse en lo siguiente:

- Definir configuraciones no constantes y secretas, como variables de entorno.
- En entornos de desarrollo, defina las variables de entorno en un archivo `.env`, y agréguese `.env` a `.gitignore`.
- Utilice los mecanismos del proveedor de la nube para definir variables de entorno (secretas) para los entornos de control de calidad, ensayo y producción.
- Utilice un único archivo `settings.py` que se configura a sí mismo a partir de las variables de entorno. Combinar la arquitectura y la configuración del proyecto (por ejemplo: aplicaciones instaladas y claves secretas).

DJANGO-ENVIRON CON LISTAS DE VARIABLES DE ENTORNO EN ARCHIVOS .ENV

`django-environ` es el paquete de Python que le permite usar la metodología de doce factores para configurar su aplicación Django con variables de entorno.

Uno de los principales casos de uso list sería: `ALLOWED_HOSTS`.

Supongamos que tenemos hosts en nuestro `settings.py`:

```
1 # settings.py ...
2 ALLOWED_HOSTS = [
3     "example.com",
4     "awesomedomain.com",
5     "stagingdomain.com",
6     "10.0.0.2",
7     "198.162.44.72",
8 ]
```

¿Cómo podemos mover este codificado list/array a un `.env` archivo y usarlo con `django-environ`?

SOLUCIÓN

Primero agregue **env.list** en su settings:

```
1 # settings.py
2 ALLOWED_HOSTS = env.list("ALLOWED_HOSTS")
```

Luego, agregue sus hosts con comas separadas en **.env** y las que se encuentran en la raíz de su proyecto, o en una ubicación adecuada:

```
1 # .env
2 ...
3 ALLOWED_HOSTS=example.com,awesomedomain.com,stagingdomain.com,10.0.0.2,
4 198.162.44.72
```

EXTRA

Su archivo de variables de entorno **.env**, puede ser cualquier cosa y estar en cualquier lugar **.envs/.local**, **.env.prod** siempre que lo configure correctamente:

```
1 # settings.py import os
2 import environ
3 # Set the project base directory BASE_DIR =
4 os.path.dirname(os.path.dirname(os.path.abspath(__file__)))
5 # Initiate related env class env = environ.Env()
6 # Take environment variables from .env.dev file
7 environ.Env.read_env(os.path.join(BASE_DIR, '.env.dev'))
```

- Para más información, puedes visitar los siguientes enlaces:
<https://tuts-station.com/how-to-set-up-environment-variables-in-django.html>
<https://apibakery.com/blog/django/django-settings-howto/>