Ficheros Hook en Virtualenvwraper

18 de Mayo de 2017 a las 01:20

Virtualenvwrapper es un conjunto de extensiones de *virtualenv* que permite tener un entorno aislado por cada proyecto, con sus paquetes y configuración.

Dentro de esta herramienta existen una serie de ficheros denominados ***Hook***, que se activan cuando ocurren una determinada acción, cada fichero esta relacionado con una acción. Estos ficheros pueden contener una serie de comandos que seran ejecutados cuando sean activados.

Estos ficheros tienen dos ambitos:

- ***Ámbito global:*** dentro de la carpeta *.virtualenvs*, y afectara a todos los entornos que tengamos.
- ***Ámbito local:*** dentro de la carpeta *.virtualenvs,* tendremos un carpeta por cada proyecto y dentro existirá una carpet *bin,* ruta *.virtualenvs/nombreProyecto/bin*.

La carpeta *.virtualenvs* es creada por defecto en la carpeta de usuario (HOME), */home/nombreusuario*,donde ejecutas *virtualenvwrapper*.

Ficheros Hook

Como he indicado anteriormente hay varios ficheros de ese tipo y cada uno relacionado con una acción de activación,sus nombres ayudan bastante para saber con que acción se activan. Tenemos principalmente dos tipos de *Hooks*, los *pre* que son ejecutados antes de la acción y los *post* que son ejecutados después de la acción.

Los ficheros son los siguientes:

- ***Initialize:*** se ejecuta cuando virtualenvwrapper.sh es ejecutado.
- ***Postactivate***: se ejecuta después de ejecutar el comando *workon,*en virtualenvwrapper, que permite activar un entorno virtual.
- ***Postdeactivate***: se ejecuta después de ejecutar el comando deactivate, para salir de un entorno virtual.
- **Postmkproject**: se ejecuta después de crear un nuevo entorno y situado en su directorio.
- **Postmkvirtualenv**: se ejecuta después de ejecutar *mkvirtualenv*, para crear un entorno virtual.
- **Postrmvirtualenv**: se ejecuta después de ejecutar *rmvirtualenv*, para borrar un entorno virtual.
- **Preactivate**: se ejecuta antes de activar, *workon,* un entorno virtual
- **Predeactivate**: se ejecuta antes de salir de un entorno virtual,*deactivate.*
- **Premkproject**: se ejecuta despues de crear un entorno y antes de situarse en el directorio del proyecto.

- **Premkvirtualenv**: se ejecuta antes de crear un entorno.
- **Prermvirtualenv**: se ejecuta antes de borrar un entorno.

Todos estos ficheros pueden ejecutar uno o varios comando cuando son activados.

Ejemplos

Aqui mostraré para que utilizo los hooks y que acciones ejecuto, aunque está enfocado para un proyecto Django, con cualquier aplicación de Python se pueden utilizar los ficheros *hooks*.

Postmkvirtualenv

Cuando creo un nuevo entorno, *mkvirtualenv,* quiero que la carpeta de trabajo este en un directorio específico donde almaceno todos los entorno que utilizo y que después se situe en esa carpeta.

Utilizo el fichero *postmkvirtualenv,* dentro de un ámbito global, para todos los entornos.

```
proj_name=$(basename $VIRTUAL_ENV)
mkdir $HOME/.virtualenvs/Proyectos/$proj_name
add2virtualenv $HOME/.virtualenvs/Proyectos/$proj_name
cd $HOME/.virtualenvs/Proyectos/$proj_name
```

Primero almaceno en una variable el nombre del entorno que voy a crear.

Segundo, creo una carpeta en el directorio especificado con el nombre del entorno.

Tercero, añado el entorno creado con la ruta que he especificado en la línea anterior.

Por último, me situo en el directorio donde he creado el entorno.

Postactivate y Predeactivate

En los proyectos de Django cuando despliegas en producción no es recomendable poner variables de entorno con información sensible(contraseñas, *secret_key*, direcciones de correo, direcciones de host...)en el fichero de settings.

En mi caso utilizo esos ficheros hook para exportar las variables cuando activamos el entorno y eliminar las variables cuando desactivamos el entorno.

Empecemos por *Postactivate.*

```
export SECRET_KEY='clave correspondiente'

#email formulario
export EMAIL_HOST='direccion smtp'
export EMAIL_HOST_USER='correo'
```

```
export EMAIL_HOST_PASSWORD='contraseña'
```

Como vemos hay una serie de variables que exportamos, SECRET_KEY y los datos para el envío de correos en un formulario de contacto.

En el archivo settings.py de la aplicación Django se llamarian a esas variables ,hay diversas formas de realizar esta tarea, de la forma que yo lo hago, aquí.

El otro fichero,*predeactivate*, lo utilizo para eliminar esas variables de entorno, cuando desactivo el entorno.

```
unset SECRET_KEY
#formulario contacto
unset EMAIL_HOST
unset EMAIL_HOST_USER
unset EMAIL_HOST_PASSWORD
```

Conclusión

Este tipo de ficheros *Hook*, permiten automatizar ciertas tareas en el desarrollo de código, saber utilizarlo permite facilitar el trabajo a un desarrollador. Este artículo solo pretendia ser una pequeña introducción a este tipo de ficheros y su uso con *virtualenvwrapper,* otra herramienta muy útil.