

zc.buildout para desarrolladores Plone

Emanuel Sartor
emanuel@menttes.com

¿Qué es un buildout?

¿Qué es un buildout?

La palabra **buildout** hace referencia a una descripción de un conjunto de partes y al software para crear y ensamblar esas partes.

A menudo se usa informalmente para hacer referencia a un sistema instalado basado en la definición de un buildout.

¿Qué es un buildout?

Por ejemplo, si estamos creando una aplicación llamada Foo, entonces ***el buildout de Foo*** será la colección de configuraciones y aplicaciones específicas que permiten crear una instancia de la aplicación.

Informalmente podríamos referirnos a tal instancia como el buildout de Foo.

¿Qué es un buildout?

zc.buildout es una herramienta que nos permite definir buildouts. Estas son sus principales características:

- Permite definición de buildouts de forma declarativa
- Basado en python
- Orientado a desarrollador
- Repetible
- Es fácil trabajar con *eggs*

Todo muy lindo...

**¿Pero cómo me ayuda esto en el desarrollo
Plone?**

Desarrollo Plone

Para configurar un entorno de desarrollo, usualmente los siguientes pasos son requeridos:

- Descargar, compilar e instalar Zope
- Crear una instancia Zope y configurarla
- Descargar e instalar Plone
- Instalar productos y eggs adicionales requeridos por mi proyecto
- Agregar a la instancia productos y paquetes que estoy desarrollando

Desarrollo Plone

Todos estos pasos se pueden realizar manualmente, pero esto genera una serie de problemas:

- Lento para generar nuevos entornos de desarrollo.
- Cada desarrollador es responsable de crear su entorno de desarrollo, y algunas diferencias sutiles en la configuración puede desembocar en errores en el código.

Todo esto puede ser automático usando `zc.buildout`.

Mmm... suena bien, pero seguro que tengo que leer cientos de tutoriales y un par de libros para usar `zc.buildout`.

¿Cómo harías todo eso con `zc.buildout`?

Desarrollo Plone con **zc.buildout**

Vamos a empezar creando un buildout que descargue Zope, lo compile, lo instale, cree una instancia y la configure.

```
$ mkdir buildout
```

```
$ cd buildout
```

```
$ wget http://svn.zope.org/zc.buildout/trunk/bootstrap/bootstrap.py
```

```
$ vim buildout.cfg
```

buildout.cfg

[buildout]

```
parts =  
    zope2  
    instance
```

[zope2]

```
recipe = plone.recipe.zope2install  
url = http://www.zope.org/Products/Zope/2.10.5/Zope-2.10.5-final.tgz
```

[instance]

```
recipe = plone.recipe.zope2instance  
zope2-location = ${zope2:location}  
user = admin:admin  
http-address = 8080  
debug-mode = on  
verbose-security = on
```

buildout.cfg

La sección [buildout] es el punto de inicio del archivo. Este lista un número de **partes** las cuales serán configuradas más adelante en el archivo.

buildout.cfg

Cada parte tiene una receta (**recipe**) asociada, la cual es el nombre de un egg que sabe cómo realizar una tarea determinada, por ejemplo, compilar Zope o crear una instancia.

buildout.cfg

Una receta toma típicamente algunas opciones de configuración.

Hasta ahora tenemos...

- **bootstrap.py**: es un script que va a instalar setuptools y zc.buildout en el directorio buildout que creamos sin interferir con los paquetes instalados globalmente.
- **buidout.cfg**: la definición de un buildout que instala Zope y configura una instancia.

zc.buildout en acción

```
$ python2.4 bootstrap.py
```

```
$ bin/buildout
```

[esperamos un rato]

```
$ bin/instance fg
```

Listo, ya podemos acceder a nuestra instancia Zope 2.10.5 apuntando nuestro navegador a <http://localhost:8080>

zc.buildout en acción

Supongamos que por error instalamos Zope 2.10.5, pero que en realidad necesitábamos instalar la versión 2.10.4. Supongamos también que queremos que la instancia ocupe el puerto 8090 en lugar del 8080.

En ese caso, en el archivo `buildout.cfg` modificamos la URL de la cuál se descarga Zope, así también como el número de puerto y luego volvemos a ejecutar:

```
$ bin/buildout
```

Bien... comienzo a sentirme mejor.

¿Cómo instalamos Plone?

buildout.cfg (primera parte)

[buildout]

```
parts =  
    zope2  
    instance  
    plone  
eggs =
```

[plone]

```
recipe = plone.recipe.plone  
# recipe = plone.recipe.plone==3.0.1
```

[zope2]

```
recipe = plone.recipe.zope2install  
url = ${plone:zope2-url}
```

buildout.cfg (segunda y última parte)

[instance]

```
recipe = plone.recipe.zope2instance
zope2-location = ${zope2:location}
user = admin:admin
http-address = 8080
debug-mode = on
verbose-security = on
eggs =
    ${buildout:eggs}
    ${plone:eggs}
zcml =

products =
    ${plone:products}
```

buildout.cfg

Las recetas pueden exponer variables para ser usadas por otras partes en la configuración del buildout.

Por ejemplo, en [zope2] se hace referencia a `${plone:zope2-url}`, que es una variable emitida por la parte [plone] y que tiene como valor el URL del Zope adecuado para la versión de Plone a instalar.

buildout.cfg

Se puede especificar explícitamente la versión de la receta a utilizar (recordar que las recetas son eggs). Un ejemplo de esto es el comentario en la sección [plone] en el que se especifica que se usará la versión 3.0.1 de la receta plone.recipe.plone.

Muchas gracias