

Getting Started with Zope 3

Róman Joost

20th October 2009

1 Zope

The Z Object Publishing Framework (Zope) was the first application server written in Python, which made it possible to manage and publish objects through the web. Since 1998, the Zope framework has evolved, becoming more sophisticated and complex, but also less scalable. Developers found that the more complex an application became the less portable and manageable the code base was due to the multiple inheritance. New development paradigms were used to create a component architecture which tried to split up responsibilities. The component architecture allows the creation of reusable components which can be tested and distributed independently; functioning more like a toolbox.

2 A bit of Theory

One disadvantage of Zope 2 was the difficulty of managing high-complexity applications with multiple-inheritance. This resulted in less reusable code. The component architecture in Zope 3 solved this problem using latest software development techniques[2] to spread the responsibilities between objects. Unfortunately, these major improvements in software architecture created an equally major learning curve for developers.

3 Bootstrapping

Before we get started, let me give some advice to any people new to Zope 3; have a look at Grok[3]. Grok writes most of the configuration to glue together components in Python whilst leaving you with full access to the component architecture.

For the rest of us, let's get cracking. Bootstrap your Zope 3 application with zope-project:

```
1 $ zopeproject --no-buildout
```

The reason why I used the `--no-buildout` option is because most users want to commit their newly created project before running a buildout – which creates additional directories used in the sandbox that are not necessary to be held in a repository.

If you don't have `zopeproject` installed, use `easyinstall` to install it from `pypi`¹.

4 Orientation

A quick orientation: `zopeproject` sets up your project with the `paste` http server. It's good for bootstrapping a project, although I think if the project is getting a bit more serious you'll want to switch to other http servers like `twisted` using the `ZDaemon`.

Let's dive into the project folder. Have a quick look at the `*.zcml` files. The `configure.zcml` configures your application, whereas the `ftesting.zcml` configures your functional tests.

Now that's covered, it's time to build.

5 Start up

I don't want to present a "how-to" here on developing in Zope 3. I want to point out that you have to configure each model you write with ZCML directives. The ZCML is used as a glue between Python and the component architecture. It allows you to plug-in and plug-out components as well as configuring them with parameters.

If you want to dive into programming with Zope 3, check out the information published on the web^[5]. I can recommend the book from Philipp von Weitershausen, "Web Component Development with Zope 3"^[4].

6 Problems

6.1 Different ZCML options

Sometimes tests differ in behaviour from actually using the application. If this occurs, check if your `site.zcml` differs from your `ftesting.zcml`.

6.2 Forbidden Errors

Every Zope object is wrapped in a security proxy. The access to that object is configured via ZCML by the developer. By default though, the access is *forbidden*. If you run into an error like this:

```
Error type: zope.security.interfaces.ForbiddenAttribute
2 Error object: ('description', <Model object at 0xb62ad8ec>)
```

¹<http://pypi.python.org/pypi/zopeproject>

your model lacks the security declarations. Fix it by using the ZCML directives `class`² and `require`³.

6.3 Unauthorised

An unauthorised error is always issued if the user who is accessing an object in Zope doesn't have the correct permission to do so. Those problems are harder to fix. You have to figure out why the user doesn't have the permissions:

- Check the ZCML files, especially `site.zcml` and `ftesting.zcml`, if the authentication setup including the principals reflects what you intend to do.
- If the error occurs in a test, check the test setup. Maybe you forgot to provide the necessary credentials.
- If it happens with the `zope.testbrowser`, check if the `handleErrors` attribute of the `testbrowser` instance is set before the browser tries to login.

6.4 Subscribers are executed twice

This could be difficult to spot, but it can happen if the configuration file becomes bigger than usual. Check if you registered the subscriber more than once.

References

- [1] Zope Component Architecture Approach; Zope Wiki, <http://wiki.zope.org/zope3/ComponentArchitectureApproach>
- [2] Software Design Patterns, http://en.wikipedia.org/wiki/Software_design_patterns
- [3] Grok Project Homepage, <http://grok.zope.org>
- [4] Web Component Development with Zope 3, Philipp von Weiterhausen, <http://www.amazon.com/gp/product/354076447X?ie=UTF8&tag=worldcookery-20&linkCode=as2&camp=1638&creative=6742&creativeASIN=3540338071K>
- [5] Zope 3 in 30 minutes, Baiju M. <http://zissue.berlios.de/z3/Zope3In30Minutes.html>

²Zope 3 API documentation http://apidoc.zope.org/++apidoc++/ZCML/http_co__sl__sl_namespaces.zope.org_sl_zope/class/index.html

³Zope 3 API documentation; require http://apidoc.zope.org/++apidoc++/ZCML/http_co__sl__sl_namespaces.zope.org_sl_zope/require/index.html