repoze.BFG

ROALD DAHL

ILLUSTRATED BY QUENTIN BLAKE

The BFG

The Big Friendly Giant

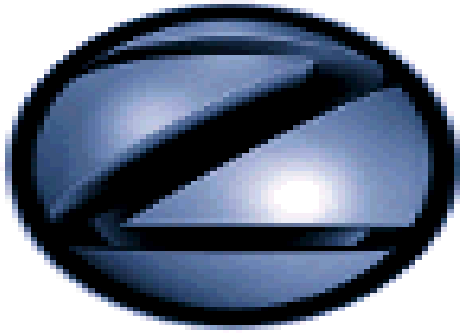
Yes, *another* web framework

repoze.bfg is a small, fast, down-to-earth, open source Python web development framework.

repoze.bfg

- WSGI-based deployment: Paste and mod_wsgi compatible.
- Runs under Python 2.4, 2.5, and 2.6.
- Runs on UNIX, Windows, Jython and Google App Engine.
- Chris McDonough, from Agendaless Consulting, is the primary developer.

BFG was inspired by



ZOPE

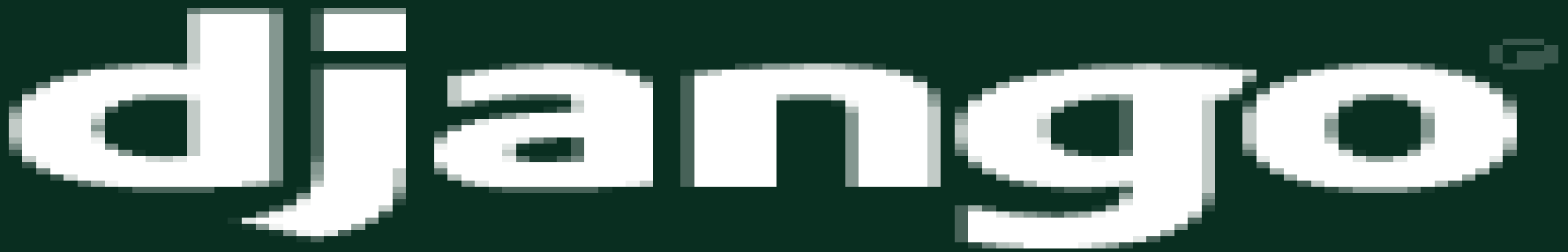
z object publishing environment

Did you say Zope?

- Like Zope, repoze.bfg applications can be configured declaratively and are easily extendable.
- We use declarative security and traversal.
- However, repoze.bfg does not force you to “drink the Kool-aid”. No ZODB, no Zope Component Architecture are required.
- It doesn't depend on the “everything but the kitchen sink” Zope collection of packages.

Pylons

- The URL dispatch mechanism of repoze.bfg was inspired by Pylons' Routes system.
- Like Pylons, repoze.bfg is mostly policy free.
- In fact, repoze.bfg is very similar to Pylons in scope and size.
- Unlike Pylons, repoze.bfg resolves a URL into a context before calling a view and thus allows context-sensitive declarative security for individual objects (yes, the Zope legacy).



- The concepts of view and model are used very similarly to the way Django uses them.
- Django has a great culture of documentation, we want repoze.bfg to have this as well.
- Unlike Django repoze.bfg is not a “full stack” framework. It makes no assumptions about filesystem layout or persistence mechanisms.

BFG core values

- Simplicity
- Fun
- Minimalism
- Documentation

BFG core values

- Speed
- Familiarity
- Trustability
- Openness

Simplicity: Pay only for what you eat

A complete BFG app

```
1  from webob import Response
2  from paste.httpserver import serve
3  from repoze.bfg.configuration import Configurator
4
5  def hello_world(request):
6      return Response('Hello world!')
7
8  def goodbye_world(request):
9      return Response('Goodbye world!')
10
11 if __name__ == '__main__':
12     config = Configurator()
13     config.begin()
14     config.add_view(hello_world)
15     config.add_view(goodbye_world, name='goodbye')
16     config.end()
17     app = config.make_wsgi_app()
18     serve(app, host='0.0.0.0')
```

A more extensible app

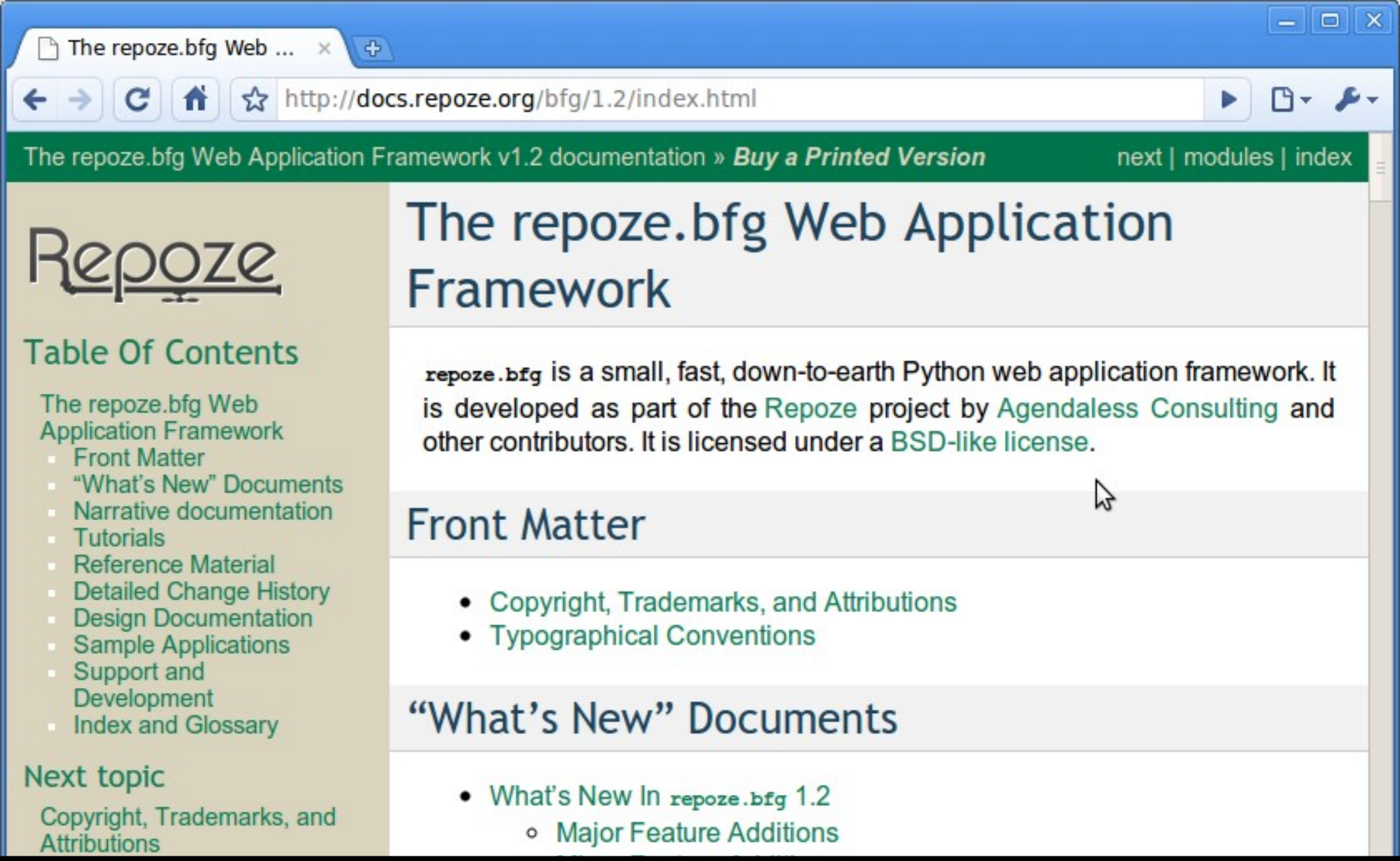
```
from webob import Response
from paste.httpserver import serve
from repoze.bfg.configuration import Configurator

def hello_world(request):
    return Response('Hello world!')

if __name__ == '__main__':
    config = Configurator()
    config.begin()
    config.load_zcml('configure.zcml')
    config.end()
    app = config.make_wsgi_app()
    serve(app, host='0.0.0.0')
```

```
<configure xmlns="http://namespaces.repoze.org/bfg">

  <include package="repoze.bfg.includes" />

  <view
     view="helloworld.hello_world"
     />

</configure>
```

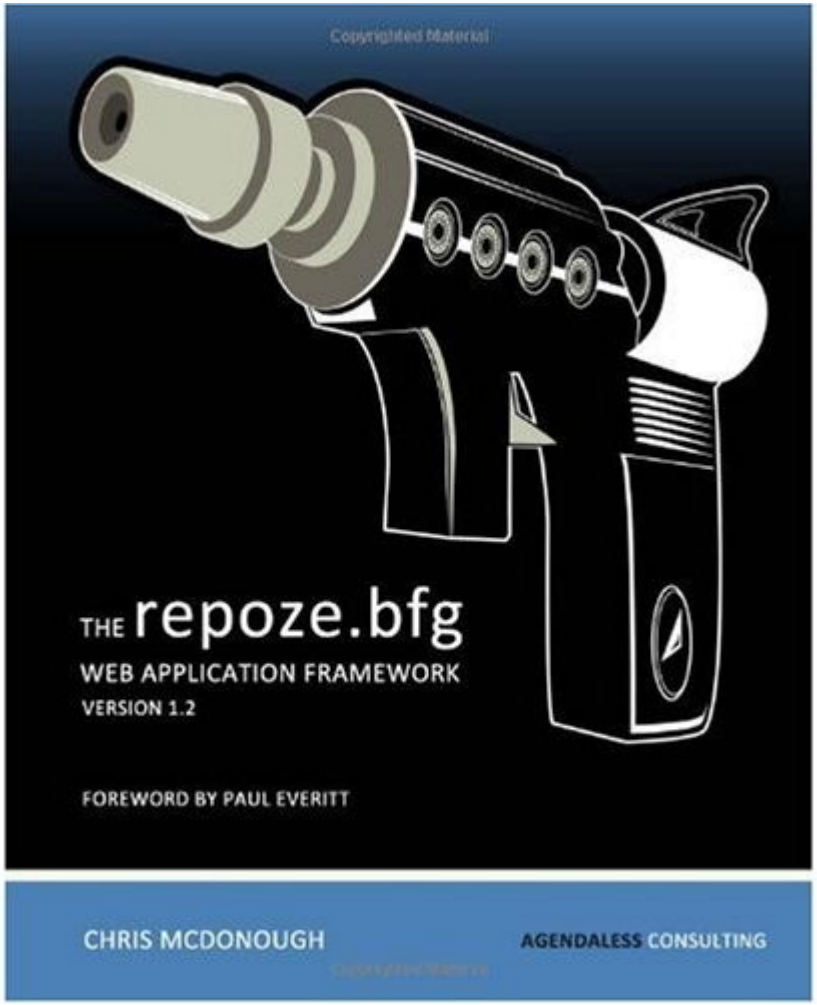
Or:

```
from repoze.bfg.view import bfg_view
from webob import Response

@bfg_view(name='hello', request_method='GET')
def hello(request):
    return Response('Hello')
```

100% unit tested

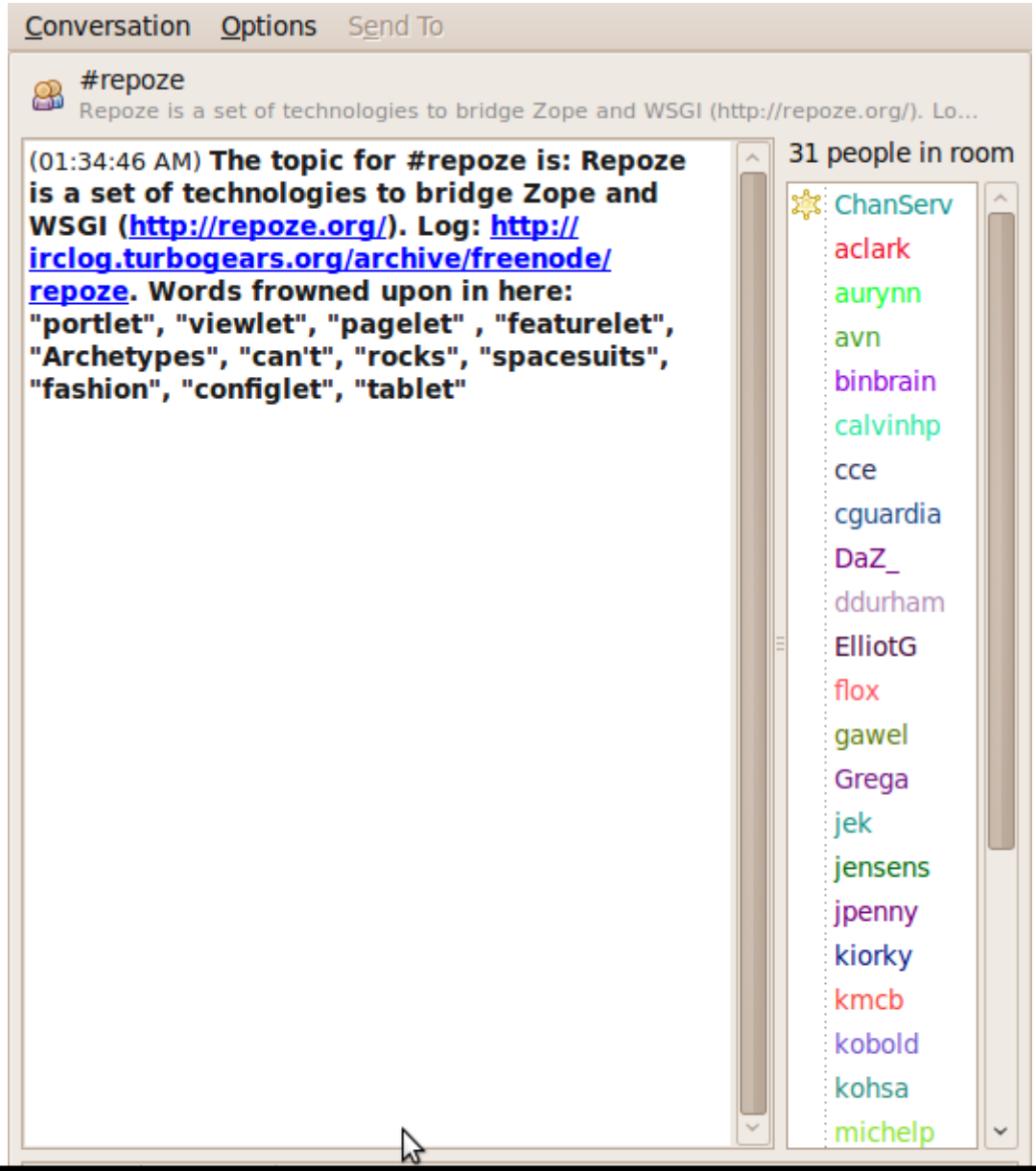


if it's not documented, it's a bug



Available right now, at the exhibit hall!

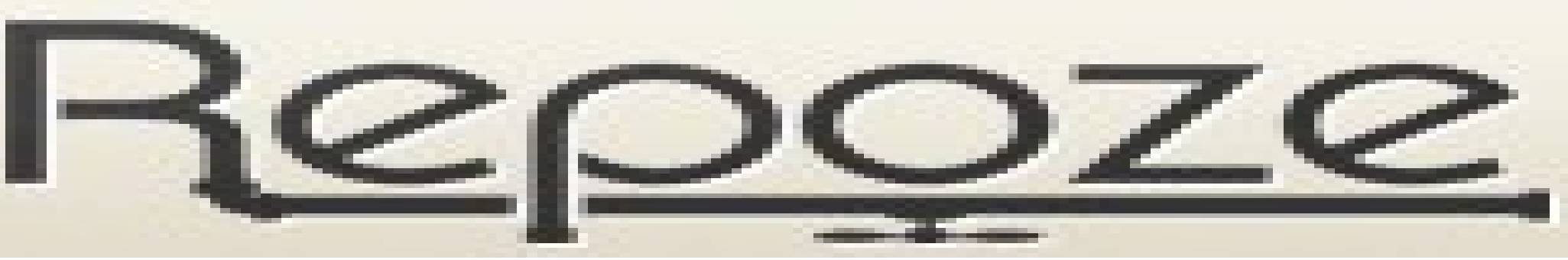
Documentation: The BFG book


Conversation
Options
Send To
#repoze
Repoze is a set of technologies to bridge Zope and WSGI (http://repoze.org/). Lo...
(01:34:46 AM) The topic for #repoze is: Repoze is a set of technologies to bridge Zope and WSGI (http://repoze.org/). Log: http://irclog.turbogears.org/archive/freenode/repoze. Words frowned upon in here: "portlet", "viewlet", "pagelet" , "featurelet", "Archetypes", "can't", "rocks", "spacesuits", "fashion", "configlet", "tablet"
31 people in room
ChanServ
aclark
aurynn
avn
binbrain
calvinhp
cce
cguardia
DaZ_
ddurham
ElliotG
flox
gawel
Grega
jek
jensens
jpenny
kiorky
kmcb
kobold
kohsa
michelp

Community

State of BFG

- Current version is 1.2, released just before PyCon.
- The BFG book covers this recent version, so get it while it's hot.
- In active development, but don't we all say that?
- We're working with the Pylons community to see if a common base can be used across both frameworks, keeping their personality intact (codename: Marco).



- repoze.bfg is part of a software collection under the repoze “brand”.
- The namespace implies a notional community, not software that must be used together.
- Packages in the repository can be used in isolation.

svn.repoze.org: what's in the repo

- **Chameleon**: a fast an extensible template engine. Supports ZPT and Genshi.
- **repoze.who**: identification and authentication framework for arbitrary WSGI applications.
- **repoze.profile**: aggregates Python profiling data across all requests to a WSGI application.
- **repoze.zodbconn**: manages ZODB databases and WSGI middleware which makes a ZODB connection available to downstream applications.

svn.repoze.org: what's in the repo

- **repoze.tm2**: WSGI middleware that implements a transaction policy.
- **repoze.bitblt**: a WSGI middleware component which automatically scales images.
- **repoze.workflow**: a content workflow system.
- **repoze.what**: an authorization framework for WSGI applications.
- **repoze.filesafe**: safely handle creation of files on the filesystem by integrating with the repoze.tm2 transaction manager.

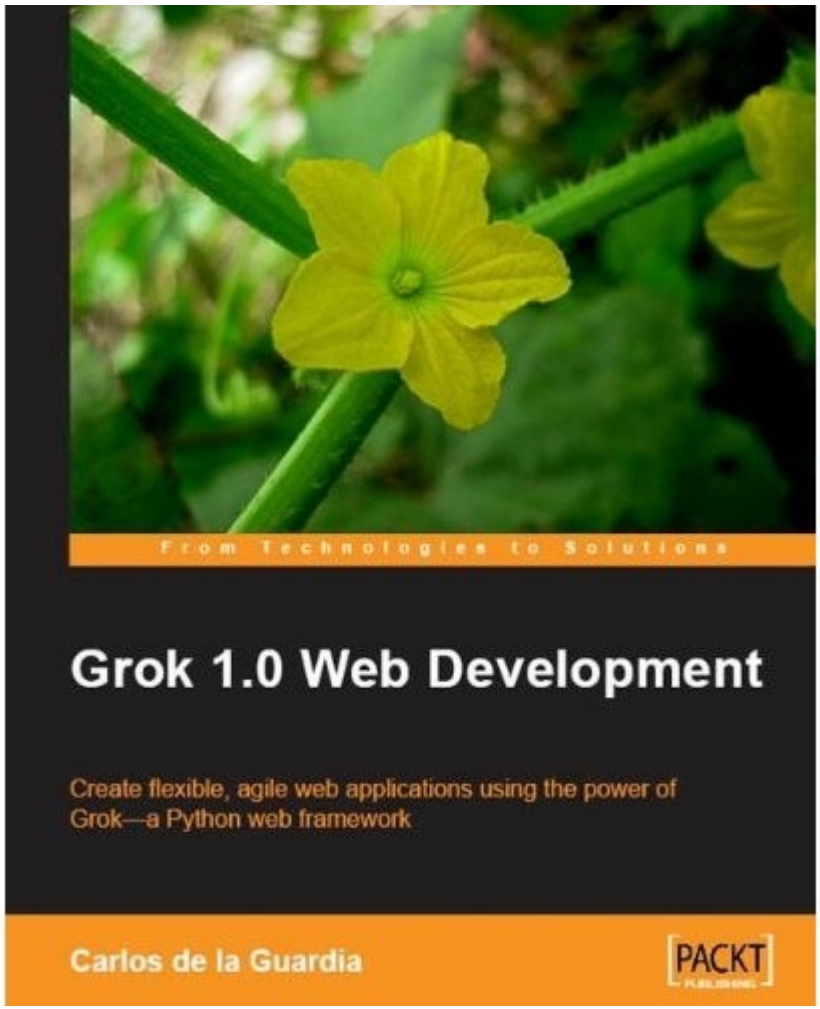
svn.repoze.org: what's in the repo

- **repoze.catalog**: An indexing and searching system based on zope.index.
- **repoze.session**: A sessioning system (server side state for web applications) based on ZODB.
- **repoze.monty**: a library that, given a WSGI environment dictionary, will return a dictionary back containing "converted" form/query string elements.
- **Venusian**: Venusian is a library which allows you to defer the action of decorators from import time to a separate scan phase.
- **repoze.urispace**: implements the URISpace 1.0 spec, as proposed to the W3C by Akamai.

Participate!

- Join us for an open space session right after the next break (4 PM).
- Sprint with us at the conference sprints starting Monday.
- Go to the IRC channel or mailing list and ask away.

And now for something completely different



WorldPlone Day

Thank you.

Follow me on twitter @cguardia (cguardia@yahoo.com)