CS 683 Emerging Technologies
Fall Semester, 2005
Doc 9 Adapter & Tests
Contents

Adapter..... 3
Size..... 4
File Representation..... 8

References

Python Source Code

Zope Source Code

Zope On-line Docs

Web Component Development with Zope 3, von Weitershausen

Adapter

Situation

Class A

- Implements interface B
- Does all/most of what you need

But need a interface C

Solution

Write a new class that implement interface C and uses class A

Zope Examples

Size

Files

Size

When listing files Zope displays the size

Garden.html is book, but size is not displayed

	Name	Title	Size	Created	Modified
☐	Formatting.html		31 lines	9/27/05 9:47 PM	9/27/05 9:47 PM
☐	Garden.html		n/a		
☐	Peanut.html		n/a		
☐	bookstore.css		1 KB	9/27/05 3:37 AM	9/27/05 3:37 AM
☐	macro.html		10 lines	9/26/05 4:41 PM	9/26/05 4:47 PM
☐	page.html		9 lines	9/26/05 4:45 PM	9/26/05 4:57 PM

We need to provide a class that implements ISize for Books

ISized

sizeForDisplay()

Returns a string giving the size

sizeForSorting()

Returns a tuple (basic_unit, amount)

Used for sorting among different kinds of sized objects. 'amount' need only be sortable among things that share the same basic unit.

BookSize

```
from zope.interface import implements
from zope.app.size.interfaces import ISized

class BookSize(object):
    implements(ISized)

    def __init__(self, context):
        self.context = context

    def sizeForSorting(self):
        chars = 0
        chars += len(self.context.title)
        chars += len(self.context.author)

        chars += len(str(self.context.date))
        return ('byte', chars)

    def sizeForDisplay(self):
        unit, chars = self.sizeForSorting()
        return str(chars) + ' characters'
```

Configuring Zope to use BookSize
bookstore/configure.zcml

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

<interface
   interface=".interfaces.IBook"
   type="zope.app.content.interfaces.IContentType"/>

<content class=".book.Book">
   <factory
      id="bookstore.book.Book"
      title="Create a new book"
      description="This factory instantiates new books"/>
   <require
      permission="zope.View"
      interface=".interfaces.IBook" />
   <require
      permission="zope.ManageContent"
      set_schema=".interfaces.IBook"/>
</content>

<adapter
     for=".interfaces.IBook"
     provides="zope.app.size.interfaces.ISized"
     factory=".size.BookSize"
     />

<include package=".browser" />

</configure>
```

New File Listing

	Name	Title	Size	Created	Modified
☐	Formatting.html		31 lines	9/27/05 9:47 PM	9/27/05 9:47 PM
☐	Garden.html		34 characters		
☐	Peanut.html		n/a		
☐	bookstore.css		1 KB	9/27/05 3:37 AM	9/27/05 3:37 AM
☐	macro.html		10 lines	9/26/05 4:41 PM	9/26/05 4:47 PM
☐	page.html		9 lines	9/26/05 4:45 PM	9/26/05 4:57 PM

File Representation

Zope can treat objects as files for

- HTTP
- FTP
- WebDAV
- XML-RPC

Zope uses adapters to make object look like files

Relevant interfaces in zope.app.filerrepesentation

- IReadFile
- IWriteFile
- IReadDirectory
- IWriteDirectory
- IFileFactory
- IDirectFactory

Making a Book accessible via FTP

Zope supports FTP

Any object with an IReadFile is accessible via FTP

Converting a Book to a String

```
from persistent import Persistent
from zope.interface import implements
from zope.schema.fieldproperty import FieldProperty
from bookstore.interfaces import IBook
from datetime import date

class Book(Persistent):
    implements(IBook)

    title = FieldProperty(IBook['title'])
    author =  FieldProperty(IBook['author'])
    date = FieldProperty(IBook['date'])

    def asString(self):
        return self.title + ';' + self.author + ';' + str(self.date)

    def __fromString(main, string):
        book = Book()
        attributes = string.split(';')
        book.title = unicode(attributes[0])
        book.author = unicode(attributes[1])
        dateStringParts = attributes[2].split('-')
        dateParts = map(int, dateStringParts)
        book.date = date(*dateParts)
        return book

    fromString = classmethod(__fromString)
```

Issues

- classmethod

  __fromString(main, string)

  main is value of __name__

- Argument unpacking

  date(*dateParts)

  date(year, month, day)

Tests

```
#! /usr/bin/env python

import sys
sys.path.append('/usr/local/Zope-3.1.0c3/lib/python/')
sys.path.append('/Users/whitney/Courses/683/Fall05/pythonCode/
BetaZope/lib/python')

import unittest
from zope.app.tests.placelesssetup import PlacelessSetup

from bookstore.book import Book
from datetime import date

class TestBook(PlacelessSetup, unittest.TestCase):

    def testAsString(self):
        book = Book()
        book.title = u'a'
        book.author = u'b'
        book.date = date(2005, 5, 6)
        self.assertEqual(book.asString(),u'a;b;2005-05-06')

    def testFromString(self):
        book = Book.fromString(u'a;b;2005-05-06')
        self.assertEqual(book.asString(),u'a;b;2005-05-06')

def suite():
    suite = unittest.TestSuite()
    suite.addTest(unittest.makeSuite(TestBook))
    return suite

if __name__ == '__main__':
    unittest.main()
```

BookReadFile

```
from zope.interface import implements
from zope.app.filerepresentation.interfaces import IRead-
File
from bookstore.book import Book


class BookReadFile(object):
    implements(IReadFile)

    def __init__(self, context):
        self.context = context
        self.data = context.asString().encode('utf-8')

    def read(self):
        return self.data

    def size(self):
        return len(self.data)
```

bookstore/configure.zcml

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

<interface
    interface=".interfaces.IBook"
    type="zope.app.content.interfaces.IContentType"
/>

<content class=".book.Book">
    ...
</content>

<adapter
      for=".interfaces.IBook"
      provides="zope.app.size.interfaces.ISized"
      factory=".size.BookSize"
      />

  <adapter
      for=".interfaces.IBook"
      provides="zope.app.filerepresentation.interfaces.IReadFile"
      factory=".filerepresentation.BookReadFile"
      permission="zope.View"
      />

<include package=".browser" />

</configure>
```

FTP Demo

Size Tests

```
#! /usr/bin/env python

import sys
sys.path.append('/usr/local/Zope-3.1.0c3/lib/python/')
sys.path.append('/Users/whitney/Courses/683/Fall05/pythonCode/
BetaZope/lib/python')

import unittest
from zope.app.tests.placelesssetup import PlacelessSetup

from bookstore.book import Book
from bookstore.size import BookSize

class TestSize(PlacelessSetup, unittest.TestCase):

    def testSizeForSorting(self):
        book = Book.fromString(u'a;b;2005-05-06')
        bookSize = BookSize(book)
        unit , size = bookSize.sizeForSorting()
        self.assertEqual(unit, 'byte')
        self.assertEqual(size, 12)

def suite():
    return   unittest.makeSuite(TestSize)


if __name__ == '__main__':
    unittest.main()
```

All Tests

```
#! /usr/bin/env python

import sys
sys.path.append('/usr/local/Zope-3.1.0c3/lib/python/')
sys.path.append('/Users/whitney/Courses/683/Fall05/pytho
nCode/BetaZope/lib/python')

import unittest

import bookstore.bookTests
import bookstore.sizeTests

if __name__ == '__main__':
    suite = unittest.TestSuite()
    suite.addTest(bookstore.bookTests.suite())
    suite.addTest(bookstore.sizeTests.suite())
    unittest.TextTestRunner().run(suite)
```

But what about testing IFileInterface?