

CS 683 Emerging Technologies

Fall Semester, 2005

Doc 10 Views & Metadata

Contents

Views.....	3
Simple.....	3
Non-Html View.....	6
Metadata.....	13
Turning on Dublin Core for a Class.....	14
Custom Metadata.....	18

Copyright ©, All rights reserved. 2005 SDSU & Roger Whitney, 5500 Campanile Drive, San Diego, CA 92182-7700 USA. OpenContent (<http://www.opencontent.org/opl.shtml>) license defines the copyright on this document.

References

Web Component Development with Zope 3, von Weiter-
shausen

Views Simple

Adding ** as a simplest thing to demo feature

The Secret Garden

By ****Someone****

Printed on 2005-12-12

Text uses separate class rather than modifying current classes

Claims this is Python way - but is it good?

browser/simpleBookView.py

```
class SimpleBookView(object):  
  
    def renderAuthor(self):  
        return '**' + self.context.author + '**'
```

browser/bookview.pt

```
<html xmlns="http://www.w3.org/1999/xhtml"
xmlns:tal="http://xml.zope.org/namespaces/tal"
xmlns:i18n="http://xml.zope.org/namespaces/i18n">
<head>
  <title tal:content="context/title/title">Title</title>
</head>
<body>
<h2 tal:content="context/title/title">recipe name goes
here</h2>
<p>
  By <span tal:replace="structure view/renderAuthor">
</span>
</p>
<p>
  Printed on
  <span tal:replace="context/date"></span>

</p>
</body>
</html>
```

browser/configure.zcml

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

  <browser:page
    for="bookstore.interfaces.IBook"
    name="index.html"
    template="bookview.pt"
    class=".simpleBookView.SimpleBookView"
    permission="zope.View"
  />

  ...
  <browser:resource name="bookstore.css"
    file="bookstore.css" />
</configure>
```

Non-HTML View

Create a text view that is downloadable

The Secret Garden

By ****Someone****

Printed on 2005-12-12

Download as:

- [Text File](#)

Text File link downloads the file

bookstore/text/interfaces.py

```
from zope.interface import Interface
from zope.schema import ASCII

class ITextPresentation(Interface):
    data = ASCII(
        title = u'Data',
        description = u'String representation',
        required = True)
```

bookstore/text/bookText.py

```
from bookstore.text.interfaces import ITextPresentation
from zope.interface import implements

class BookText:
    implements(ITextPresentation)

    def __init__(self, context, request):
        self.data = u'Title:' + context.title + \
            '\rAuthor:' + context.author + \
            '\rDate:' + str(context.date)
```

bookstore/text/browser.py

```
from zope.app import zapi
from zope.app.publisher.browser import BrowserView
from bookstore.text.interfaces import ITextPresentation

class TextView(BrowserView):

    def __call__(self):
        text = zapi.getViewProviding(self.context, ITextPresentation,
                                     self.request)
        filename = zapi.name(self.context) + '.txt'
        response = self.request.response
        response.setHeader('Content-Disposition',
                           'attachment; filename=%s' % filename)
        response.setHeader('Content-Type', 'application/txt')
        response.setHeader('Content-Length', len(text.data))
        response.write(text.data)
```


bookstore/text/configure.zcml

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

  <view
    for="bookstore.interfaces.IBook"
    provides=".interfaces.ITextPresentation"
    type="zope.publisher.interfaces.IRequest"
    factory=".bookText.BookText"
  />

  <browser:page
    for="bookstore.interfaces.IBook"
    name="txt"
    class=".browser.TextView"
    permission="zope.View"
    menu="zmi_views" title="Text File"
  />

</configure>
```

bookstore/configure.zcml

```
<configure xmlns="http://namespaces.zope.org/zope">  
  
  <interface  
    interface=".interfaces.IBook"  
    type="zope.app.content.interfaces.IContentType"  
  />  
  
  ...  
  
  <include package=".browser" />  
  <include package=".text" />  
  
</configure>
```

At this point the view exists

No way for end users to interact with the view

Adding a Menu

bookstore/browser/configure.zcml

```
<configure
  xmlns="http://namespaces.zope.org/zope"
  xmlns:browser="http://namespaces.zope.org/browser"
  >
  <browser:menu
    id="alternate_views"
    title="Menu showing different ways to view an object"
  />
```

bookstore/browser/simpleBookView.py

```
from zope.app import zapi, servicenames

class SimpleBookView(object):

    def renderAuthor(self):
        return '**' + self.context.author + '**'

    def alternateViews(self):
        #Works in Zope3.0 but not Zope3.1
        menu_service = \
            zapi.getService(servicenames.BrowserMenu)
        menu_id = 'alternate_views'
        return menu_service.getMenu(menu_id, self.context,
self.request)
```

bookstore/browser/bookview.py

```

<html xmlns="http://www.w3.org/1999/xhtml"
xmlns:tal="http://xml.zope.org/namespaces/tal" >
<head>
  <title tal:content="context/title/title">Title</title>
</head>
<body>
<h2 tal:content="context/title/title">recipe name goes here</h2>
<p>
  By <span tal:replace="structure view/renderAuthor"> </span>
</p>
<p>
  Printed on
  <span tal:replace="context/date"></span>

</p>
<h4>Download as:</h4>

<ul>
  <li tal:repeat="item view/alternateViews">
    <a href=""
      tal:attributes="href
        string:${context/@@absolute_url}/${item/action}"
      tal:content="item/title">alternate view</a>
  </li>
</ul>

</body>
</html>

```

Metadata

Data that describes and gives information about other data

Dublin Core Metadata

<http://dublincore.org/>

Defines some standard metadata

Supported by Zope

Some Dublin Metadata

Contributor	Format	Rights
Coverage	Identifier	Source
Creator	Language	Subject
Date	Publisher	Title
Description	Relation	Type

Turning on Dublin Core for a Class bookstore/configure.zcml

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

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

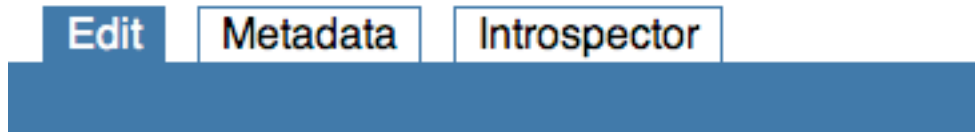
<content class=".book.Book">
  <factory
    id="bookstore.book.Book"
    title="Create a new book"
    description="This factory instantiates new books"
  />
  <implements
    interface="zope.app.annotation.interfaces.IAttributeAnnotatable"
  />
  <require
    permission="zope.View"
    interface=".interfaces.IBook"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

</content>
```

When we create books, the metadata is stored in the class

Seeing the Metadata

When editing a book there is a Metadata tag



Edit

Title

author

Date

When listing a book, Created & Modified fields are filled

	Name	Title	Size	Created	Modified
<input type="checkbox"/>	Omni		48 KB	9/19/05 4:40 PM	9/19/05 4:40 PM
<input type="checkbox"/>	SecretGarden		34 characters	10/4/05 4:42 PM	10/4/05 4:42 PM

Accessing the Metadata

We can access & change metadata

Here we access metadata via TAL

bookstore/browser/bookview.pt

```
<html xmlns="http://www.w3.org/1999/xhtml"
xmlns:tal="http://xml.zope.org/namespaces/tal" >
<head>
  <title tal:content="context/title/title">Title</title>
</head>
<body>
<h2 tal:content="context/title/title">recipe name goes here</h2>
<p>
  By <span tal:replace="structure view/renderAuthor"> </span>
</p>
<p>
  Printed on
  <span tal:replace="context/date"></span>

</p>
<p tal:define="created context/zope:created;
  modified context/zope:modified">
  This book entered on: <span tal:replace="python: created"/>.<br/>
  It was last modified on: <span tal:replace="python: modified"/>.
</p>

</body>
</html>
```


The new view of a Book

The Secret Garden

By ****Someone****

Printed on 2005-12-12

This book entered on: 2005-10-04 16:42:40.508171+00:00.

It was last modified on: 2005-10-04 16:42:40.508171+00:00.

Custom Metadata A Ranking System

The Secret Garden

By ****Someone****

Printed on 2005-12-12

Ratings

This book has received an average rating of **4.0** (4 votes).

Rate this book: 1 2 3 4 5

Added to bookstore/interfaces.py

```
from zope.schema import Float
from zope.schema import Int
from zope.app.annotation.interfaces import IAnnotatable

class IRatable(IAnnotatable):
    """Marker interface that promises that an implementing object
       maybe rated using ``IRating`` annotations.
    """

class IRating(Interface):
    """Give and query rating about objects, such as recipes.
    """

    def rate(rating):
        """Rate the current object with `rating`.
        """

    averageRating = Float(
        title=u"Average rating",
        description=u"The average rating of the current object",
        required=True
    )

    numberOfRatings = Int(
        title=u"Number of ratings",
        description=u"The times the current has been rated",
        required=True
    )
```

bookstore/rating.py

```
from persistent.dict import PersistentDict
from persistent.list import PersistentList
from zope.interface import implements
from zope.app.annotation.interfaces import IAnnotations
from bookstore.interfaces import IRating

KEY = "bookstore.rating"

class Rating(object):
    implements(IRating)

    def __init__(self, context):
        self.context = context
        annotations = IAnnotations(context)
        mapping = annotations.get(KEY)
        if mapping is None:
            blank = {'average': 0.0, 'ratings': PersistentList()}
            mapping = annotations[KEY] = PersistentDict(blank)
        self.mapping = mapping

    def rate(self, rating):
        ratings = self.mapping['ratings']
        ratings.append(float(rating))
        self.mapping['average'] = sum(ratings)/len(ratings)

    def averageRating(self):
        return self.mapping['average']
    averageRating = property(averageRating)

    def numberOfRatings(self):
        return len(self.mapping['ratings'])
    numberOfRatings = property(numberOfRatings)
```

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"
  />
  <implements interface=
    "zope.app.annotation.interfaces.IAttributeAnnotatable
    .interfaces.IRatable"
  />
  <require
    permission="zope.View"
    interface=".interfaces.IBook"
  />
  <require
    permission="zope.ManageContent"
    set_schema=".interfaces.IBook"
  />
</content>
```

bookstore/configure.zcml continued

```
<adapter
  for=".interfaces.IRatable"
  provides=".interfaces.IRating"
  factory=".rating.Rating"
  trusted="true"
/>

<content class=".rating.Rating">
  <require
    permission="zope.View"
    attributes="averageRating numberOfRatings rate"
  />
</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" />
<include package=".text" />

</configure>
```

bookstore/browser/bookview.pt

```

<html xmlns="http://www.w3.org/1999/xhtml"
xmlns:tal="http://xml.zope.org/namespaces/tal" >
<head><title tal:content="context/title/title">Title</title></head>
<body>
<h2 tal:content="context/title/title"></h2>
<p>
By <span tal:replace="structure view/renderAuthor"/></p>
<p>
Printed on<span tal:replace="context/date"></span> </p>

<h4>Ratings</h4>
<p tal:define=
    "rating view/rating;
    average rating/averageRating;
    votes rating/numberOfRatings;
    formatter python:request.locale.numbers.getFormatter('decimal')">
This book has received an average rating of
    <strong tal:content=
        "python:formatter.format(average, '###0.0')">0.0</strong>
    (<strong tal:content="votes">12</strong> votes).</p>

<form action="@@rate.html" method="post">
    <p><span>Rate this book:</span>
    <tal:loop tal:repeat="rating view/ratingChoices">
        <input type="radio" name="rating:float"
            tal:attributes="value rating;
                id string:rating-${repeat/rating/number}"/>
        <label tal:attributes="for string:rating-${repeat/rating/number}"
            tal:content="rating">rating</label>&nbsp;
    </tal:loop>
    <input type="submit" value="Rate"/>
    </p>
</form>
</body>
</html>

```

bookstore/browser/simpleBookView.py

```
from zope.app import zapi, servicenames
from bookstore.interfaces import IRating

class SimpleBookView(object):

    def renderAuthor(self):
        return ''' + self.context.author + '''

    def alternateViews(self):
        print servicenames
        menu_service = zapi.getService('BrowserMenu')
        menu_id = 'alternate_views'
        return menu_service.getMenu(menu_id, self.context,
self.request)

    def rating(self):
        return IRating(self.context)

    def rate(self, rating):
        IRating(self.context).rate(rating)
        self.request.response.redirect('.')

ratingChoices = (1, 2, 3, 4, 5)
```


Added to bookstore/browser/configure.zcml

```
<browser:page
  for="bookstore.interfaces.IBook"
  name="rate.html"
  class=".simpleBookView.SimpleBookView"
  attribute="rate"
  permission="zope.View"
/>
```

Current Files in bookstore

```
__init__.py
book.py
browser/
  __init__.py
  bookstore.css
  bookview.pt
  configure.zcml
  simpleBookView.py
configure.zcml
filerepresentation.py
interfaces.py
rating.py
size.py
text/
  __init__.py
  bookText.py
  browser.py
  configure.zcml
  interfaces.py
```