

XML on the Web

INFO 344 Winter 2007

On the Web

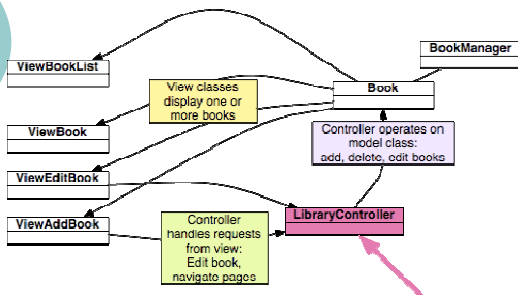
- `$_ENV['REDIRECT_REMOTE_USER'];`
- Using bridge tables in databases

Today

- The MVC Controller Revisited
- XML
- XML in PHP
- RSS: Site updates via XML

MVC: The Controller

The MVC pattern



The Controller

- Hopefully obvious by now what models and views do
- Without a Controller pages don't *do anything*
 - Don't respond to user input
 - Don't interact with the database
- A single Controller page can tie everything together

Controller basics

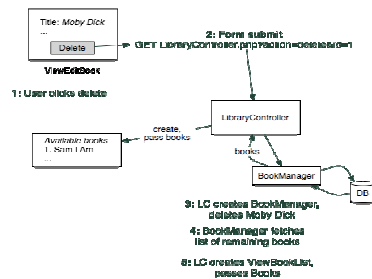
- The Controller provides a set of *actions* that the application can perform
 - Add a new book to the database
 - Delete a book
 - Show the book search page for "F. Scott Fitzgerald"
- The Controller manipulates the other classes to perform these actions
 - Calls BookManager to create and fetch Book objects
 - Passes Books to the Viewer class to display them

Controller basics, continued

- The Controller file takes in parameters from the HTTP request, using `$_REQUEST`
 - Can be passed values through links or forms
- Every Controller action contains the following steps
 - Retrieve action parameter
 - Choose appropriate action
 - Fetch additional parameters as needed

Controller sequence of events

Scenario: User is viewing Moby Dick's edit page



Linking views to controllers

- The Controller class is a PHP page that receives instructions from forms or URL string
- View pages interact with the Controller through links or forms
- Links to controller should pass necessary info
 - Action (always!)
 - Any additional parameters (book ID, user/pass, etc)

Index page in MVC

- Our Controller class is now responsible for all program actions
 - We created an "index" action for the front page
 - Now, our index.php file should just forward to the Controller

```
<?php
// index.php
// redirect to LibraryController
header("Location:
LibraryController.php?action=index");
?>
```

Questions about MVC?

XML and RSS

What is XML?

- XML is a language for specifying structured documents
 - Elements <tag>, attributes
- XML is a *language to describe languages*
 - By itself, doesn't describe much
 - Can be used to describe many types of data
 - Animals, recipes, geographic data, graphics...

Why XML?

- XML is a standardized format
 - Can be read and written by most languages
- XML is human-readable (if written well)
- XML is flexible
 - Can be altered with style sheets
 - Transformations between different types of XML
 - e.g. HTML table to chart
- Supports records, lists, and trees
- Plain text; platform independent

Why XML ... Not

- Verbose and redundant
- Can be difficult to read
- Inefficient to parse, store, and transmit
- Parsers must deal with arbitrary levels of nesting and errors
- No concept of data types
- Hierarchical not relational

XML and Databases

- Before XML – ODBC
 - ODBC granted access to your database
 - Required significant client configuration
- XML
 - Not a replacement for a DBMS!
 - A standardized format to “stream” data from your DBMS so that it can be consumed/read by others
- Easy to go from DB to XML
- Hard to go from XML to DB

Sample XML

```
<?xml version="1.0" encoding="UTF-8"?>
<animal type="cat">
  <name>Murray</name>
  <weight unit="lbs">16.0</weight>
  <colors>
    <eyes>green</eyes>
    <coat>orange</coat>
  </colors>
</animal>
```

Well-Formed XML

- One and only one root element
 - `<animals> ... </animals>`
- Non-empty elements have start and end tags
 - `<animal name="Murray"> ... </animal>`
- Empty elements are self closing
 - `<IEMpty />`
 - `
`
- All attribute values are quoted
- Tags may be nested but not overlap
- Elements names are case-sensitive

XML Stylesheets

```
<?xml version='1.0'?>
<?xml-stylesheet type="text/xsl" href="events5.xsl"?>
<events>
  <play Ptype = "drama">
    <title>Three Sisters </title>
    <author>
      <First_Name>Anton</First_Name>
      <Last_Name>Chekhov</Last_Name>
    </author>
  </play>
  <play Ptype = "comedy">
    <title>As You Like It</title>
    <author>
      <First_Name>William</First_Name>
      <Last_Name>Shakespeare</Last_Name>
    </author>
  </play>
</events>
```

XML Stylesheets

```
<?xml version='1.0'?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl">
  <xsl:template match="/">
    <html>
      <body>
        <table cellspacing="0" width="100%">
          <tr>
            <td align="center" bgcolor="#ccccff">
              <b>Drama At The UW</b>
            </td>
          </tr>
        </table>
        <xsl:for-each select="events/play">
          <p>
            <b><xsl:value-of select="title"/></b>
            by <xsl:value-of select="author/First_Name"/>
            <xsl:value-of select="author/Last_Name"/>
          </p>
        </xsl:for-each>
      </body>
    </html>
  </xsl:template>
</xsl:stylesheet>
```

Handling XML in PHP

- PHP provides facilities for working with XML
- Advanced features in PHP 5
 - SimpleXML library for reading XML
- PHP can output XML just as easily as HTML
 - Just output XML tags rather than HTML

SimpleXML

- PHP 5 library for reading XML
 - Converts an XML file or string to PHP object
 - Follows tree hierarchy of XML
- ```
// create SXML object from XML string ($xmlstr)
$cat = new SimpleXMLElement($xmlstr);
// output parameters
echo $cat->colors->eyes; // "blue"
// change parameters
$cat->weight = 13; // weight is now 13
// output new XML document as XML string
$fatCat = $cat->asXML();
```

see <http://us3.php.net/simplexml>

## Outputting XML

- No great library for turning objects to XML
- We can simply use *echo* to write XML text

```
echo
 "<animal><name>$name</name></
 animal>";
```
- Alternately, we can use the idea of Views to create XML documents

## Book view

---

```
<?php
// display a book object as XML
function bookAsXML($book) {
?>
 <book>
 <title><?php echo $book->title; ?></title>
 <author><?php echo $book->author;
?></author>
 <pages><?php echo $book->pages; ?></pages>
 <isbn><?php echo $book->isbn; ?></isbn>
 </book>
<?php
}
?>
```

## Book view output

---

```
<book>
 <title>Sam I Am</title>
 <author>Dr. Seuss</author>
 <pages>44</pages>
 <isbn>12345</isbn>
</book>
```

## XML on the Web: So what?

---

## Who cares about XML?

---

- XML is just another storage format, big whoop...
- XML data structures organize a set of information so that it can centrally maintained, yet delivered in many different formats.
- Useful where organizations are "loosely coupled" yet need to reliably exchange information.
- XML systems can be used to collect and redistribute information and to make available information that otherwise may not be found. (syndication)

## Who cares about XML?

---

- A number of fundamental Web technologies are XML-based
  - Knowing XML allows us to take advantage of them
- XHTML: Web document format (XML)
- SVG: Shape-based graphics (XML)
- RSS: News and updates (XML)
- SOAP/REST/XML-RPC Web Services (XML)

## SVG: Scalable Vector Graphics

---

- A *vector-based* graphics format in XML
  - Graphics consist of lines, shapes, colors (not pixels)
- Can be viewed by most modern browsers
- XML format, like HTML
  - Can use JavaScript for animation / interactivity
  - CSS for style

## SVG example

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd"
">
<svg width="100%" height="100%" version="1.1"
xmlns="http://www.w3.org/2000/svg">
<circle cx="100" cy="50" r="40" stroke="black"
stroke-width="2" fill="red"/>
</svg>
```



## RSS: Really Simple Syndication

- Problem: manually checking web pages to see whether they have updated can be tedious
  - e.g. Some people read hundreds of blogs
- RSS is an XML-based format used to describe updates that occur on a web site over time
  - Blog posts, news stories, software updates, homework assignments
- RSS allows users to [subscribe to an RSS feed](#) and be notified of updates to the source site

## How RSS works

- An RSS file contains a list of event items
  - Each item has a title, description, and publishing time
- RSS reader software can be used to subscribe to an RSS file URL (or "feed")
  - Checks file at regular intervals
  - If new update appears, display it to user
- RSS update items usually link to HTML documents
  - RSS has headline, links to HTML news story

## RSS readers

- Mac
  - NetNewsWire
  - NewsFire
- Windows
  - Firefox
  - IE7
- Web-based
  - My Yahoo!
  - Google Personalized Homepage
  - Bloglines

## RSS formats

- Several incompatible versions of the RSS standard
  - RSS 1.0 (the original)
  - RSS 2.0 (cleaner code, more features)
  - Atom (similar format managed by different group)
- Most readers support all three formats
- We'll use RSS 2.0; it's the simplest

## Example RSS file

```
<rss version="2.0">
<channel>
<title>INFO 344 Updates</title>
<link>http://faculty.washington.edu/loter/info344</link>
<description>Info 344 Updates</description>

<item>
<title>Course website is up!</title>

<link>http://faculty.washington.edu/loter/info344/</link>
<description>
The course website has been posted. This RSS feed
was written a month later, but don't tell anyone!
</description>
<pubDate>Fri, 16 Jan 2007 08:00:00 -
0800</pubDate>
</item>
</channel>
</rss>
```

## Example RSS file

```
<rss version="2.0">
 <channel>
 <title>INFO 344 Updates</title>
 <link>http://faculty.washington.edu/loter/info344</link>
 <description>Info 344 Updates</description>
 <item>
 <title>Course website is up!</title>
 <link>http://faculty.washington.edu/loter/info344</link>
 <description>
 The course website has been posted. This RSS feed
 was written a month later, but don't tell anyone!
 </description>
 <pubDate>Fri, 16 Jan 2007 08:00:00 -
 0800</pubDate>
 </item>
 </channel>
</rss>
```

**Feed description**

**Feed items**

## RSS header

```
<rss version="2.0">
<channel>
 <title>INFO 344 Updates</title>

 <link>http://faculty.washington.edu/loter/info344</link>
 <description>Info 344
 Updates</description>
```

## RSS items

```
<item>
 <title>Course website is up!</title>
 <link>http://faculty.washington.edu/loter/info344</link>
 <description>
 The course website has been posted. This
 RSS feed was written a month later,
 but don't tell anyone!
 </description>
 <pubDate>Fri, 16 Jan 2007 08:00:00 -
 0800</pubDate>
</item>
...
```

## Dates in RSS

- Dates in RSS require a specific date format  
Fri, 16 Jun 2006 08:00:00 -0800  
day, date month year hh:mm:ss timezone
- In PHP, `date(DATE_RSS)` returns the current date and time in this format
- Can store in database as a string, or convert in PHP

## Creating RSS in PHP

- Output XML as normal
- Draw from database for items
- Link to PHP web site