

CS320 Web and Internet Programming

Some Loose Ends

Chengyu Sun
California State University, Los Angeles

File Upload in Web Applications

- ◆ Homework submission in CSNS
- ◆ File attachments in Gmail
- ◆ User profile photo in web forums
- ◆ Online video, picture, file management applications ...

File Upload – The Form

```
<form action="FileUploadHandler"
      method="post"
      enctype="multipart/form-data">

  First file: <input type="file" name="file1" /> <br />
  Second file: <input type="file" name="file2" /> <br />

  <input type="submit" name="upload" value="Upload" />

</form>
```

File Upload – The Request

```
POST / HTTP/1.1
Host: cs.calstatela.edu:4040
[...]
Cookie: SITESERVER=ID=289f7e73912343a2d7d1e6e44f931195
Content-Type: multipart/form-data; boundary=-----146043902153
Content-Length: 509

-----146043902153
Content-Disposition: form-data; name="file1"; filename="test.txt"
Content-Type: text/plain

this is a test file.

-----146043902153
Content-Disposition: form-data; name="file2"; filename="test2.txt.gz"
Content-Type: application/x-gzip

???UC
```

Apache commons-fileupload

◆ <http://jakarta.apache.org/commons/fileupload/using.html>

```
FileItemFactory fileItemFactory = DiskFileItemFactory();
ServletFileUpload fileUpload = new ServletFileUpload( fileItemFactory );

List<FileItem> items = (List<FileItem>) fileUpload.parseRequest( req );
for( FileItem item : items )
{
  if( ! item.isFormFiled() )
  {
    fileItem.write( file );
  }
}
```

Example: File Upload

- ◆ Upload a file and save it to disk
 - Use absolute path
 - Use relative path
- ◆ Handle regular (i.e. non-file) form fields
- ◆ Save uploaded files to database
 - <http://dev.mysql.com/doc/refman/5.5/en/storage-requirements.html>

Store Uploaded Files

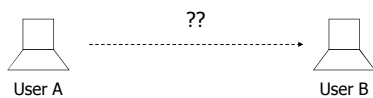
- ◆ On disk
- ◆ In database
 - BLOB, CLOB/TEXT
 - BINARY VARCAR, VARCHAR

◆ *Pros and Cons??*

Email in Web Applications

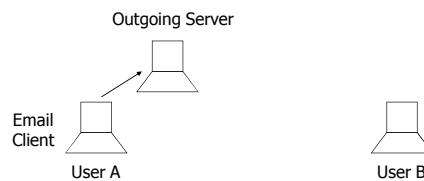
- ◆ Recover account information
- ◆ Various notifications, reminders, news letters ...

How Email Works



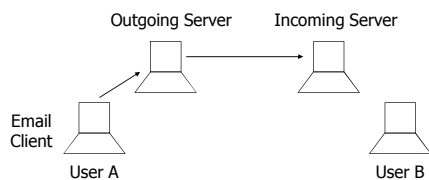
- ◆ Email address
 - user@domain, e.g. csun@calstatela.edu
 - user@host, e.g. csun@exchange.calstatela.edu

Email Client and Outgoing Email Server



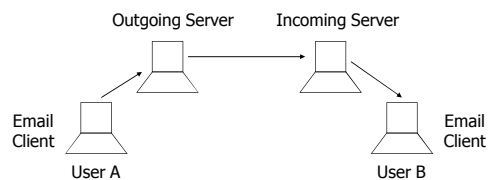
- ◆ Email client
 - E.g. Outlook, Thunderbird, web-based like Gmail
- ◆ Outgoing Server

Incoming Server and DNS



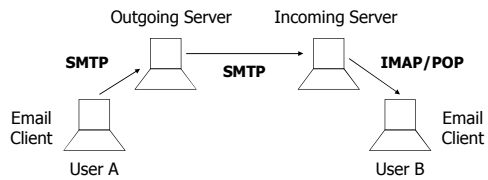
- ◆ Incoming Server
- ◆ Locating the incoming server – DNS lookup

Receiving Email



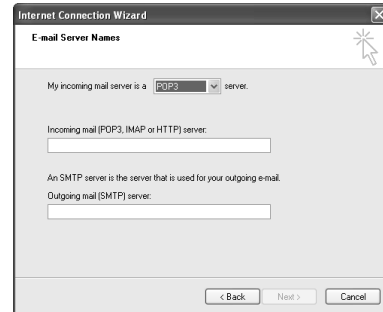
- ◆ An email client retrieves the emails from the incoming email server

Email Protocols



- ◆ Simple Mail Transfer Protocol
- ◆ Internet Message Access Protocol
- ◆ Post Office Protocol

Set up Email Servers in Outlook



JavaMail

◆ <http://www.oracle.com/technetwork/java/javamail/index.html/>

```
Properties props = System.getProperties();
props.put("mail.smtp.host", mailhost);
Session session = Session.getInstance( props );
```

```
Message msg = new MimeMessage(session);
msg.setFrom( new InternetAddress( from ) );
msg.setRecipient( RecipientType.TO, new InternetAddress( to ) );
msg.setSubject( subject );
msg.setText( content );
Transport.send( msg );
```

Example: Email

From:

To:

Subject:

Content:

Application Deployment



WAR Files

- ◆ Web application ARchive
- ◆ A JAR file for packaging, distributing, and deploying Java web applications
- ◆ Create WAR files
 - The command line tool `jar`
 - Eclipse Export -> Web -> WAR file

Deploy WAR Files to a Tomcat Server

- Place the WAR file under a specific directory
 - `webapps` for Tomcat
 - `autodeploy` for GlassFish
- Manager console or web interface