

## CS520 Web Programming

### File Upload and Email

Chengyu Sun  
California State University, Los Angeles

## 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

???
```

## Apache commons-fileupload

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

```
FileItemFactory fileItemFactory = DiskFileItemFactory( 8096, "/tmp" );
ServletFileUpload fileUpload = new ServletFileUpload( fileItemFactory );
```

```
List items = fileUpload.parseRequest( request );
for( Object o : items )
{
  FileItem item = (FileItem) items;
  if( ! item.isFormField() ) {...}
}
```

## Spring File Upload Support

- ◆ `MultipartResolver` bean
  - Support multiple request parser libraries
- ◆ Handle uploaded files
  - Treat a uploaded file as a `String` or `byte[]` field
    - Spring Reference Documentation, Chapter 13.8 - <http://static.springframework.org/spring/docs/2.0.x/reference/mvc.html>
  - Use `MultipartHttpServletRequest` and `MultipartFile`
    - <http://static.springframework.org/spring/docs/2.0.x/api/org.springframework.web.multipart/MultipartHttpServletRequest.html>

## Spring File Upload Example

◆ CSNS:  
`student/UploadFilesController.java`

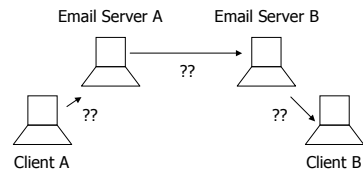
```
if( request instanceof MultipartHttpServletRequest )
{
  Iterator i = r.getFileNames();
  if( i.hasNext() )
  {
    MultipartFile uploadedFile = r.getFile( (String) i.next() );
    if( !uploadedFile.isEmpty() ) {...}
  }
}
```

## Saving the Uploaded Files

- ◆ Database
  - Option A: BINARY VARCHAR or VARCHAR
  - Option B: BLOB or CLOB
- ◆ Option C: disk file
- ◆ *Pros and Cons??*

## How Email Works

- ◆ SMTP, IMAP, POP



## JavaMail

- ◆ <http://java.sun.com/products/javamail/>

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

Message msg = new MimeMessage(session);
...
Transport.send( msg );
```

## Spring Email Support

- ◆ Declare a `mailSender` bean
- ◆ Mail message classes
  - `SimpleMailMessage`
    - ◆ No attachment, no special character encoding
  - `MimeMailMessage`

## Configure Mail Sender

```
<bean id="mailSender"
      class="org.springframework.mail.javamail.JavaMailSenderImpl">
  <property name="host" value="localhost">>
</bean>
```

- ◆ Additional properties
  - port
  - username, password

## SimpleMailMessage Example

- ◆ CSNS: `ResetPasswordController.java`

```
SimpleMailMessage message;
message = new SimpleMailMessage( messageTemplate );
message.setTo( user.getEmail() );
message.setText( "Your password has been reset to ..." );

mailSender.send( message );
```

## Lessons Learned

- ◆ It's simple as long as you understand some basics and find the right tools/libraries.