CS520 Web Programming
File Upload and Email

Chengyu Sun
California State University, Los Angeles

Echo Server

- Display HTTP requests
  - http://volume1.coreservlets.com/archive/Chapter19.html

File Upload 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( 8096, "/tmp" );
ServletFileUpload fileUpload = new ServletFileUpload( fileItemFactory );

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

Spring Fileupload Support

- Support multiple request parsers
  - Declare a `multipartResolver` bean
  - Files are handled as objects of `MultipartFile`

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

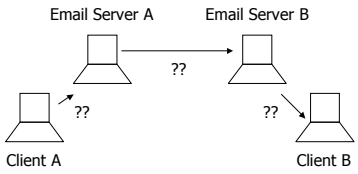
Saving the Uploaded Files

- Database
  - Option A: BINARY VARCAR 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
- Create `SimpleMailMessage`

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

```
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.