

File upload

Programmazione in Ambienti Distribuiti

V 1.0 © Marco Torchiano 2007

File upload

- L'upload di file non può essere eseguito usando un normale form
 - Il metodo GET non è utilizzabile perchè consente di inviare solo una limitata quantità di dati
 - La codifica normale non è in grado di gestire il contenuto di un file
-

Parsing della richiesta

- Sul lato server è necessario fare il *parsing* della HTTP request
 - Si può utilizzare la libreria FileUpload sviluppata dal progetto Apache Commons
 - <http://commons.apache.org/FileUpload>
-

Form HTML

```
<form action="Upload"
  enctype="multipart/form-data"
  method="POST">
```

Permette di codificare il contenuto dei file e gli altri campi del form

Il metodo POST invia il contenuto nel corpo della HTTP request

```
<input type="file" name="file">
<input type="submit" value="Upload">
</form>
```

Campo per la selezione di un file

Procedura server

- Setup del parser
 - Parsing della HTTP request
 - Prelievo del campo contenente il file
 - Scrittura del file nella destinazione finale
-

Setup del parser

```
// create request parser
DiskFileUpload fu = new DiskFileUpload();
// maximum size before a FileUploadException
fu.setSizeMax(1000000); // this is 1 MB
// maximum size that will be stored in memory
fu.setSizeThreshold(4096);
// the location for saving data if size > thresh
fu.setRepositoryPath(
    System.getProperty("java.io.tmpdir"));
```

Setup del parser

```
List fileItems = fu.parseRequest(req);
for(Iterator it=fileItems.iterator();
    it.hasNext();){
    FileItem file = (FileItem)it.next();
    // file sia contenuto nell'attributo 'file'
    if(file.getFieldName().equals("file")){
        // write file in its final location
    }
}
```

Scrittura del file

```
File fOUT = new File(path);
InputStream fIS = file.getInputStream();
OutputStream fOS = new FileOutputStream(fOUT);
while (fIS.available() > 0){
    fOS.write(fIS.read());
}
fIS.close();
fOS.close();
```
