[FileUpload 서블릿 구현]

클라이언트단(브라우저)에서 파일을 선택해서 서버에 전송하는 것을 파일업로드라 한다. 웹 사용자가 전송하려는 파일을 선택할 수 있게 〈input type="file"……〉 태그를 〈form〉 태그 안에 사용한다.

또한 요청 방식은 반드시 POST 여야 하며 enctype="form-data"라는 속성을 추가로 사용해서 전송하려는 파일의 내용이 인코딩되지 않은 상태로 전송되도록 구현해야 한다. 〈form〉 태그에 enctype="form-data"을 설정한 경우에는 입력 폼에 사용자가 입력한 내용이 name=value&name=value... 형식으로 인코딩되어 전달되는 것이 아니고 여러 파티션(multipart)으로 나뉘어서 서버에 전송된다.

〈form〉 태그를 구성하는 각각의 〈input〉 태그별로 그리고 업로드되는 파일별로 하나의 파트를 이루게 되며 각각의 파티션은 "-----boundary ID"로 구분된다. 이렇게 구성하는 Content-Type을 multipart/form-data 라고 한다.

```
Headers Preview Response Timing
  Accept-Language: ko-KR,ko;q=0.9,en-US;q=0.8,en;q=0.7
  Connection: keep-alive
  Content-Length: 552894
  Content-Type: multipart/form-data; boundary=----WebKitFormBoundaryhZUeqZ3djDBXgajL
  Host: localhost:8000
  Origin: http://localhost:8000
  Referer: http://localhost:8000/edu/ajaxexam/ex06_08.html
  User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
  ecko) Chrome/63.0.3239.108 Safari/537.36
▼ Request Payload
   -----WebKitFormBoundaryhZUeqZ3djDBXgajL
  Content-Disposition: form-data; name="theAuthor"
   -----WebKitFormBoundaryhZUeqZ3djDBXgajL
  Content-Disposition: form-data; name="theAge"
   -----WebKitFormBoundaryhZUeqZ3djDBXgajL
  Content-Disposition: form-data; name="theFile"; filename="HoonJunglebook.ttf"
  Content-Type: application/octet-stream
  -----WebKitFormBoundaryhZUeqZ3djDBXgajL--
```

[파일업로드를 처리하는 서블릿 구현]

```
@MultipartConfig (location = "c:/Temp/uploadtest")
```

public class PartTestServlet extends HttpServlet {

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

```
request.setCharacterEncoding("utf-8");

Collection<Part> parts = request.getParts();

for(Part part : parts) {

    String filename = part.getSubmittedFileName();

    if (filename != null) {

        part.write(fileName);

    } else {

        // 다른 작업
    }
```