

## File Upload/Download Service

### 1. Summary

The **e-government framework** adopts Commons FileUpload providing various file upload API as an open source.

Spring also guides the single file upload using [Commons FileUpload](#).

While Spring currently provides excellent api for the single file upload, an error occurs when you try to upload several files of the same name at the time of **multiple file upload**.

See [multipart multi file upload support problem](#) for details of error.

This manual explains the alternatives that enables **multiple file uploads** rather than single file upload.

### 2. Description

#### Data Transfer Type

Type of data transfer includes GET type and POST type, as well as "multipart/form-data" of ENCTYPE property.

##### \* GET Type

Since the form data is exposed in URL, there is a limit in the contents of input. Data up to 256byte~4096byte can be transmitted.

##### \* POST Type

There is no limit in the contents of input since the data is transmitted without exposed to URL.

While it seems there will be no problem in transmitting data this way, there is a limit in the amount of data that can be sent.

A problem may occur when transmitting file or data with big capacity.

Form data transmission type used at this time is "multipart/form-data" of ENCTYPE property.

EgovFrameWork provides [CommonsMultipartResolver](#) class processing file upload using Apache Commons FileUpload API provided by spring, and register CommonsMultipartResolve to bean in the setting file.

Apache Commons FileUpload provides excellent api for single file upload. However, an error occurs when you try to upload several files with same name at the time of multiple file upload.

See [multipart multi file upload support problem](#) for the problems of error when you upload several files.

Since it was guided in details in [Spring's multipart \(fileupload\) support](#) regarding file upload using multipart in Spring, it will not be covered in this manual.

You can download the web application sample project in the eclipse project form that contains the sample code presented by the contents to help understand the execution of function.

#### File Upload / Download

See the following page for more details on File Upload / Download.

- [File Upload](#)
- [File Download](#)
- [multipart multi file upload support problem](#)

#### Execute Example Sample

1. Download utilappSample file from SVN.
2. Select the folder to download from Eclipse and import the project.
3. Check whether there is library file in lib.
4. Select Index.jsp under src folder, right-click to execute Run As > Run On Server.
5. Check that Tomcat is executed normally in Console window.
6. In this sample, MultiCommonsMultipartResolver complementing CommonsMultipartResolver is used, but it is recommended to use **CommonsMultipartResolver**.

Please note that we have guided the method of processing by adding other form name if adding JavaScript that uses multiple files with a single file(one column sample).

### A Screen Executed at Browser when Executing Index.jsp of Sample utilappSample



### N. Reference

- [Spring's multipart \(fileupload\) support {mvc-multipart}](#)
- [Spring's multipart \(fileupload\) support {CommonsMultipartResolver.html}](#)

- [Spring's multipart \(fileupload\) support {RequestFacade}](#)
- [Spring's multipart \(fileupload\) support {MultipartHttpServletRequest}](#)
- [Spring's multipart \(fileupload\) support {CommonsFileUploadSupport}](#)
- [Spring's multipart \(fileupload\) support {CommonsFileUploadSupport}](#)
- [Spring's multipart \(fileupload\) support {parseFileItems}](#)
- [commons.apache.org {fileupload}](#)
- [commons.apache.org{empty-parse}](#)