



---

## **Framework Corporatiu J2EE**

### **Servei d'Upload de Fitxers**

**Versió 1.3**

Barcelona, 16 / març / 2007



## Històric de modificacions

Data	Autor	Comentaris	Versió
16/01/2006	Atos Origin, sae openTrends	Versió inicial del document	1.0
13/02/2007	Atos Origin, SAE	Versió 1.2 d'OpenFrame	1.2
16/03/2007	Atos Origin, SAE	Versió 1.3 d'OpenFrame	1.3

### Llegenda de Marcadors



## Índex

<b>1.</b>	<b>INTRODUCCIÓ .....</b>	<b>4</b>
1.1.	PROPÓSIT .....	4
1.2.	CONTEXT I ESCENARIS D'ÚS .....	4
1.3.	VERSIONS I DEPENDÈNCIES .....	4
1.3.1.	<i>Versions</i> .....	4
1.3.2.	<i>Dependències Bàsiques</i> .....	5
1.4.	A QUI VA DIRIGIT .....	5
1.5.	DOCUMENTS I FONTS DE REFERÈNCIA .....	5
1.6.	GLOSSARI .....	5
<b>2.</b>	<b>DESCRIPCIÓ DETALLADA .....</b>	<b>6</b>
2.1.	ARQUITECTURA I COMPONENTS .....	6
2.1.1.	<i>Interfícies i Components Genèrics</i> .....	6
2.1.2.	<i>Implementació basada en Spring</i> .....	7
2.2.	INSTAL·LACIÓ I CONFIGURACIÓ .....	8
2.2.1.	<i>Instal·lació</i> .....	8
2.2.2.	<i>Configuració</i> .....	8
2.3.	UTILITZACIÓ DEL SERVEI .....	10
2.3.1.	<i>Definició de la pàgina JSP</i> .....	10
2.3.2.	<i>Accés des de la classe Action</i> .....	10
2.4.	EINES DE SUPORT .....	12
2.5.	INTEGRACIÓ AMB ALTRES SERVEIS .....	12
2.5.1.	<i>Integració amb el Servei Multiidioma i Excepcions</i> .....	12
2.6.	PREGUNTES FREQUÈNTS .....	12
<b>3.</b>	<b>EXEMPLES .....</b>	<b>13</b>
3.1.	EXEMPLE .....	13
3.1.1.	<i>Exemple de formulari</i> .....	13
3.1.2.	<i>Exemple d'accés des de la Action</i> .....	13
<b>4.</b>	<b>ANNEXOS .....</b>	<b>14</b>

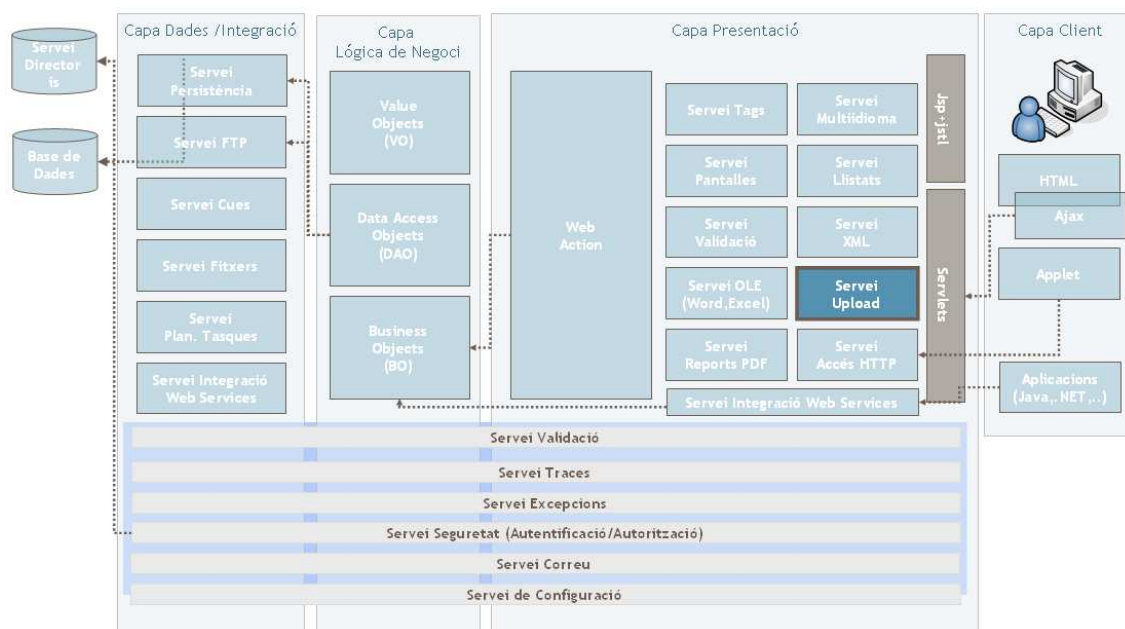
## 1. Introducció

### 1.1. Propòsit

Aquest servei permet que puguem llegir tractar els fitxers que hagin adjuntat els usuaris des dels formularis HTML en el seu navegador Web. Aquest tractament pot ser llegir el fitxer de forma directa, desar-lo al sistema de fitxers o per exemple emmagatzemar-lo en una base de dades.

### 1.2. Context i Escenaris d'Ús

El Servei de Upload de Fitxers es troba ubicat dins els serveis continguts a la capa de Presentació de openFrame.



### 1.3. Versions i Dependències

En el present apartat es mostren quines són les versions i dependències necessàries per fer ús del Servei.

#### 1.3.1. Versions

No s'han produït canvis respecte la versió 1.2.



### 1.3.2. Dependències Bàsiques

Nom	Tipus	Versió	Descripció
commons-beanutils-core	jar	1.7.0	
commons-fileupload	jar	1.0	<a href="http://jakarta.apache.org/commons">http://jakarta.apache.org/commons</a>
commons-logging	jar	1.0.4	<a href="http://jakarta.apache.org/commons">http://jakarta.apache.org/commons</a>
commons-logging-api	jar	1.0.4	
jsp-api	jar	2.0	
junit	jar	3.8.1	
log4j	jar	1.2.12	
maven-news-plugin	plugin	1.0	<a href="http://maven-plugins.sourceforge.net">http://maven-plugins.sourceforge.net</a>
maven-jalopy-plugin	plugin	1.3.1	
openFrame-services-configuration	jar	1.0	
openFrame-services-exceptions	jar	1.0	
openFrame-services-logging	jar	1.0	
openFrame-services-web	jar	1.0	
servlet-api	jar	2.4	
spring	jar	1.2.5	
spring-mock	jar	1.2.5	
struts	jar	1.2.7	

## 1.4. A qui va dirigit

Aquest document va dirigit als següents perfils:

- Programador. Per conèixer l'ús del servei
- Arquitecte. Per conèixer quins són els components i la configuració del servei

## 1.5. Documents i Fonts de Referència

- [1] Spring File <http://static.springframework.org/spring/docs/1.2.x/reference/mvc.html#mvc-multipart>  
Upload

## 1.6. Glossari

## 2. Descripció Detallada

### 2.1. Arquitectura i Components

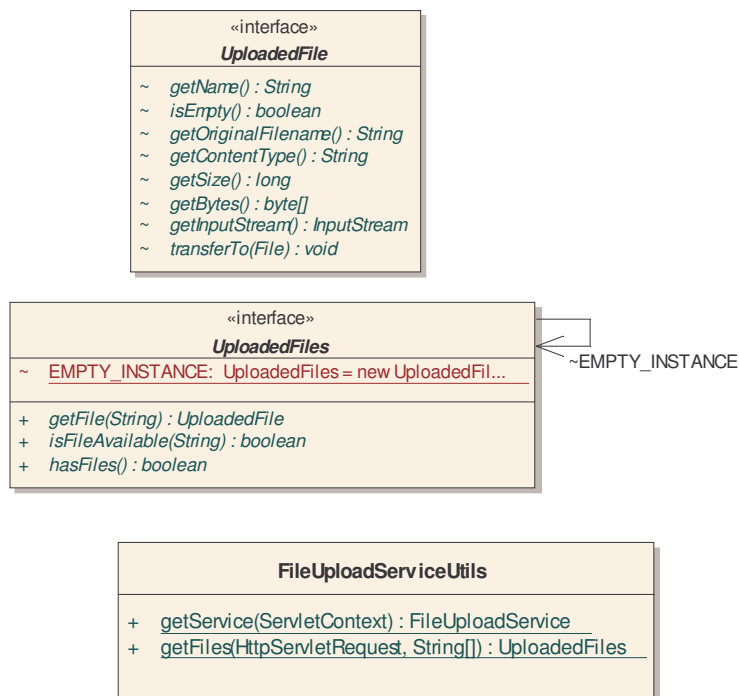
Els components podem classificar-los en:

- Interfícies i Components Genèrics. Interfícies del servei i components d'ús general amb independència de la implementació escollida.
- Implementació basada en Spring

#### 2.1.1. Interfícies i Components Genèrics

El servei defineix les següents interfícies i components genèrics:

Component	Package	Descripció
FileUploadService	net.opentrends.openframe.services.fileupload	Interfície que permet obtenir els fitxers de la petició HTTP

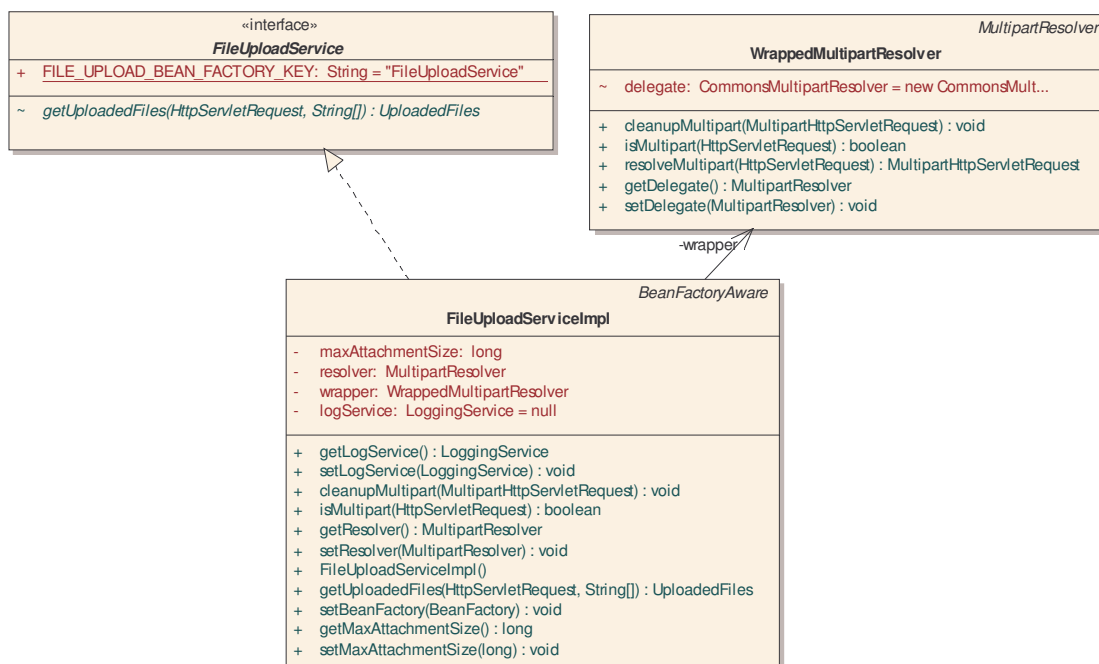


UploadedFiles	net.opentrends.openframe.services.fileupload	Interfície que defineix el mapa de fitxers adjuntats a la petició
UploadedFile	net.opentrends.openframe.services	Interfície per l'obtenció del contingut i

Component	Package	Descripció
	.fileupload	informació d'un fitxer concret de la llista de 'UploadedFiles'
FileUploadServiceException	net.opentrends.openframe.services.fileupload.exception	Excepció llençada pel servei
FileUploadServiceUtils	net.opentrends.openframe.services.fileupload	Classe utilitat que permet accedir al servei a partir del servletcontext o obtenir els fitxers de forma directa

### 2.1.2. Implementació basada en Spring

Component	Package	Descripció
FileUploadServiceImpl	net.opentrends.openframe.services.fileupload.impl	Implementació de la interfície 'FileUploadService' basada en Spring
UploadedFilesImpl	net.opentrends.openframe.services.webservices.impl	Implementació de la interfície 'UploadedFiles'. Conté una classe inner denominada 'MultipartFileWrapper' que implementa la interfície 'UploadedFile'



UploadedFilesImpl
- files: HashMap = new HashMap()
+ UploadedFilesImpl(List)
+ getFile(String) : UploadedFile
+ isFileAvailable(String) : boolean
+ add(String, MultipartFile) : void
+ hasFiles() : boolean

## 2.2. Instal·lació i Configuració

### 2.2.1. Instal·lació

La instal·lació del servei requereix de la utilització de la llibreria 'openFrame-services-fileupload' i les dependències indicades a l'apartat 'Introducció-Versions i Dependències'.

### 2.2.2. Configuració

La configuració del Servei implica els següents passos:

- 1) Definir la implementació del Servei que s'usarà
- 2) Definir la localització del fitxer de propietats del servei
- 3) Definir les propietats del servei

#### Definició del Servei

```
<bean id="fileUploadService"
...>
```

#### ***Fitxer de configuració: openFrame-services-fileupload.xml***

*Ubicació proposada:* <PROJECT\_ROOT>/src/main/resources/spring

Indicar el bean del servei (amb id 'fileUploadService') i la implementació escollida. Per la implementació 'FileUploadServiceImpl' definirem les següents propietats:

Propietat	Requerit	Descripció
maxAttachmentSize	No	Màxim tamany permès abans que es rebutgin els fitxers. Valor per defecte: 1 (no hi ha límit). Usar millor una referència per definir el seu valor en un fitxer extern (veure 'Definició de les Propietats del Servei')
logService	No	Referència al Servei de Traces
resolver	Sí	Detector de càrrega de fitxers . Usar el valor 'org.springframework.web.multipart.commons.CommonsMultipartResolver'



Propietat	Requerit	Descripció

Exemple:

```
<bean id="uploadResolver"
class="org.springframework.web.multipart.commons.CommonsMultipartResolver">
</bean>

<bean id="fileUploadService"
class="net.opentrends.openframe.services.fileupload.impl.FileUploadServiceImpl">
  <property name="maxAttachmentSize"
value="{fileUploadService.maxAttachmentSize}"/>
  <property name="logService" ref="loggingService"/>
  <property name="resolver" ref="uploadResolver"/>
</bean>
```

#### Definició de la Localització del Fitxer de Propietats del Servei

```
<bean id="configManager"
...>
```

#### *Fitxer de configuració: openFrame-services-configuration.xml*

Ubicació proposada: <PROJECT\_ROOT>/src/main/resources/spring

Seguint el propòsit general del Servei de Configuració definirem a la propietat 'basePropertyFiles' una nova localització pel fitxer de propietats del servei.

Exemple:

```
<bean id="configurationService"
class="net.opentrends.openframe.services.configuration.springframework.beans.factory.config.HostPropertyPlaceholderConfigurer">
  <property name="basePropertyFiles">
    <list>
      <value>classpath:jdbc/jdbc.properties</value>
      <value>classpath:mail/mail.properties</value>
      <value>classpath:file/fileUploadService.properties</value>
      <value>classpath:file/fileService.properties</value>
    </list>
  </property>
</bean>
```

#### Definició de les Propietats del Servei



### *Fitxer de configuració: fileUploadService.properties*

Ubicació proposada: <PROJECT\_ROOT>/src/main/resources/file

En aquest fitxer definirem les propietats que permet el servei (veure 'Definició del Servei')  
Exemple:

```
fileUploadService.maxAttachmentSize=1024
```

## **2.3. Utilització del Servei**

La utilització del servei comporta 2 passos:

- 1) Definir la pàgina JSP
- 2) Definir l'accés des de la classe Action

### **2.3.1. Definició de la pàgina JSP**

A la pàgina JSP cal fer tenir en consideració:

- 1) En el tag <form> (veure 'Servei de Tags') assegurar-se de definir les propietats i valors:
  - method="post"
  - enctype="multipart/form-data"

Exemple:

```
<fwk:form action="files.do" styleId="actionForm" reqCode="upload"
width="500" method="post" enctype="multipart/form-data">
```

- 2) Usar el tag '<file>' (veure document 'Servei de Tags')

```
<fwk:file styleId="tramitFile" property="tramitFile"/>
```

### **2.3.2. Accés des de la classe Action**

openFrame exposa una interfície d'utilització, "net.opentrends.openframe.fileupload.FileUploadService", que amaga la implementació real. La interfície té el següent aspecte:



```
package net.opentrends.openframe.services.fileupload;

...

/**
 * Interface to detect fileUploads.
 *
 */
public interface FileUploadService {
    public static final String
FILE_UPLOAD_BEAN_FACTORY_KEY="FileUploadService";
    UploadedFiles getUploadedFiles(HttpServletRequest request, String[]
fieldsToBind);
}
```

Aquesta interfície només exposa un mètode 'getUploadedFiles' que a partir del 'HttpServletRequest' retorna un objecte de tipus UploadedFiles. Aquest objecte permet obtenir cadascun dels fitxers enviats.

Així, podem realitzar les següents crides:

- getFile(String tagId): obtenir el contingut del fitxer que s'adjunta en el camp del formulari amb el nom indicat a 'tagId'
- isFileAvailable(String tagId): comprovar que s'ha adjuntat un fitxer en el camp del formulari amb nom indicat amb 'tagId'
- hasFiles: comprovar si s'han adjuntat fitxers

```
public interface UploadedFiles {
    ...
    public UploadedFile getFile(String tagId);
    public boolean isFileAvailable(String tagId);
    public boolean hasFiles();
}
```

Per últim, una vegada obtingut un 'UploadFile' en concret podem obtenir la seva informació i les seves dades de diverses formes (veure la signatura dels mètodes continguts a la interfície).

```
public interface UploadedFile {
    String getName();
    boolean isEmpty();
    String getOriginalFilename();
    String getContentType();
    long getSize();
    byte[] getBytes() throws IOException;
    InputStream getInputStream() throws IOException;
    void transferTo(File dest) throws IOException,
        IllegalStateException;
}
```



Val la pena observar que mitjançant el mètode `getInputStream` es possible llegir l'objecte per tal de transferir-lo per entrada-sortida java convencional, però a més el mètode “`transferTo`” possibilita exportar el fitxer pujat a un destí de fitxer en el servidor.

Si hi ha qualsevol error com per exemple que excedeixi el límit configurable de tamany de fitxers, la implementació llença una “`net.opentrends.openframe.services.fileupload.exception.FileServiceException`” on s'especifiquen els detalls de l'error (error, codi d'error, capa i subsistema).

## 2.4. Eines de Suport

## 2.5. Integració amb Altres Serveis

### 2.5.1. Integració amb el Servei Multiidioma i Excepcions

El Servei de Upload de Fitxers llença diverses excepcions amb codis per tal que el missatge pugui ser internacionalitzat. En el següent exemple es mostren els codis que s'han de definir:

```
openFrame.services.fileupload.attachment_size_exceeded=Attachment size exceeded: {0} bytes (max: {1})
openFrame.services.fileupload.error_resolving_multipart_uploads=Error retrieving
file {0} from request
```

## 2.6. Preguntes Freqüents

## 3. Exemples

### 3.1. Exemple

#### 3.1.1. Exemple de formulari

```
<fwk:form action="files.do" styleId="actionForm" reqCode="upload" width="500"
method="post" enctype="multipart/form-data">
    <fwk:text styleId="currentDir" property="currentDir"/>
    <fwk:file styleId="file" property="file"/>
    <fwk:submit styleId="uploadActionImage"/>
</fwk:form>
```

#### 3.1.2. Exemple d'accés des de la Action

```
UploadedFiles files = FileUploadServiceUtils.GetFiles(
    context.getRequest(),
    new String[] { "file1" });

if (files.hasFiles()) {
    if (files.isFileAvailable("file1")) {
        UploadedFile file1 = files.getFile("file1");
    }
}
```

En aquest cas hem fet servir el mètode estàtic “getFiles” de la classe auxiliar “net.opentrends.openframe.services.fileupload.FileUploadServiceUtils”. Aquest mètode permet utilitzar el servei sense haver de localitzar-lo expressament.



## **4. Annexos**