Generalitat de Catalunya
Centre de Telecomunicacions
i Tecnologies de la Informació

Framework Corporatiu J2EE

Servei de Fitxers

Versió 1.1

Barcelona, 2 / octubre / 2006

Històric de modificacions

Data	Autor	Comentaris	Versió
13/01/2006	Atos Origin, sae openTrends	Versió inicial del document	1.0
02/10/2006	Atos Origin	Versió 1.1 openFrame	1.1

Llegenda de Marcadors

Índex

1. Introducció

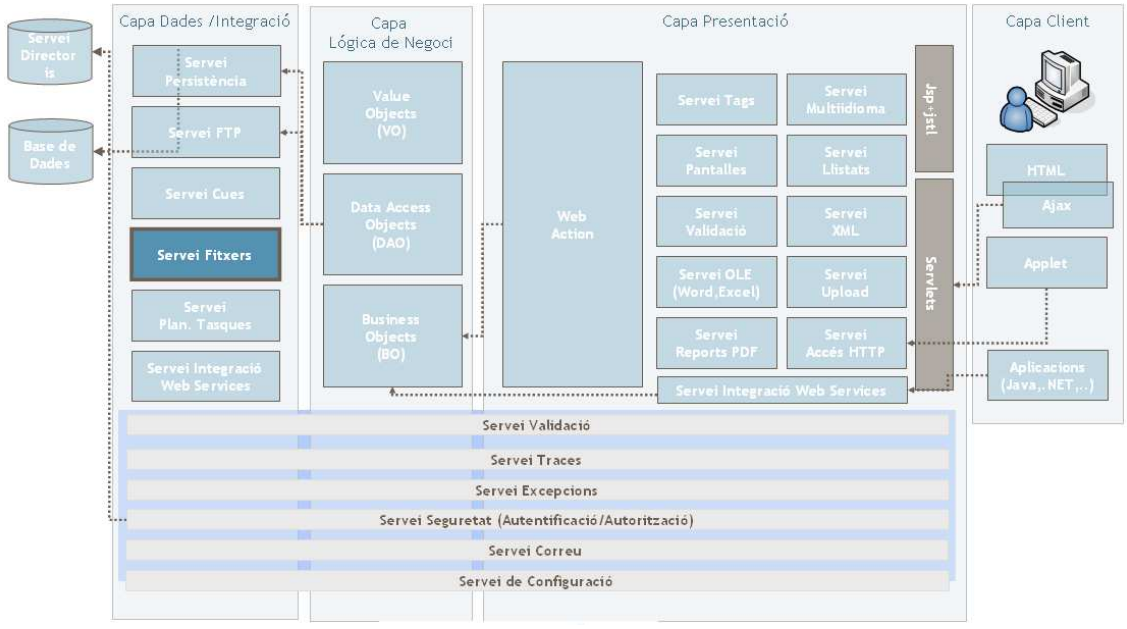
1.1. Propósit

El Servei de Fitxers té com a objectiu principal oferir la possibilitat de gestió i accés a fitxers. Les seves característiques es troben diferenciades en:

- Creació, esborrat i cópia de fitxers
- Obtenció d'informació dels fitxers (path, nom, extensió, etc.)
- Manipulació i obtenció del contingut dels fitxers mitjançant l'ús de streams o fluxos d'entrada/sortida

1.2. Context i Escenaris d'Ús

El Servei de Fitxers es troba dins dels serveis d'integració de openFrame.



1.3. Versions i Dependències

1.3.1. Versions

No hi ha canvis respecte la versió 1.0

1.3.2. Dependències

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

Nom	Tipus	Versió	Descripció
commons-io	jar	1.1	http://jakarta.apache.org/commons

1.4. A qui va dirigit

Aquest document va dirigit als següents perfils:

- Programador. Per conéixer l'ús del servei
- Arquitecte. Per conéixer quins són els components i la configuració del servei
- Administrador. Per conéixer cóm configurar el servei en cadascun dels entorns en cas de necessitat

1.5. Documents i Fonts de Referència

[1] Commons-io http://jakarta.apache.org/commons/io/

1.6. Glossari

Commons IO

Llibreria de Jakarta en Java que permet la interacció amb fitxers

2. Descripció Detallada

2.1. Arquitectura i Components

openFrame ofereix una arquitectura de gestió de fitxers deslligada de la implementació.

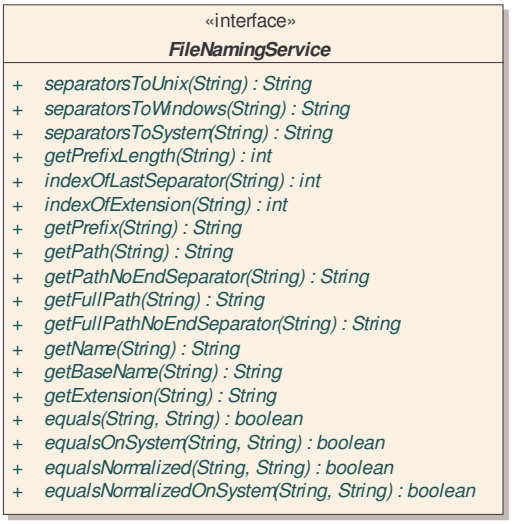
Actualment, i a causa del seu grau de facilitat openFrame es basa en la llibreria del projecte Jakarta Commons IO. No obstant s'han encapsulat els mètodes "static" de la classes utilitàries amb interfícies per a garantir l'estabilitat de l'API a llarg termini.

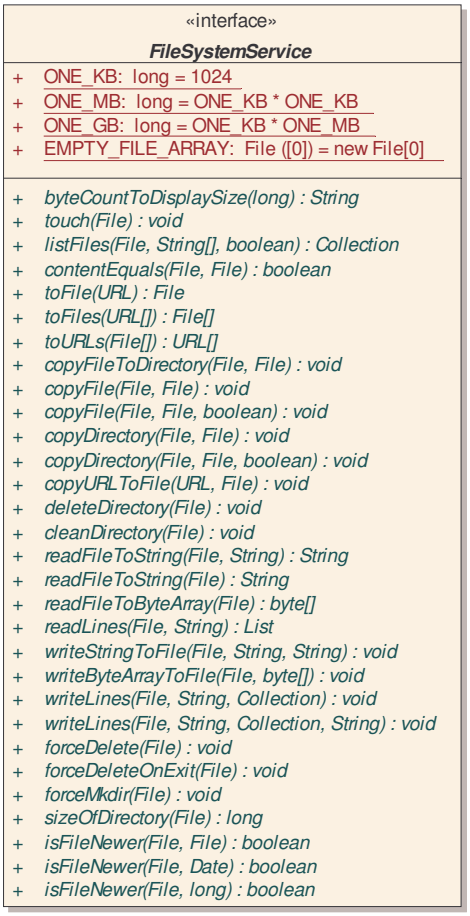
Els components podem classificar-los en:

- Interfícies i Components Genérics. Interfícies del servei i components d'ús general amb independència de la implementació escollida.
- Implementació de les interfícies basada en Commons IO

2.1.1. Interfícies i Components Genérics

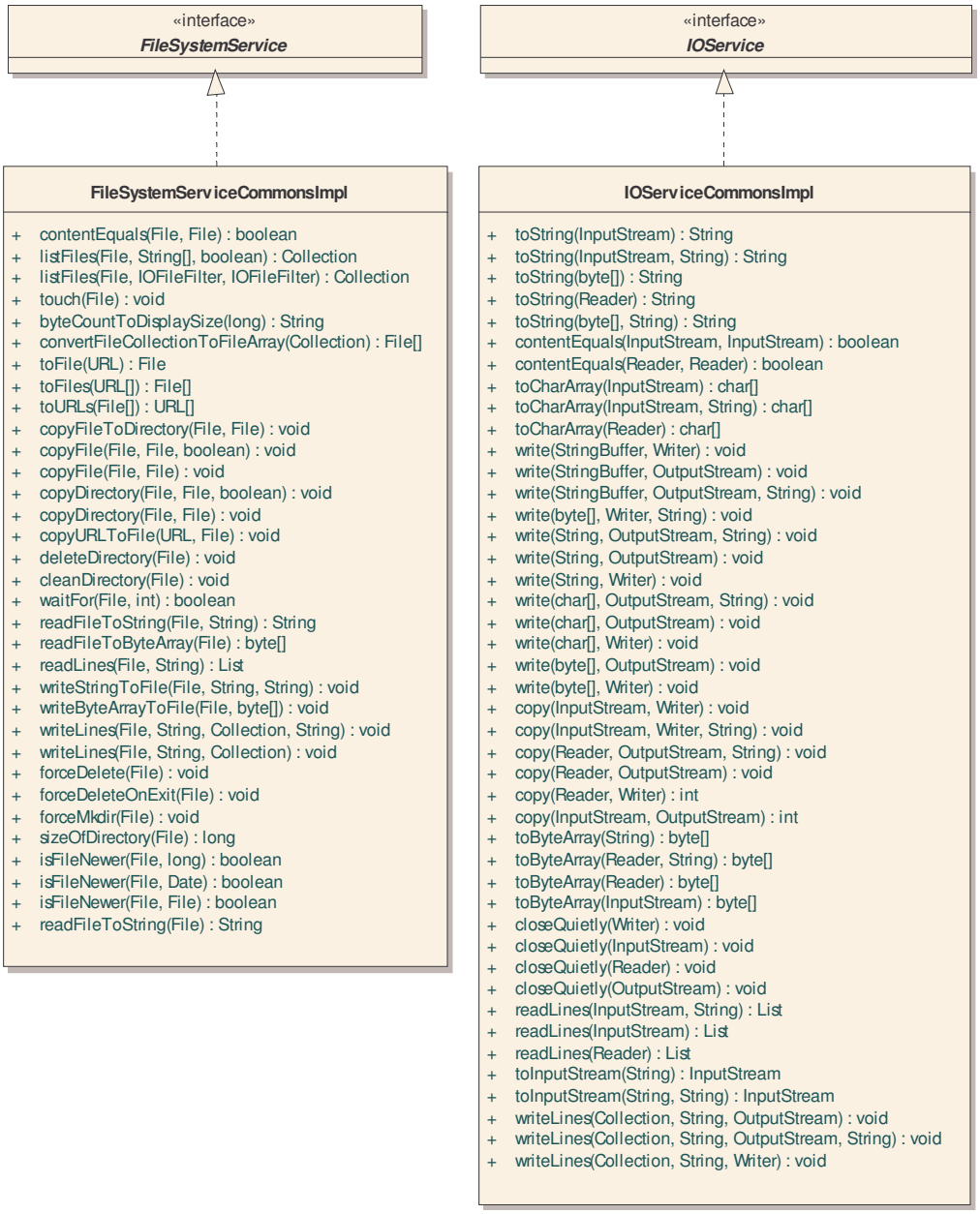
Component	Package	Descripció
FileSystemService	net.opentrends.openframe.services.file	Interfície que defineix operacions de creació, esborrat i cópia de fitxers
FileNamingService	net.opentrends.openframe.services.file	Interfície d'obtenció d'informació sobre els fitxers (path, nom, extensió, etc.)
IOService	net.opentrends.openframe.services.file	Interfície de manipulació de fitxers mitjançant fluxos input/output


«interface»
FileNamingService
+ separatorsToUnix(String) : String
+ separatorsToWindows(String) : String
+ separatorsToSystem(String) : String
+ getPrefixLength(String) : int
+ indexOfLastSeparator(String) : int
+ indexOfExtension(String) : int
+ getPrefix(String) : String
+ getPath(String) : String
+ getPathNoEndSeparator(String) : String
+ getFullPath(String) : String
+ getFullPathNoEndSeparator(String) : String
+ getName(String) : String
+ getBaseName(String) : String
+ getExtension(String) : String
+ equals(String, String) : boolean
+ equalsOnSystem(String, String) : boolean
+ equalsNormalized(String, String) : boolean
+ equalsNormalizedOnSystem(String, String) : boolean


«interface»
FileSystemService
+ ONE_KB: long = 1024
+ ONE_MB: long = ONE_KB * ONE_KB
+ ONE_GB: long = ONE_KB * ONE_MB
+ EMPTY_FILE_ARRAY: File ([0]) = new File[0]
+ byteCountToDisplaySize(long) : String
+ touch(File) : void
+ listFiles(File, String[], boolean) : Collection
+ contentEquals(File, File) : boolean
+ toFile(URL) : File
+ toFiles(URL[]) : File[]
+ toURLs(File[]) : URL[]
+ copyFileToDirectory(File, File) : void
+ copyFile(File, File) : void
+ copyFile(File, File, boolean) : void
+ copyDirectory(File, File) : void
+ copyDirectory(File, File, boolean) : void
+ copyURLToFile(URL, File) : void
+ deleteDirectory(File) : void
+ cleanDirectory(File) : void
+ readFileToString(File, String) : String
+ readFileToString(File) : String
+ readFileToByteArray(File) : byte[]
+ readLines(File, String) : List
+ writeStringToFile(File, String, String) : void
+ writeByteArrayToFile(File, byte[]) : void
+ writeLines(File, String, Collection) : void
+ writeLines(File, String, Collection, String) : void
+ forceDelete(File) : void
+ forceDeleteOnExit(File) : void
+ forceMkdir(File) : void
+ sizeOfDirectory(File) : long
+ isFileNewer(File, File) : boolean
+ isFileNewer(File, Date) : boolean
+ isFileNewer(File, long) : boolean


«interface»
IOService
+ DIR_SEPARATOR_UNIX: char = '/'
+ DIR_SEPARATOR_WINDOWS: char = '\\'
+ DIR_SEPARATOR: char = File.separatorChar
+ LINE_SEPARATOR_UNIX: String = "\n"
+ LINE_SEPARATOR_WINDOWS: String = "\r\n"
+ closeQuietly(Reader) : void
+ closeQuietly(Writer) : void
+ closeQuietly(InputStream) : void
+ closeQuietly(OutputStream) : void
+ toByteArray(InputStream) : byte[]
+ toByteArray(Reader) : byte[]
+ toByteArray(Reader, String) : byte[]
+ toCharArray(InputStream) : char[]
+ toCharArray(InputStream, String) : char[]
+ toCharArray(Reader) : char[]
+ toString(InputStream) : String
+ toString(InputStream, String) : String
+ toString(Reader) : String
+ readLines(InputStream) : List
+ readLines(InputStream, String) : List
+ readLines(Reader) : List
+ toInputStream(String) : InputStream
+ toInputStream(String, String) : InputStream
+ write(byte[], OutputStream) : void
+ write(byte[], Writer) : void
+ write(byte[], Writer, String) : void
+ write(char[], Writer) : void
+ write(char[], OutputStream) : void
+ write(char[], OutputStream, String) : void
+ write(String, Writer) : void
+ write(String, OutputStream) : void
+ write(String, OutputStream, String) : void
+ write(StringBuffer, Writer) : void
+ write(StringBuffer, OutputStream) : void
+ write(StringBuffer, OutputStream, String) : void
+ writeLines(Collection, String, OutputStream) : void
+ writeLines(Collection, String, OutputStream, String) : void
+ writeLines(Collection, String, Writer) : void
+ copy(InputStream, OutputStream) : int
+ copy(InputStream, Writer) : void
+ copy(InputStream, Writer, String) : void
+ copy(Reader, Writer) : int
+ copy(Reader, OutputStream) : void
+ copy(Reader, OutputStream, String) : void
+ contentEquals(InputStream, InputStream) : boolean
+ contentEquals(Reader, Reader) : boolean

2.1.2. Components Implementació Commons IO



Component	Package	Descripció
FileSystemServiceCommonsImpl	net.opentrends.openframe.services.file.impl.commons	Implementació que defineix operacions de creació, esborrat i cópia de fitxers basada en Commons IO

Component	Package	Descripció
FileNamingServiceCommonsImpl	net.opentrends.openframe.services.file.impl.commons	Implementació de la interfície d'obtenció d'informació sobre els fitxers (path, nom, extensió, etc.) basada en Commons IO
IOServiceCommonsImpl	net.opentrends.openframe.services.file.impl.commons	Implementació de la interfície de manipulació de fitxers mitjançant fluxos input/output basada en Commons IO

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-file' i les dependències indicades a l'apartat 'Introducció-Versions i Dependències'.

2.2.2. Configuració

Fitxer de configuració: openFrame-services-exceptions.xml

Ubicació proposada: <PROJECT_ROOT>/src/main/resources/spring

La configuració del Servei de Fitxers és força senzilla, doncs només cal definir els beans que farem servir com a implementacions. En l'actualitat implica la definició dels següents beans:

- fileSystemService. Indicar la implementació escollida en el camp class i indicar les propietats:

Propietat	Requerit	Descripció
lazy-init	Sí	Indicar 'true'
singleton	No	Indicar 'true', doncs els mètodes són estàtics i no es poden crear instàncies. Per defecte és 'true'

- fileNamingService. Indicar la implementació escollida en el camp class i indicar les mateixes propietats que 'fileSystemService'
- ioService. Indicar la implementació escollida en el camp class i indicar les mateixes propietats que 'fileSystemService'

Exemple:

```
<beans>

   <bean id="fileSystemService"
class="net.opentrends.openframe.services.file.impl.commons.FileSystemServiceCommon
sImpl"
          lazy-init="true"
```

```
            singleton="true"/>

    <bean id="fileNamingService"
class="net.opentrends.openframe.services.file.impl.commons.FileNamingServiceCommon
sImpl"
            lazy-init="true"
            singleton="true"/>

    <bean id="ioService"
class="net.opentrends.openframe.services.file.impl.commons.IOServiceCommonsImpl"
            lazy-init="true"
            singleton="true"/>

</beans>
```

2.3. Utilització del Servei

La utilització del Servei es basa en l´ús de les interfícies definides a l'apartat 'Arquitectura i Components'. Consultar l'apartat 'Exemples' per veure alguns exemples de tractament de fitxers amb la API proporcionada.

2.4. Eines de Suport

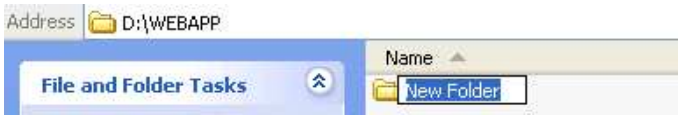
2.5. Integració amb Altres Serveis

2.6. Preguntes Freqüènts

3. Exemples

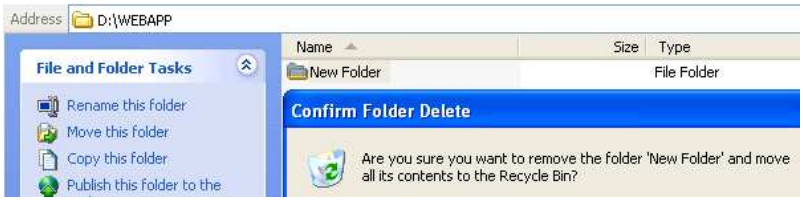
3.1. Exemples d'Operacions de Manteniment de Directoris i Fitxers

3.1.1. Crear una nova carpeta



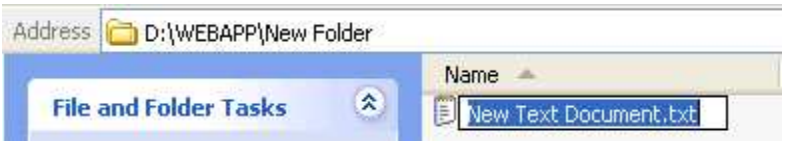

```
File newFolder = new File("D:/WEBAPP/New Folder");
fileSystemService.forceMkdir(newFolder);
```

3.1.2. Eliminar una carpeta




```
fileSystemService.deleteDirectory(newFolder);
```

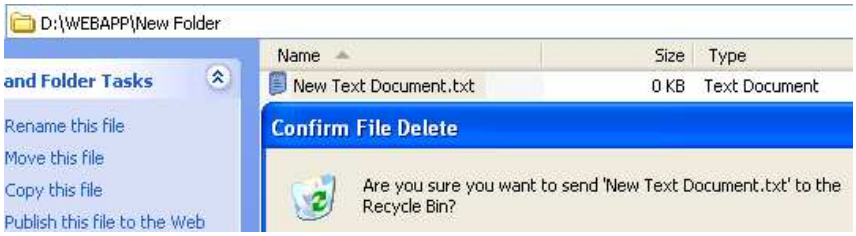
3.1.3. Crear un fitxer



Aquest comandament usa el constructor de java.io.File.

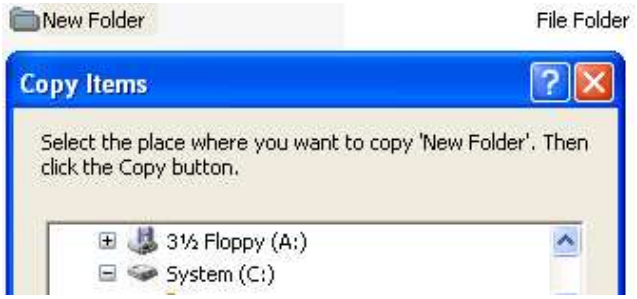
```
File newFile = new File("D:/WEBAPP/New Folder/New Text Document.txt");
```

3.1.4. Eliminar un fitxer



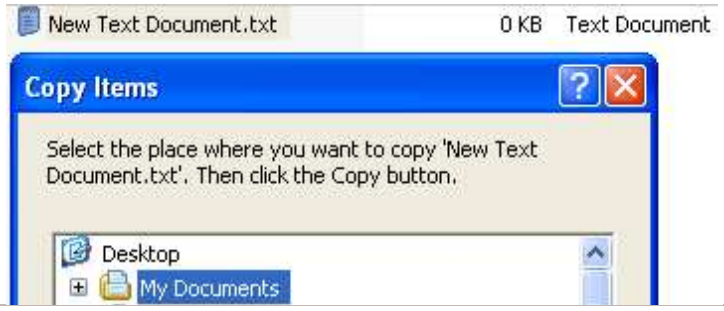

```
File file = new File("D:/WEBAPP/New Folder/New Text Document.txt");
fileSystemService.forceDelete(file);
```

3.1.5. Copiar una carpeta




```
File folder = new File("D:/WEBAPP/New Folder");
fileSystemService.copyDirectory(folder, new File("D:/WEBAPP/Destination"));
```

3.1.6. Copiar un fitxer




```
File sourceFile = new File("D:/WEBAPP/New Folder/New Text Document.txt");
fileSystemService.copyFileToDirectory(sourceFile, new File("D:/WEBAPP/Dest"));
```

3.1.7. Llegir un fitxer

1. **Cap a un String**

```
File srcFile = new File("D:/WEBAPP/New Folder/New Text Document.txt");
String destinationString = fileSystemService.readFileToString(srcFile);
```

2. **Cap a una llista de cadenes**

```
File srcFile = new File("D:/WEBAPP/New Folder/New Text Document.txt");
List destinationList = fileSystemService.readLines(srcFile, "ISO-8859-1");
Iterator iter = destinationList.iterator();
while (iter.hasNext()) {
     String line = (String) iter.next();
     // Process...
}
```

3.2. Exemple d'Obtenció d'Informació dels Fitxers

En aquest exemple es pot observar cóm podem obtenir informació d'un fitxer.

```
File file = new File("D:/WEBAPP/New Folder/New Text Document.txt");

String prefix = fileNamingService.getPrefix(file.getPath());
// => D:/

String path= fileNamingService.getPath(file.getPath());
// => WEBAPP/New Folder/

String baseName = fileNamingService.getBaseName(file.getPath());
// => New Text Document

String extension = fileNamingService.getExtension(file.getPath());
// => txt
```

3.3. Exemple de Lectura del Contingut d'un Recurs

IOService conté mètodes utilitaris per a llegir, escriure i llegir amb fluxos de tipus InputStream, OutputStream, Reader i Writer.

Per exemple, podríem llegir els bytes des d'una URL, i imprimir-los en la pantalla. Amb les classes de Java clàssiques del package java.io, el codi seria el següent:

```
InputStream in = new URL( "http://jakarta.apache.org" ).openStream();
    try {
        InputStreamReader inR = new InputStreamReader( in );
        BufferedReader buf = new BufferedReader( inR );
        String line;
```

```
            while ( ( line = buf.readLine() ) != null ) {
                System.out.println( line );
            }
        } finally {
            in.close();
        }
```

Mentre que amb l'ús del servei IOService, el codi serà:

```
InputStream in = new URL( "http://jakarta.apache.org" ).openStream();
    try {
        System.out.println( ioService.toString( in ) );
    } finally {
        IOUtils.closeQuietly(in);
    }
```

Cóm veiem, hem guanyat en senzillesa.

4. Annexos