



API Administración w@rdA

Consejería de Justicia y Administración Pública
Junta de Andalucía

Mayo 2005
v0.9.0

Tabla de Contenidos

DbConnectionConfig	4
ieci.tecdoc.idoc.admin.api	6
ObjFactory	7
ieci.tecdoc.idoc.admin.api.archive	14
Archive	15
ArchiveAccess	22
ArchiveDefs	26
ArchiveFld	28
ArchiveFlds	32
Archiveldx	36
Archiveldxs	37
ArchiveMisc	39
ArchivePerms	41
ArchiveUpdInfo	43
Archives	48
BasicArchive	49
BasicArchives	49
BasicDirectories	50
BasicDirectory	51
Directories	51
Directory	52
ieci.tecdoc.idoc.admin.api.exception	57
AdminException	57
AdminResourceFile	58
ArchiveErrorCodes	58
DeptErrorCodes	61
DirErrorCodes	62
GroupErrorCodes	63
LoginErrorCodes	64
RepositoryErrorCodes	64
UserErrorCodes	66
VolumeErrorCodes	67
VolumeListErrorCodes	69
ieci.tecdoc.idoc.admin.api.system	71

SystemCfg	71
ieci.tecdoc.idoc.admin.api.user	72
BasicUser	73
BasicUsers	74
Department	74
Departments	79
Group	81
GroupUserRel	86
Groups	87
LdapGroup	88
LdapGroups	92
LdapUser	93
LdapUsers	97
Login	99
Permission	100
Permissions	102
User	103
UserAccess	109
UserDefs	115
UserProfile	117
UserProfiles	119
Users	120
ieci.tecdoc.idoc.admin.api.volume	123
Repositories	123
Repository	125
Volume	131
VolumeDefs	137
VolumeList	139
VolumeLists	143
Volumes	145



Class DbConnectionConfig

```
java.lang.Object
|
+--ieci.tecdoc.core.db.DbConnectionConfig
```

< [Constructores](#) > < [Metodos](#) >

```
public class DbConnectionConfig
extends java.lang.Object
```

Clase que contiene la configuración de conexión con la base de datos.

Constructores

DbConnectionConfig

```
public DbConnectionConfig(java.lang.String ctxName,
                           java.lang.String user,
                           java.lang.String pwd)
```

Constructor

Parametros:

ctxName - nombre del dataSource
user - usuario de base de datos
pwd - contraseña de base de datos

DbConnectionConfig

```
public DbConnectionConfig(java.lang.String drvClsName,
                           java.lang.String url,
                           java.lang.String user,
                           java.lang.String pwd)
```

Constructor

Parametros:

drvClsName - nombre de la clase del driver de base de datos
url - url de base de datos
user - usuario de base de datos
pwd - contraseña de base de datos

Metodos



getPwd

```
public java.lang.String getPwd()
```

Devuelve la contraseña de base de datos

Retorno:

contraseña de base de datos

getUrl

```
public java.lang.String getUrl()
```

Devuelve la url de conexión de base de datos

Retorno:

url de conexión de base de datos

getUser

```
public java.lang.String getUser()
```

Devuelve el usuario de base de datos

Retorno:

usuario de base de datos

isCntByDriver

```
public boolean isCntByDriver()
```

Devuelve true si la conexión es por driver

Retorno:

true si la conexión es por driver

setCtxName

```
public void setCtxName(java.lang.String ctxName)
```

Establece el nombre del datasource

Parametros:

ctxName - nombre del datasource



setDrvClsName

```
public void setDrvClsName(java.lang.String drvClsName)
```

Establece el nombre de la clase del driver de base de datos

Parametros:

drvClsName - nombre de la clase del driver de base de datos

setPwd

```
public void setPwd(java.lang.String pwd)
```

Establece el valor de la contraseña de base de datos

Parametros:

pwd - contraseña de base de datos

setUrl

```
public void setUrl(java.lang.String url)
```

Establece el valor de la url de conexión a base de datos

Parametros:

url - url de conexión a base de datos

setUser

```
public void setUser(java.lang.String user)
```

Establece el valor del usuario de base de datos

Parametros:

user - usuario de base de datos

toString

```
public java.lang.String toString()
```

Reemplaza:

toString in class java.lang.Object

Paquete `ieci.tecdoc.idoc.admin.api`

Clases

[ObjFactory](#)



Factoría de creación de objetos de administración

ieci.tecdoc.idoc.admin.api

Class ObjFactory

```
java.lang.Object
|
+--ieci.tecdoc.idoc.admin.api.ObjFactory
```

< [Metodos](#) >

```
public class ObjFactory
extends java.lang.Object
```

Factoría de creación de objetos de administración

Metodos

createArchive

```
public static Archive createArchive(int userConnected,
                                     int parentId,
                                     boolean isLdap)
```

Crea una instancia (sin valores iniciales) de un archivador.

Parametros:

userConnected - Identificador del usuario conectado en la aplicación de administración.
 parentId - Identificador del directorio padre.
 isLdap - Indica si se está trabajando con Ldap o no.

Retorno:

La instancia del archivador creado.

createArchiveAccess

```
public static ArchiveAccess createArchiveAccess()
```

Crea la instancia para obtener el acceso al sistema de archivadores.

Retorno:

La instancia.

createArchiveFld

```
public static ArchiveFld createArchiveFld()
```

Crea una instancia de la estructura de un campo

Retorno:

la instancia de la estructura



createArchiveFlds

```
public static ArchiveFlds createArchiveFlds()
```

Crea la instancia de la estructura de campos

Retorno:

La instancia

createArchiveIdx

```
public static ArchiveIdx createArchiveIdx()
```

Crea la instancia de la estructura de un índice

Retorno:

La instancia

createArchiveIdxs

```
public static ArchiveIdxs createArchiveIdxs()
```

Crea la instancia de la estructura de índices

Retorno:

La instancia

createArchiveMisc

```
public static ArchiveMisc createArchiveMisc(java.lang.String fdrName,  
                                             int volListType,  
                                             int volListId)
```

Crea la instancia para información extra del archivador

Parametros:

fdrName - título de las carpetas
volListType - tipo de lista de volúmenes
volListId - Identificador de lista de volúmenes

Retorno:

La instancia

createArchivePerms

```
public static ArchivePerms createArchivePerms()
```

Crea una instancia (sin valores iniciales) de permisos de archivador

Retorno:

La instancia de los permisos de archivador



createArchiveUpdInfo

```
public static ArchiveUpdInfo createArchiveUpdInfo()
```

Crea la instancia de la estructura de actualización de un archivador

Retorno:

- la instancia

createArchiveUpdInfo

```
public static ArchiveUpdInfo createArchiveUpdInfo( java.lang.String name,  
                                                    java.lang.String remarks,  
                                                    int adminUserId,  
                                                    boolean ftsInContents,  
                                                    ArchiveFlds flds,  
                                                    ArchiveIdxs idxs,  
                                                    ArchiveMisc misc)
```

Crea la instancia de la estructura de actualización de un archivador

Parametros:

name - - nombre del archivador
remarks - - descripción
adminUserId - - identificador del administrador del archivador
ftsInContents - - búsqueda en contenido de fichero (true/false)
flds - - estructura con la definición de campos (antiguos y nuevos)
idxs - - estructura con la definición de índices (antiguos y nuevos)
misc - - estructura con la información extra del archivador

Retorno:

La instancia

createArchives

```
public static Archives createArchives()
```

Crea una instancia (sin valores inicialis) de lista de archivadores

Retorno:

La instancia

createDepartment

```
public static Department createDepartment()
```

Crea una instancia para un departamento invesDoc

Retorno:

La instancia



createDepartment

```
public static Department createDepartment(int userConnected,  
                                           int parentId)
```

Crea una instancia (sin valores iniciales) de un departamento.

Parametros:

userConnected - Identificador del usuario conectado en la aplicación de administración.
parentId - Identificador del padre de departamento que pertenece.

Retorno:

La instancia del departamento creado.

createDirectories

```
public static Directories createDirectories()
```

Crea una instancia (sin valores iniciales) de lista de directorios

Retorno:

La instancia

createDirectory

```
public static Directory createDirectory(int userConnected,  
                                          int parentId,  
                                          boolean isLdap)
```

Crea una instancia (sin valores iniciales) de un directorio.

Parametros:

userConnected - Identificador del usuario conectado en la aplicación de administración.
parentId - Identificador del directorio padre.
isLdap - Indica si se está trabajando con Ldap o no.

Retorno:

La instancia del directorio creado.

createGroup

```
public static Group createGroup()
```

Crea una instancia para un grupo invesDoc

Retorno:

La instancia



createGroup

```
public static Group createGroup(int userConnected)
```

Crea una instancia (sin valores iniciales) de un grupo.

Parametros:

userConnected - Identificador del usuario conectado en la aplicación de administración.

Retorno:

La instancia del grupo creado.

createGroupUserRel

```
public static GroupUserRel createGroupUserRel()
```

Crea una instancia para la relación grupos - usuarios.

Retorno:

La instancia

createLdapGroup

```
public static LdapGroup createLdapGroup(int userConnected)
```

Crea una instancia (sin valores iniciales) de un grupo Ldap.

Parametros:

userConnected - Identificador del usuario conectado en la aplicación de administración.

Retorno:

La instancia del grupo creado.

createLdapUser

```
public static LdapUser createLdapUser(int userConnected)
```

Crea una instancia (sin valores iniciales) de un usuario Ldap.

Parametros:

userConnected - Identificador del usuario conectado en la aplicación de administración.

Retorno:

La instancia del usuario creado.



createRepository

```
public static Repository createRepository(int userConnected)
```

Crea una instancia (sin valores iniciales) de un repositorio.

Parametros:

userConnected - Identificador del usuario conectado en la aplicación de administración.

Retorno:

La instancia del repositorio creado.

createSystem

```
public static SystemCfg createSystem()
```

Crea una instancia para el sistema invesDoc

Retorno:

La instancia

createUser

```
public static User createUser()
```

Crea una instancia para un usuario invesDoc

Retorno:

La instancia

createUser

```
public static User createUser(int userConnected,  
int deptId)
```

Crea una instancia (sin valores iniciales) de un usuario.

Parametros:

userConnected - Identificador del usuario conectado en la aplicación de administración.

deptId - Identificador del departamento.

Retorno:

La instancia del usuario creado.

createUserAccess

```
public static UserAccess createUserAccess()
```

Crea un instancia para obtener accesos al sistema de usuarios invesDoc.

Retorno:

La instancia



createVolume

```
public static Volume createVolume(int userConnected,  
                                   int repId)
```

Crea una instancia (sin valores iniciales) de un volumen.

Parametros:

userConnected - Identificador del usuario conectado en la aplicación de administración.
repId - Identificador del repositorio al que pertenece

Retorno:

La instancia del volumen creado.

createVolumeList

```
public static VolumeList createVolumeList(int userConnected)
```

Crea una instancia (sin valores iniciales) de una lista de volúmenes.

Parametros:

userConnected - Identificador del usuario conectado en la aplicación de administración.

Retorno:

La instancia de la lista de volúmenes creada.



Paquete `ieci.tecdoc.idoc.admin.api.archive`

Interfaces

[Archive](#)

Proporciona toda la funcionalidad necesaria para manejar un archivador.

[ArchiveAccess](#)

Maneja los accesos a la gestión de directorios y archivadores.

[ArchiveFld](#)

Proporciona toda la funcionalidad necesaria para manejar un campo del archivador.

[ArchiveFlds](#)

[Archiveldx](#)

[Archiveldxs](#)

[ArchiveMisc](#)

[ArchivePerms](#)

Proporciona toda la funcionalidad necesaria para manejar permisos sobre archivadores invesDoc.

[ArchiveUpdInfo](#)

[Archives](#)

Proporciona toda la funcionalidad necesaria para manejar archivadores invesDoc.

[BasicArchive](#)

[BasicArchives](#)

[BasicDirectories](#)

[BasicDirectory](#)

[Directories](#)

Proporciona toda la funcionalidad necesaria para manejar árbol de directorios invesDoc.

[Directory](#)

Proporciona toda la funcionalidad necesaria para manejar directorios invesDoc.

Clases

[ArchiveDefs](#)

Proporciona las definiciones de todos los identificadores relativos a la administración de directorios y archivadores.



ieci.tecdoc.idoc.admin.api.archive

Interface Archive

< [Metodos](#) >

public interface **Archive**

Proporciona toda la funcionalidad necesaria para manejar un archivador.

Metodos

create

```
public void create()  
    throws java.lang.Exception
```

Crea el archivador.

Lanza:

java.lang.Exception - Si se produce algún error al crear. Por ejemplo, el archivador ya existe.

delete

```
public void delete()  
    throws java.lang.Exception
```

Elimina el archivador cargado

Lanza:

java.lang.Exception - Si se produce algún error al eliminar

delete

```
public void delete(int archiveId)  
    throws java.lang.Exception
```

Elimina el archivador indicado

Parametros:

archiveId - Identificador del archivador

Lanza:

java.lang.Exception -



existsFdrsInArch

```
public boolean existsFdrsInArch()  
    throws java.lang.Exception
```

Obtiene si existen carpetas dadas de alta en el archivador

Retorno:

true / false

Lanza:

java.lang.Exception -

getAccessType

```
public int getAccessType()
```

Obtiene el tipo de acceso del archivador.

Retorno:

El tipo de acceso mencionado.

getAcsId

```
public int getAcsId()
```

Obtiene el identificador de acceso del archivador.

Retorno:

El identificador mencionado.

getAdminUserId

```
public int getAdminUserId()
```

Obtiene el identificador del administrador del archivador

Retorno:

El identificador mencionado.

getAdminUsers

```
public BasicUsers getAdminUsers()
```

Obtiene la lista de usuarios (identificadores y nombres) que pueden ser administradores de este archivador.

Retorno:

La lista mencionada.



getCreationDate

```
public java.util.Date getCreationDate()
```

Obtiene la fecha de creación del archivador.

Retorno:

La fecha mencionada.

getCreatorId

```
public int getCreatorId()
```

Obtiene el identificador del usuario que ha creado del archivador.

Retorno:

El identificador mencionado.

getFldsDef

```
public ArchiveFlds getFldsDef()
```

Obtiene información referente a la definición de campos del archivador.

Retorno:

Estructura con la información mencionada.

getId

```
public int getId()
```

Obtiene el identificador del archivador.

Retorno:

El identificador mencionado.

getIdxsDef

```
public ArchiveIdxs getIdxsDef()
```

Obtiene información referente a la definición de índices del archivador.

Retorno:

Estructura con la información mencionada.



getMiscDef

```
public ArchiveMisc getMiscDef()
```

Obtiene información referente a lista de volúmenes y título de carpeta del archivador.

Retorno:

Estructura con la información mencionada.

getName

```
public java.lang.String getName()
```

Obtiene el nombre del archivador.

Retorno:

El nombre mencionado.

getParentId

```
public int getParentId()
```

Obtiene el identificador del directorio padre del archivador cargado

Retorno:

El identificador del directorio padre

getParentId

```
public int getParentId(int archiveId)  
    throws java.lang.Exception
```

Obtiene el identificador del directorio padre del archivador

Parametros:

archiveId - Identificador del archivador

Retorno:

El identificador del directorio padre

Lanza:

java.lang.Exception -

getRemarks

```
public java.lang.String getRemarks()
```

Obtiene la descripción del archivador.

Retorno:

La descripción mencionada.



getUpdateDate

```
public java.util.Date getUpdateDate()
```

Obtiene la fecha de actualización del archivador.

Retorno:

La fecha mencionada.

getUpdaterId

```
public int getUpdaterId()
```

Obtiene el identificador del usuario que ha actualizado el archivador.

Retorno:

El identificador mencionado.

isFtsInContents

```
public boolean isFtsInContents()
```

Obtiene si hay búsqueda en contenido de fichero para el archivador

Retorno:

true / false

load

```
public void load(int id)
    throws java.lang.Exception
```

Carga un archivador.

Parametros:

id - Identificador del archivador.

Lanza:

java.lang.Exception - Si se produce algún error al leer la información del archivador.

setFldsDef

```
public void setFldsDef(ArchiveFlds fldsArch)
```

Establece información referente a la definición de campos del archivador

Parametros:

Estructura - con la información mencionada.



setFtsInContents

```
public void setFtsInContents(boolean ftsInContents)
```

Establece la existencia o no de búsqueda en contenido de fichero para el archivador

Parametros:

ftsInContents - (true /false)

setId

```
public void setId(int archId)
```

Establece el identificador del archivador.

Parametros:

name - El identificador del archivador.

setIdxsDef

```
public void setIdxsDef(ArchiveIdxs idxsArch)
```

Establece información referente a la definición de índices del archivador.

Parametros:

Estructura - con la información mencionada.

setLdap

```
public void setLdap(boolean ldap)
```

Indica si se está trabando con usuarios de un repositorio Ldap (true) o no (false).

Parametros:

ldap - Si se trata de usuarios Ldap o no.

setMiscDef

```
public void setMiscDef(ArchiveMisc miscArch)
```

Establece información referente a lista de volúmenes y título de carpeta del archivador.

Parametros:

Estructura - con la información mencionada.



setName

```
public void setName(java.lang.String name)
```

Establece el nombre del archivador.

Parametros:

name - El nombre mencionado.

setRemarks

```
public void setRemarks(java.lang.String remarks)
```

Establece la descripción del archivador.

Parametros:

remarks - La descripción mencionada.

toString

```
public java.lang.String toString()
```

Muestra una representación de los valores de la clase en formato XML.

Retorno:

La representación mencionada.

Reemplaza:

toString in class java.lang.Object

toXML

```
public java.lang.String toXML()
```

Obtiene la información del directorio en formato XML.

Retorno:

La información mencionada.

update

```
public void update(ArchiveUpdInfo updInfo)  
throws java.lang.Exception
```

Actualiza la información del archivador.

Parametros:

updInfo - Estructura con la totalidad de la información

Lanza:

java.lang.Exception - Si se produce algún error al actualizar



updateParentId

```
public void updateParentId(int parentId,
                             int archiveId)
    throws java.lang.Exception
```

Actualiza el padre para el archivador indicado no hace falta que se realice una carga del archivador.

Parametros:

parentId - Identificador del padre
archiveId - Identificador del archivador

Lanza:

java.lang.Exception - Si se produce algún error al actualizar

ieci.tecdoc.idoc.admin.api.archive

Interface ArchiveAccess

< [Metodos](#) >

```
public interface ArchiveAccess
```

Maneja los accesos a la gestión de directorios y archivadores.

Metodos

userCanAddPermOnArch

```
public boolean userCanAddPermOnArch(int connectedUserId,
                                       int archId)
    throws java.lang.Exception
```

Obtiene si el usuario conectado puede asignar permisos sobre el archivador.

Parametros:

connectedUserId - Identificador del usuario conectado.
archId - Identificador del archivador.

Retorno:

true / false

Lanza:

java.lang.Exception - Errores



userCanCreateArch

```
public boolean userCanCreateArch(int connectedUserId,  
                                int dirId)  
    throws java.lang.Exception
```

Obtiene si el usuario conectado puede crear un archivador en el directorio especificado.

Parametros:

connectedUserId - Identificador del usuario conectado.
dirId - Identificador del directorio

Retorno:

true / false

Lanza:

java.lang.Exception - Errores

userCanCreateDir

```
public boolean userCanCreateDir(int connectedUserId,  
                                int dirId)  
    throws java.lang.Exception
```

Obtiene si el usuario conectado puede crear el directorio.

Parametros:

connectedUserId - Identificador del usuario conectado.
dirId - Identificador del directorio.

Retorno:

true / false

Lanza:

java.lang.Exception - Errores

userCanDeleteArch

```
public boolean userCanDeleteArch(int connectedUserId,  
                                 int archId)  
    throws java.lang.Exception
```

Obtiene si el usuario conectado puede eliminar el archivador.

Parametros:

connectedUserId - Identificador del usuario conectado.
archId - Identificador del archivador.

Retorno:

true / false

Lanza:

java.lang.Exception - Errores



userCanDeleteDir

```
public boolean userCanDeleteDir(int connectedUserId,
                                int dirId)
    throws java.lang.Exception
```

Obtiene si el usuario conectado puede eliminar el directorio.

Parametros:

connectedUserId - Identificador del usuario conectado.
dirId - Identificador del directorio.

Retorno:

true / false

Lanza:

java.lang.Exception - Errores

userCanEditArch

```
public boolean userCanEditArch(int connectedUserId,
                                int archId)
    throws java.lang.Exception
```

Obtiene si el usuario conectado puede editar el archivador.

Parametros:

connectedUserId - Identificador del usuario conectado.
archId - Identificador del archivador.

Retorno:

true / false

Lanza:

java.lang.Exception - Errores

userCanEditDir

```
public boolean userCanEditDir(int connectedUserId,
                                int dirId)
    throws java.lang.Exception
```

Obtiene si el usuario conectado puede editar el directorio.

Parametros:

connectedUserId - Identificador del usuario conectado.
dirId - Identificador del directorio.

Retorno:

true / false

Lanza:

java.lang.Exception - Errores



userCanMoveArch

```
public boolean userCanMoveArch(int connectedUserId,
                               int archId,
                               int dstDirId)
    throws java.lang.Exception
```

Obtiene si el usuario conectado puede mover el archivador al destino especificado.

Parametros:

connectedUserId - Identificador del usuario conectado.
archId - Identificador del archivador.
dstDirId - Identificador del directorio destino.

Retorno:

true / false

Lanza:

java.lang.Exception - Errores

userCanMoveDir

```
public boolean userCanMoveDir(int connectedUserId,
                               int dirId,
                               int dstDirId)
    throws java.lang.Exception
```

Obtiene si el usuario conectado puede mover el directorio.

Parametros:

connectedUserId - Identificador del usuario conectado.
dirId - Identificador del directorio.
dstDirId - Identificador del directorio destino.

Retorno:

true / false

Lanza:

java.lang.Exception - Errores

userCanViewArch

```
public boolean userCanViewArch(int connectedUserId,
                               int archId)
    throws java.lang.Exception
```

Obtiene si el usuario conectado puede ver el archivador.

Parametros:

connectedUserId - Identificador del usuario conectado.
archId - Identificador del archivador.

Retorno:

true / false

Lanza:

java.lang.Exception - Errores



userCanViewDir

```
public boolean userCanViewDir(int connectedUserId,
                             int dirId)
    throws java.lang.Exception
```

Obtiene si el usuario conectado puede ver el directorio.

Parámetros:

connectedUserId - Identificador del usuario conectado.
dirId - Identificador del directorio.

Retorno:

true / false

Lanza:

java.lang.Exception - Errores

ieci.tecdoc.idoc.admin.api.archive

Class ArchiveDefs

```
java.lang.Object
|
+--ieci.tecdoc.idoc.admin.api.archive.ArchiveDefs
```

< [Campos](#) >

```
public class ArchiveDefs
    extends java.lang.Object
```

Proporciona las definiciones de todos los identificadores relativos a la administración de directorios y archivadores.

Campos

ACCESS_TYPE_PROTECTED

```
public static int ACCESS_TYPE_PROTECTED
    El acceso al elemento es protegido.
```

ACCESS_TYPE_PUBLIC

```
public static int ACCESS_TYPE_PUBLIC
    El acceso al elemento es público.
```

ARCH_DET_CLF_DEF

```
public static int ARCH_DET_CLF_DEF
    Identificador de Detalle de clasificadores en archivador
```



ARCH_DET_FLD_DEF

```
public static int ARCH_DET_FLD_DEF
    Identificador de Detalle de campos e índices en archivador
```

ARCH_DET_MISC_DEF

```
public static int ARCH_DET_MISC_DEF
    Identificador de Detalle de miscelánea en archivador
```

ARCH_DET_VLD_DEF

```
public static int ARCH_DET_VLD_DEF
    Identificador de Detalle de validaciones en archivador
```

ARCH_FLAG_FTSINCONTENTS

```
public static int ARCH_FLAG_FTSINCONTENTS
    Identificador de flag de archivador Búsqueda en contenido de fichero
```

ARCH_FLAG_NONE

```
public static int ARCH_FLAG_NONE
    Identificador de flag de archivador ninguno
```

ARCH_TYPE_SICRES_INPUT

```
public static int ARCH_TYPE_SICRES_INPUT
    Identificador de Archivador invesSicres entrada
```

ARCH_TYPE_SICRES_OUTPUT

```
public static int ARCH_TYPE_SICRES_OUTPUT
    Identificador de Archivador invesSicres salida
```

ARCH_TYPE_STANDARD

```
public static int ARCH_TYPE_STANDARD
    Identificador de Archivador Estándar
```

DAT_NODE_TYPE_ARCHIVE

```
public static int DAT_NODE_TYPE_ARCHIVE
    El nodo corresponde a un archivador.
```



DAT_NODE_TYPE_DIRECTORY

```
public static int DAT_NODE_TYPE_DIRECTORY
```

El nodo corresponde a un directorio.

DIR_TYPE_SICRES

```
public static int DIR_TYPE_SICRES
```

Identificador de *Directorio inveSicres*.

DIR_TYPE_STANDARD

```
public static int DIR_TYPE_STANDARD
```

Identificador de *Directorio Estándar*.

ieci.tecdoc.idoc.admin.api.archive

Interface ArchiveFld

< [Metodos](#) >

```
public interface ArchiveFld
```

Proporciona toda la funcionalidad necesaria para manejar un campo del archivador.

Metodos

getColName

```
public java.lang.String getColName()
```

Obtiene el nombre de la columna en base de datos referente al campo

Retorno:

nombre de columna

getId

```
public int getId()
```

Obtiene el identificador del campo

Retorno:

dicho identificador



getLen

```
public int getLen()
```

Obtiene la longitud del campo.

Retorno:

Longitud mencionada.

getName

```
public java.lang.String getName()
```

Obtiene el nombre del campo

Retorno:

dicho nombre

getRemarks

```
public java.lang.String getRemarks()
```

Obtiene la descripción

Retorno:

descripción

getType

```
public int getType()
```

Obtiene el tipo de campo en base de datos

Retorno:

Tipo de campo.

isDoc

```
public boolean isDoc()
```

Obtiene si el campo tiene búsqueda documental.

Retorno:

true / false



isExt

```
public boolean isExt()
```

Obtiene si el campo es extendido.

Retorno:

true / false

isMult

```
public boolean isMult()
```

obtiene si el campo es multivalor.

Retorno:

true / false

isNullable

```
public boolean isNullable()
```

Obtiene la nulidad del campo.

Retorno:

true / false

isRel

```
public boolean isRel()
```

Obtiene si el campo es relacional.

Retorno:

true / false

setDoc

```
public void setDoc(boolean isDoc)
```

Establece o no la búsqueda documental de un campo.

Parametros:

isDoc - true / false



setLen

```
public void setLen(int len)
```

Establece la longitud de un campo.

Parametros:

len - longitud

setMult

```
public void setMult(boolean mult)
```

Establece si un campo es multivalor o no

Parametros:

mult - true / false

setName

```
public void setName(java.lang.String name)
```

Establece el nombre del campo

Parametros:

name - nombre del campo

setNullable

```
public void setNullable(boolean nullable)
```

Establece si un campo es obligatorio o no

Parametros:

nullable - true / false

setRemarks

```
public void setRemarks(java.lang.String remarks)
```

Establece la descripción del campo

Parametros:

remarks - descripción



setType

```
public void setType(int type)
```

Establece el tipo de campo.

Parámetros:

type - tipo de campo

toString

```
public java.lang.String toString()
```

Muestra una representación de los valores de la clase en formato XML.

Retorno:

La representación mencionada.

Reemplaza:

toString in class java.lang.Object

toXML

```
public java.lang.String toXML()
```

Obtiene la información del campo en formato XML.

Retorno:

La información mencionada.

ieci.tecdoc.idoc.admin.api.archive

Interface ArchiveFlds

< [Metodos](#) >

```
public interface ArchiveFlds
```

Metodos



add

```
public void add(java.lang.String name,
               int type,
               int len,
               boolean isNullable,
               boolean isDoc,
               boolean isMult,
               java.lang.String remarks)
    throws java.lang.Exception
```

Añade una definición de campo. (el cálculo del identificador es interno, nunca tiene que se establecido)

Parámetros:

name - Nombre del campo
type - Tipo del campo (ArchiveFldType)
len - Longitud del campo en caso de texto y texto largo en otro caso es = 0
isNullable - true /false (posibilidad de tener valores nulos)
isDoc - true / false (documental)
isMult - true /false (multivalor)
remarks - Descripción

Lanza:

java.lang.Exception - - Errores

addFld

```
public void addFld(ArchiveFld fld)
```

Añade una definición de campo.

Parámetros:

fld - Definición del campo.

clear

```
public void clear()
```

Inicializa la estructura de campos

count

```
public int count()
```

Obtiene el número de campos.

Retorno:

Número mencionado



get

```
public ArchiveFld get(int index)
```

Obtiene la definición de un campo a partir de un índice de la colección.

Parámetros:

index - Índice

Retorno:

La estructura mencionada.

getFldDefById

```
public ArchiveFld getFldDefById(int fldId) throws java.lang.Exception
```

Obtiene la definición de un campo a partir del identificador.

Parámetros:

fldId - Identificador del campo

Retorno:

La definición mencionada.

Lanza:

java.lang.Exception - Si el campo no existe

getFldIdByName

```
public int getFldIdByName(java.lang.String name) throws java.lang.Exception
```

Obtiene el identificador del campo a partir del nombre.

Parámetros:

name - Nombre del campo

Retorno:

Identificador del campo

Lanza:

java.lang.Exception - Si no existe el campo



getFtsFlds

```
public ieci.tecdoc.idoc.admin.internal.ArchiveFldsImpl getFtsFlds()  
                                                    throws java.lang.Exception
```

Obtiene una colección con las definiciones de los campos documentales.

Retorno:

Colección de campos documentales

Lanza:

java.lang.Exception - Errores

getFtsFldsCount

```
public int getFtsFldsCount()
```

Obtiene el número de campos documentales en la colección.

Retorno:

Número de campos documentales

getMultFldsCount

```
public int getMultFldsCount()
```

Obtiene el número de campos multivalores en la colección.

Retorno:

Número de campos multivalores

getRelFldsCount

```
public int getRelFldsCount()
```

Obtiene el número de campos relacionales en la colección.

Retorno:

Número de campos relacionales

toString

```
public java.lang.String toString()
```

Muestra una representación de los valores de la clase en formato XML.

Retorno:

La representación mencionada.

Reemplaza:

toString in class java.lang.Object



toXML

```
public java.lang.String toXML()
```

Obtiene la información de los campos en formato XML.

Retorno:

La información mencionada.

ieci.tecdoc.idoc.admin.api.archive

Interface ArchivelDx

< [Metodos](#) >

```
public interface ArchivelDx
```

Metodos

getFldsId

```
public java.util.ArrayList getFldsId()
```

Obtiene una lista de enteros (Integer) con los identificadores de los campos que componen el índice

Retorno:

Lista de identificadores de campos

getId

```
public int getId()
```

Obtiene el identificador del índice

Retorno:

El identificador

getName

```
public java.lang.String getName()
```

Obtiene el nombre del índice

Retorno:

El nombre



isUnique

```
public boolean isUnique()
```

Obtiene si el índice es único ó no

Retorno:

(true/false)

setFldsId

```
public void setFldsId(java.util.ArrayList fldsId)
```

Establece una lista de enteros (Integer) con los identificadores de los campos que componene el índice

Parametros:

idsFld - Lista de identificadores de campos

toString

```
public java.lang.String toString()
```

Muestra una representación de los valores de la clase en formato XML.

Retorno:

La representación mencionada.

Reemplaza:

toString in class java.lang.Object

toXML

```
public java.lang.String toXML()
```

Obtiene la información del índice en formato XML.

Retorno:

La información mencionada.

ieci.tecdoc.idoc.admin.api.archive

Interface ArchivelDxs

< [Metodos](#) >

```
public interface ArchivelDxs
```



Metodos

add

```
public void add(java.lang.String name,  
               boolean isUnique,  
               java.util.ArrayList idsFld)
```

Añade la definición de un índice.

Parametros:

name - Nombre del índice

isUnique - true/false

idsFld - Lista con los identificadores de los campos que componen el índice

addIdx

```
public void addIdx(ArchiveIdx item)
```

Añade la definición de un índice.

Parametros:

item - Definición del índice

clear

```
public void clear()
```

Inicializa la estructura de índices

count

```
public int count()
```

Obtiene el número de índices en la estructura.

Retorno:

Número de índices

get

```
public ArchiveIdx get(int index)
```

Obtiene la definición del índice.

Parametros:

index - Índice de la estructura

Retorno:

Errores



getIdxIdByName

```
public int getIdxIdByName(java.lang.String name)  
    throws java.lang.Exception
```

Obtiene el identificador del índice por su nombre.

Parametros:

name - Nombre del índice

Retorno:

Identificador del índice

Lanza:

java.lang.Exception - Errores

toString

```
public java.lang.String toString()
```

Muestra una representación de los valores de la clase en formato XML.

Retorno:

La representación mencionada.

Reemplaza:

toString in class java.lang.Object

toXML

```
public java.lang.String toXML()
```

Obtiene la información de los índices en formato XML.

Retorno:

La información mencionada.

ieci.tecdoc.idoc.admin.api.archive

Interface ArchiveMisc

< [Metodos](#) >

```
public interface ArchiveMisc
```

Metodos



getFdrName

```
public java.lang.String getFdrName()
```

Obtiene el título de las carpetas.

Retorno:

Título mencionado

getVolListId

```
public int getVolListId()
```

Obtiene el identificador de la lista de volúmenes.

Retorno:

Identificador mencionado

getVolListType

```
public int getVolListType()
```

Obtiene el tipo de lista de volúmenes.

Retorno:

Tipo uno de: ArchiveVolListType.STATIC ArchiveVolListType.NONE

setFdrName

```
public void setFdrName(java.lang.String fdrName)
```

Establece el título de las carpetas

Parametros:

fdrName - Título mencionado

setMisc

```
public void setMisc(java.lang.String fdrName,  
                    int volListId,  
                    int volListType)
```

Establece los datos referenciados en los parámetros.

Parametros:

fdrName - Título de las carpetas
volListId - Lista de volúmenes
volListType - Tipo de lista de volúmenes



setVolListId

```
public void setVolListId(int volListId)
```

Establece el identificador de la lista de volúmenes.

Parametros:

volListId - Identificador de la lista de volúmenes

setVolListType

```
public void setVolListType(int volListType)
```

Establece el tipo de lista de volúmenes.

Parametros:

volListType - Tipo (Ej: ArchiveVolListType.STATIC)

toString

```
public java.lang.String toString()
```

Muestra una representación de los valores de la clase en formato XML.

Retorno:

La representación mencionada.

Reemplaza:

toString in class java.lang.Object

toXML

```
public java.lang.String toXML()
```

Obtiene la información de miscelánea en formato XML.

Retorno:

La información mencionada.

ieci.tecdoc.idoc.admin.api.archive

Interface ArchivePerms

< [Metodos](#) >

```
public interface ArchivePerms
```

Proporciona toda la funcionalidad necesaria para manejar permisos sobre archivadores invesDoc.



Metodos

addPerms

```
public void addPerms(int destType,
                    int destId,
                    int acsId,
                    int perms)
    throws java.lang.Exception
```

Añade los permisos especificados

Parametros:

destType - Destinatario del permiso (departamento, usuario, grupo) (ej: Defs.DESTINATION_USER)
destId - Identificador del destinatario
acsId - Identificador de objeto sobre el que se añade el permiso (ej: AcsId del archivador)
perms - Permisos (ej: Defs.OBJ_PERM_CREATION | Defs.OBJ_PERM_UPDATE)

Lanza:

java.lang.Exception - Errores

deletePerms

```
public void deletePerms(int destType,
                       int destId,
                       int archId,
                       int perms)
    throws java.lang.Exception
```

Elimina los permisos especificador

Parametros:

destType - Destinatario del permiso (departamento, usuario, grupo) (ej: Defs.DESTINATION_USER)
destId - Identificador del destinatario
archId - Identificador del archivador
perms - Permisos (ej: Defs.OBJ_PERM_CREATION | Defs.OBJ_PERM_UPDATE)

Lanza:

java.lang.Exception - Errores



loadPerms

```
public int loadPerms(int destType,
                    int destId,
                    int archIdd)
    throws java.lang.Exception
```

Obtiene los permisos.

Parámetros:

destType - Destinatario del permiso (departamento, usuario, grupo) (ej: Defs.DESTINATION_USER)
 destId - Identificador del destinatario
 archId - Identificador del archivador

Retorno:

Permisos (uno ó varios de: Defs.OBJ_PERM_CREATION Defs.OBJ_PERM_DELETION
 Defs.OBJ_PERM_UPDATE Defs.OBJ_PERM_QUERY)

Lanza:

java.lang.Exception - Errores

ieci.tecdoc.idoc.admin.api.archive

Interface ArchiveUpdInfo

< [Metodos](#) >

```
public interface ArchiveUpdInfo
```

Metodos

ModifyDefFlds

```
public void ModifyDefFlds(boolean modify)
```

Establece si se ha modificado ó no la definición de los campos referente a tipo, longitud, documental, obligatorio ó multivalor, si alguno de estos valores se ha modificado el parámetro debería ser true.

Parámetros:

modify - - true / false

ModifyListVols

```
public void ModifyListVols(boolean modify)
```

Establece si se ha modificado ó no la lista de volúmenes asociada al archivador.

Parámetros:

modify - true / false



getAdminUserId

```
public int getAdminUserId()
```

Obtiene el administrador del archivador.

Retorno:

Identificador

getDeleteFlds

```
public java.util.ArrayList getDeleteFlds()
```

Obtiene los identificadores de los campos eliminados.

Retorno:

Lista con los identificadores

getDeleteIdxs

```
public java.util.ArrayList getDeleteIdxs()
```

Obtiene los identificadores de los índices eliminados.

Retorno:

Lita con los identificadores

getFldsDef

```
public ArchiveFlds getFldsDef()
```

Obtiene la estructura de la definición de los campos del archivador.

Retorno:

Estructura de campos

getIdxsDef

```
public ArchiveIdxs getIdxsDef()
```

Obtiene la estructura de la definición de índices.

Retorno:

Estructura de índices



getMiscDef

```
public ArchiveMisc getMiscDef()
```

Obtiene la información del título de carpetas y lista de volúmenes asociada.

Retorno:

Información

getName

```
public java.lang.String getName()
```

Obtiene el nombre del archivador.

Retorno:

Nombre

getNewFlds

```
public java.util.ArrayList getNewFlds()
```

Obtiene los identificadores de los campos añadidos.

Retorno:

Lista con los identificadores

getNewIdx

```
public java.util.ArrayList getNewIdx()
```

Obtiene los identificadores de los índices añadidos.

Retorno:

Lista con los identificadores

getRemarks

```
public java.lang.String getRemarks()
```

Obtiene la descripción del archivador.

Retorno:

Descripción



isFtsInContents

```
public boolean isFtsInContents()
```

Obtiene si hay búsqueda en contenido de fichero en el archivador.

Retorno:

true / false

isModifyDefFlds

```
public boolean isModifyDefFlds()
```

Obtiene si se ha modificado la definición de los campos.

Retorno:

true / false

isModifyListVols

```
public boolean isModifyListVols()
```

Obtiene si la lista de volúmenes ha sido modificada.

Retorno:

true / false

setAdminUserId

```
public void setAdminUserId(int adminUserId)
```

Establece el administrador del archivador.

Parametros:

adminUserId - Identificador

setFldsDef

```
public void setFldsDef(ArchiveFlds fldsDef)
```

Establece la estructura de la definición de campos debe contener la información de los antiguos y los nuevos.

Parametros:

fldsDef - Estructura de campos.



setFtsInContents

```
public void setFtsInContents(boolean ftsInContents)
```

Establece la existencia o no de búsqueda en contenido de fichero en el archivador.

Parametros:

ftsInContents - true / false

setIdxsDef

```
public void setIdxsDef(ArchiveIdxs idxsDef)
```

Establece la estructura de la definición de índices del archivador, debe contener la información de los antiguos y de los nuevos.

Parametros:

idxsDef - Estructura de índices

setMiscDef

```
public void setMiscDef(ArchiveMisc miscDef)
```

Establece la información del título de carpetas y lista de volúmenes asociada, si no se cambia debe contener la información antigua.

Parametros:

miscDef - Información mencionada

setName

```
public void setName(java.lang.String name)
```

Establece el nombre del archivador.

Parametros:

name - Nombre

setRemarks

```
public void setRemarks(java.lang.String remarks)
```

Establece la descripción del archivador.

Parametros:

remarks - Descripción



setUpdateFlds

```
public void setUpdateFlds(java.util.ArrayList newFldsId,
                          java.util.ArrayList delFldsId,
                          boolean isModifyDefFlds)
```

Establece un array de campos creador, otro de campos eliminados, Si se ha modificado la definición de campos referente a tipo, longitud, documental, obligatorio ó multivalor el parámetro isModifyDefFlds debería ser true.

Parametros:

newFldsId - Lista con identificadores de campos nuevos
 delFldsId - Lista con identificadores de campos eliminados
 isModifyDefFlds - true / false

setUpdateIdxs

```
public void setUpdateIdxs(java.util.ArrayList newIdxsId,
                          java.util.ArrayList delIdxsId)
```

Establece un array de índices creados, otro de índices eliminados.

Parametros:

newIdxsId - Lista con los identificadores de los índices nuevos.
 delIdxsId - Lista con los identificadores de los índices eliminados.

ieci.tecdoc.idoc.admin.api.archive

Interface Archives

< [Metodos](#) >

public interface **Archives**

Proporciona toda la funcionalidad necesaria para manejar archivadores invesDoc.

Metodos

getArchivesByDirectory

```
public BasicArchives getArchivesByDirectory(int parentDirId)
                                             throws java.lang.Exception
```

Obtiene información básica de archivadores de un mismo padre

Parametros:

parentDirId - Identificador del padre

Retorno:

La información mencionada

Lanza:

java.lang.Exception - Errores



ieci.tecdoc.idoc.admin.api.archive

Interface BasicArchive

< [Metodos](#) >

public interface **BasicArchive**

Metodos

getId

public int **getId**()

Recupera el identificador del archivador.

Retorno:

El identificador mencionado.

getName

public java.lang.String **getName**()

Recupera el nombre del archivador.

Retorno:

El nombre mencionado.

ieci.tecdoc.idoc.admin.api.archive

Interface BasicArchives

< [Metodos](#) >

public interface **BasicArchives**

Metodos



count

```
public int count()
```

Devuelve el número de archivadores.

Retorno:

El número de archivadores mencionado.

get

```
public BasicArchive get(int index)
```

Devuelve un archivador de la lista.

Parametros:

index - Índice del archivador que se desea recuperar.

Retorno:

El archivador mencionado.

ieci.tecdoc.idoc.admin.api.archive

Interface BasicDirectories

< [Metodos](#) >

```
public interface BasicDirectories
```

Metodos

count

```
public int count()
```

Devuelve el número de directorios.

Retorno:

El número de directorios mencionado.



get

```
public BasicDirectory get(int index)
```

Devuelve un directorio de la lista.

Parametros:

index - Índice del directorio que se desea recuperar.

Retorno:

El directorio mencionado.

ieci.tecdoc.idoc.admin.api.archive

Interface BasicDirectory

< [Metodos](#) >

```
public interface BasicDirectory
```

Metodos

getId

```
public int getId()
```

Recupera el identificador del directorio.

Retorno:

El identificador mencionado.

getName

```
public java.lang.String getName()
```

Recupera el nombre del directorio.

Retorno:

El nombre mencionado.

ieci.tecdoc.idoc.admin.api.archive

Interface Directories

< [Metodos](#) >

```
public interface Directories
```



Proporciona toda la funcionalidad necesaria para manejar árbol de directorios invesDoc.

Metodos

getChildrenFormDirectory

```
public BasicDirectories getChildrenFormDirectory(int dirId)
                                     throws java.lang.Exception
```

Obtiene información básica de directorios de un mismo padre

Parametros:

parentDirId - Identificador del padre

Retorno:

La información mencionada

Lanza:

java.lang.Exception - Errores

ieci.tecdoc.idoc.admin.api.archive

Interface Directory

< [Metodos](#) >

```
public interface Directory
```

Proporciona toda la funcionalidad necesaria para manejar directorios invesDoc.

Metodos

delete

```
public void delete()
           throws java.lang.Exception
```

Elimina el directorio.

Lanza:

java.lang.Exception - Si se produce algún error al eliminar. Por ejemplo, el directorio tiene hijos.

getAccessType

```
public int getAccessType()
```

Obtiene el tipo de acceso del directorio.

Retorno:

El tipo de acceso mencionado.



getAcsId

```
public int getAcsId()
```

Obtiene el identificador de acceso del directorio.

Retorno:

El identificador mencionado.

getAdminUserId

```
public int getAdminUserId()
```

Obtiene el identificador del usuario administrador del directorio.

Retorno:

El identificador mencionado.

getAdminUsers

```
public BasicUsers getAdminUsers()
```

Obtiene la lista de usuarios (identificadores y nombres) que pueden ser administradores de este directorio.

Retorno:

La lista mencionada.

getCreationDate

```
public java.util.Date getCreationDate()
```

Obtiene la fecha de creación del directorio.

Retorno:

La fecha mencionada.

getCreatorId

```
public int getCreatorId()
```

Obtiene el identificador del usuario que ha creado del directorio.

Retorno:

El identificador mencionado.



getDescription

```
public java.lang.String getDescription()
```

Obtiene la descripción del directorio.

Retorno:

La descripción mencionada.

getFlags

```
public int getFlags()
```

Obtiene los flags del directorio.

Retorno:

Los flags mencionados.

getName

```
public java.lang.String getName()
```

Obtiene el nombre del directorio.

Retorno:

El nombre mencionado.

getUpdateDate

```
public java.util.Date getUpdateDate()
```

Obtiene la fecha de actualización del directorio.

Retorno:

La fecha mencionada.

getUpdaterId

```
public int getUpdaterId()
```

Obtiene el identificador del usuario que ha actualizado el directorio.

Retorno:

El identificador mencionado.



load

```
public void load(int id)
    throws java.lang.Exception
```

Carga un directorio.

Parametros:

id - Identificador del directorio.

Lanza:

java.lang.Exception - Si se produce algún error al leer la información del directorio.

setAccessType

```
public void setAccessType(int accessType)
```

Establece el tipo de acceso del directorio.

Parametros:

accessType - El tipo de acceso.

setAdminUserId

```
public void setAdminUserId(int id)
```

Establece el identificador del usuario administrador del directorio.

Parametros:

id - El identificador mencionado.

setDescription

```
public void setDescription(java.lang.String description)
```

Establece la descripción del directorio.

Parametros:

description - La descripción del directorio.

setFlags

```
public void setFlags(int flags)
```

Establece los flags del directorio.

Retorno:

Los flags mencionados.



setLdap

```
public void setLdap(boolean ldap)
```

Indica si se está trabajando con usuarios de un repositorio Ldap (true) o no (false).

Parametros:

ldap - Si se trata de usuarios Ldap o no.

setName

```
public void setName(java.lang.String name)
```

Establece el nombre del directorio.

Parametros:

name - El nombre del directorio.

store

```
public void store()  
    throws java.lang.Exception
```

Guarda el directorio. Se utiliza tanto para inserciones como para actualizaciones.

Lanza:

java.lang.Exception - Si se produce algún error al guardar. Por ejemplo, el directorio ya existe.

toString

```
public java.lang.String toString()
```

Muestra una representación de los valores de la clase en formato XML.

Retorno:

La representación mencionada.

Reemplaza:

toString in class java.lang.Object

toXML

```
public java.lang.String toXML()
```

Obtiene la información del directorio en formato XML.

Retorno:

La información mencionada.



Paquete `ieci.tecdoc.idoc.admin.api.exception`

Clases

[AdminException](#)

[AdminResourceFile](#)

[ArchiveErrorCodes](#)

Proporciona los códigos de error específicos del API de administración de archivadores.

[DeptErrorCodes](#)

Proporciona los códigos de error específicos del API de administración de departamentos.

[DirErrorCodes](#)

Proporciona los códigos de error específicos del API de administración de directorios.

[GroupErrorCodes](#)

Proporciona los códigos de error específicos del API de administración de grupos.

[LoginErrorCodes](#)

Proporciona los códigos de error específicos del API de acceso(login) al sistema.

[RepositoryErrorCodes](#)

Proporciona los códigos de error específicos del API de administración de repositorios de volúmenes.

[UserErrorCodes](#)

Proporciona los códigos de error específicos del API de administración de usuarios.

[VolumeErrorCodes](#)

Proporciona los códigos de error específicos del API de administración de volúmenes.

[VolumeListErrorCodes](#)

Proporciona los códigos de error específicos del API de administración de lista de volúmenes.

`ieci.tecdoc.idoc.admin.api.exception`

Class AdminException

```
java.lang.Object
|
+--ieci.tecdoc.idoc.admin.api.exception.AdminException
```

< [Metodos](#) >

```
public class AdminException
extends java.lang.Object
```



Metodos

throwException

```
public static void throwException(long errorCode)
    throws ieci.tecdoc.core.exception.IeciTdException
```

Lanza una excepción específica del sistema llamante.

Parametros:

errorCode - Código de error de la excepción.

Lanza:

ieci.tecdoc.core.exception.IeciTdException - Excepción generada.

ieci.tecdoc.idoc.admin.api.exception

Class AdminResourceFile

```
java.lang.Object
|
+--ieci.tecdoc.idoc.admin.api.exception.AdminResourceFile
```

< [Campos](#) >

```
public final class AdminResourceFile
extends java.lang.Object
```

Campos

ADMIN_RESOURCE_FILE

```
public static final java.lang.String ADMIN_RESOURCE_FILE
```

ieci.tecdoc.idoc.admin.api.exception

Class ArchiveErrorCodes

```
java.lang.Object
|
+--ieci.tecdoc.idoc.admin.api.exception.ArchiveErrorCodes
```

< [Campos](#) >

```
public class ArchiveErrorCodes
extends java.lang.Object
```

Proporciona los códigos de error específicos del API de administración de archivadores.



Campos

EC_ARCH_CANNT_PERM

```
public static final long EC_ARCH_CANNT_PERM
```

El permiso no puede ser asignado sobre el archivador, asegúrese que el receptor se le puede asignar el permiso

EC_ARCH_EXISTS_FLDS_WITH_VALIDATION

```
public static final long EC_ARCH_EXISTS_FLDS_WITH_VALIDATION
```

Existen campo que han sido modificados y tiene validación

EC_ARCH_EXITS

```
public static final long EC_ARCH_EXITS
```

El archivador ya existe.

EC_ARCH_FLD_TXT_BAD_LEN

```
public static final long EC_ARCH_FLD_TXT_BAD_LEN
```

La longitud de algún campo de tipo texto no es válida

EC_ARCH_FTS_BAD_FIELDS

```
public static final long EC_ARCH_FTS_BAD_FIELDS
```

Se han definido campos documentales sobre tipo de campos no válidos

EC_ARCH_HAS_FOLDERS

```
public static final long EC_ARCH_HAS_FOLDERS
```

El archivador tiene carpetas

EC_ARCH_HAS_VALIDATION_TABLE

```
public static final long EC_ARCH_HAS_VALIDATION_TABLE
```

El archivador tiene tabla de validación sobre alguno de sus campos

EC_ARCH_IDXS_BAD_FIELDS

```
public static final long EC_ARCH_IDXS_BAD_FIELDS
```

Existen índices que tienen asociados campos extendidos ó campo multivalores, no está permitido

EC_ARCH_IDXS_DEL_FIELDS



```
public static final long EC_ARCH_IDXS_DEL_FIELDS
    Se han definido índices sobre campos no existentes
```

EC_ARCH_NOT_EXISTS

```
public static final long EC_ARCH_NOT_EXISTS
    El Archivador no existe
```

EC_ARCH_NO_FLDS

```
public static final long EC_ARCH_NO_FLDS
    El archivador no tiene campos
```

EC_ARCH_NO_ID

```
public static final long EC_ARCH_NO_ID
    El identificar del archivador no es válido
```

EC_ARCH_NO_NAME

```
public static final long EC_ARCH_NO_NAME
    El archivador no tiene nombre
```

EC_ARCH_PARAM_NO_VALID

```
public static final long EC_ARCH_PARAM_NO_VALID
    Parámetro no válido
```

EC_ARCH_REMARKS_EXIST_QUOTES

```
public static final long EC_ARCH_REMARKS_EXIST_QUOTES
    Existe dobles comilla en el campo descripción del archivador ó de alguno de sus campos
```

EC_ARCH_WITH_FDRS

```
public static final long EC_ARCH_WITH_FDRS
    El archivador tiene carpetas, no se puede modificar la definición de sus campos.
```

EC_FLD_EXISTS

```
public static final long EC_FLD_EXISTS
    El campo ya existe
```

EC_FLD_NAME_NO_VALID



```
public static final long EC_FLD_NAME_NO_VALID
    El nombre del campo no es válido
```

EC_FLD_NO_EXISTS

```
public static final long EC_FLD_NO_EXISTS
    El campo no existe
```

EC_IDX_EXISTS

```
public static final long EC_IDX_EXISTS
    El índice ya existe
```

EC_IDX_NAME_NO_VALID

```
public static final long EC_IDX_NAME_NO_VALID
    El nombre del índice no es válido
```

EC_IDX_NO_EXISTS

```
public static final long EC_IDX_NO_EXISTS
    El índice no existe
```

EC_NO_CONFIG_FTS

```
public static final long EC_NO_CONFIG_FTS
    No está configurado el uso de búsqueda documental en el sistema
```

EC_PARENT_NO_ID

```
public static final long EC_PARENT_NO_ID
    El identificador del padre no es válido
```

ieci.tecdoc.idoc.admin.api.exception

Class DeptErrorCodes

```
java.lang.Object
|
+--ieci.tecdoc.idoc.admin.api.exception.DeptErrorCodes
```

< [Campos](#) >

```
public class DeptErrorCodes
    extends java.lang.Object
```

Proporciona los códigos de error específicos del API de administración de departamentos.



Campos

EC_DEPT_DESC_EXIST_QUOTES

```
public static final long EC_DEPT_DESC_EXIST_QUOTES
```

No se permite introducir comillas en el texto.

EC_DEPT_EXIST_NAME

```
public static final long EC_DEPT_EXIST_NAME
```

Ya existe un departamento con ese nombre.

EC_DEPT_NOT_EXITS

```
public static final long EC_DEPT_NOT_EXITS
```

El departamento no existe. Por ejemplo, cuando se intenta cargar un departamento cuyo identificador no es válido.

EC_DEPT_NOT_PERMS

```
public static final long EC_DEPT_NOT_PERMS
```

El departamento no tiene permisos asignados.

EC_DEPT_NOT_PERM_FOR_PRODUCT

```
public static final long EC_DEPT_NOT_PERM_FOR_PRODUCT
```

El permiso del departamento para el producto solicitado no existe.

ieci.tecdoc.idoc.admin.api.exception

Class DirErrorCodes

```
java.lang.Object
|
+--ieci.tecdoc.idoc.admin.api.exception.DirErrorCodes
```

< [Campos](#) >

```
public class DirErrorCodes
extends java.lang.Object
```

Proporciona los códigos de error específicos del API de administración de directorios.

Campos

EC_DIR_EXITS



```
public static final long EC_DIR_EXISTS
    El directorio ya existe.
```

EC_DIR_HAS_CHILDREN

```
public static final long EC_DIR_HAS_CHILDREN
    El directorio no se puede eliminar. Tiene hijos.
```

EC_DIR_NOT_EXISTS

```
public static final long EC_DIR_NOT_EXISTS
    El directorio no existe.
```

ieci.tecdoc.idoc.admin.api.exception

Class GroupErrorCodes

```
java.lang.Object
|
+--ieci.tecdoc.idoc.admin.api.exception.GroupErrorCodes
```

< [Campos](#) >

```
public class GroupErrorCodes
    extends java.lang.Object
```

Proporciona los códigos de error específicos del API de administración de grupos.

Campos

EC_GROUP_DESC_EXIST_QUOTES

```
public static final long EC_GROUP_DESC_EXIST_QUOTES
    No se permite introducir comillas en el texto.
```

EC_GROUP_EXIST_NAME

```
public static final long EC_GROUP_EXIST_NAME
    Ya existe un grupo con ese nombre.
```

EC_GROUP_EXISTS

```
public static final long EC_GROUP_EXISTS
    El grupo ya existe. Por ejemplo, cuando se crea un grupo a partir de un servicio de directorio Ldap que ha sido creado previamente.
```

EC_GROUP_NOT_EXISTS



```
public static final long EC_GROUP_NOT_EXISTS
    El grupo no existe. Por ejemplo, cuando se intenta cargar un grupo cuyo identificador no es válido.
```

EC_GROUP_NOT_PERMS

```
public static final long EC_GROUP_NOT_PERMS
    El grupo no tiene permisos asignados.
```

EC_GROUP_NOT_PERM_FOR_PRODUCT

```
public static final long EC_GROUP_NOT_PERM_FOR_PRODUCT
    El permiso del grupo para el producto solicitado no existe.
```

ieci.tecdoc.idoc.admin.api.exception

Class LoginErrorCodes

```
java.lang.Object
|
+--ieci.tecdoc.idoc.admin.api.exception.LoginErrorCodes
```

< [Campos](#) >

```
public class LoginErrorCodes
    extends java.lang.Object
```

Proporciona los códigos de error específicos del API de acceso(login) al sistema.

Campos

EC_INVALID_METHOD_ID

```
public static final long EC_INVALID_METHOD_ID
    El método de acceso (login) no es válido para el API de administración.
```

EC_INVALID_PRODUCT_ID

```
public static final long EC_INVALID_PRODUCT_ID
    El identificador de aplicación no es válido para el API de administración.
```

ieci.tecdoc.idoc.admin.api.exception

Class RepositoryErrorCodes

```
java.lang.Object
|
+--ieci.tecdoc.idoc.admin.api.exception.RepositoryErrorCodes
```




< [Campos](#) >

```
public class RepositoryErrorCodes
extends java.lang.Object
```

Proporciona los códigos de error específicos del API de administración de repositorios de volúmenes.

Campos

EC_PREFIX

```
public static final long EC_PREFIX
    Código de error base.
```

EC_REP_EXIST_FILE

```
public static final long EC_REP_EXIST_FILE
    Ya existe un repositorio con esa ruta de acceso.
```

EC_REP_EXIST_NAME

```
public static final long EC_REP_EXIST_NAME
    Ya existe un repositorio con ese nombre.
```

EC_REP_HAS_VOLUMES

```
public static final long EC_REP_HAS_VOLUMES
    No se puede eliminar el repositorio porque tiene volúmenes.
```

EC_REP_NOT_EXIST

```
public static final long EC_REP_NOT_EXIST
    No existe el repositorio que se quiere cargar.
```

EC_REP_NOT_EXIST_FILE

```
public static final long EC_REP_NOT_EXIST_FILE
    No se ha podido crear el fichero.
```

EC_REP_REMARKS_EXIST_QUOTES

```
public static final long EC_REP_REMARKS_EXIST_QUOTES
    No se permite introducir comillas en el texto.
```



ieci.tecdoc.idoc.admin.api.exception

Class UserErrorCodes

```
java.lang.Object
|
+--ieci.tecdoc.idoc.admin.api.exception.UserErrorCodes
```

< [Campos](#) >

```
public class UserErrorCodes
extends java.lang.Object
```

Proporciona los códigos de error específicos del API de administración de usuarios.

Campos

EC_USER_CONNECTED

```
public static final long EC_USER_CONNECTED
    El usuario no se puede eliminar. Se produce cuando el usuario a eliminar coincide con el que está conectado
    administrado el sistema.
```

EC_USER_DESC_EXIST_QUOTES

```
public static final long EC_USER_DESC_EXIST_QUOTES
    No se permite introducir comillas en el texto.
```

EC_USER_EXISTS_NAME

```
public static final long EC_USER_EXISTS_NAME
    Ya existe un usuario con ese nombre
```

EC_USER_EXITS

```
public static final long EC_USER_EXITS
    El usuario ya existe. Por ejemplo, cuando se crea un usuario a partir de un servicio de directorio Ldap que ha sido
    creado previamente.
```

EC_USER_NOT_CONFIG

```
public static final long EC_USER_NOT_CONFIG
    No existe la configuración del usuario.
```

EC_USER_NOT_EXITS

```
public static final long EC_USER_NOT_EXITS
    El usuario no existe. Por ejemplo, cuando se intenta cargar un usuario cuyo identificador no es válido.
```



EC_USER_NOT_PERMS

```
public static final long EC_USER_NOT_PERMS
    El usuario no tiene permisos asignados.
```

EC_USER_NOT_PERM_FOR_PRODUCT

```
public static final long EC_USER_NOT_PERM_FOR_PRODUCT
    El permiso del usuario para el producto solicitado no existe.
```

EC_USER_NOT_PROFILES

```
public static final long EC_USER_NOT_PROFILES
    El usuario no tiene perfiles asignados.
```

EC_USER_NOT_PROFILE_FOR_PRODUCT

```
public static final long EC_USER_NOT_PROFILE_FOR_PRODUCT
    El perfil del usuario para el producto solicitado no existe.
```

EC_USER_PASS_EMPTY

```
public static final long EC_USER_PASS_EMPTY
    La contraseña no puede ser vacía.
```

EC_USER_PASS_LENGTH

```
public static final long EC_USER_PASS_LENGTH
    El tamaño de la contraseña debe ser mayor.
```

EC_USER_PASS_WHITESPACE

```
public static final long EC_USER_PASS_WHITESPACE
    Contraseña no válida. No se admiten blancos.
```

ieci.tecdoc.idoc.admin.api.exception

Class VolumeErrorCodes

```
java.lang.Object
|
+--ieci.tecdoc.idoc.admin.api.exception.VolumeErrorCodes
```

< [Campos](#) > < [Constructores](#) >



```
public class VolumeErrorCodes
extends java.lang.Object
```

Proporciona los códigos de error específicos del API de administración de volúmenes.

Campos

EC_PREFIX

```
public static final long EC_PREFIX
    Código de error base.
```

EC_VOL_EXIST_FILE

```
public static final long EC_VOL_EXIST_FILE
    Ya existe un volumen con esa ruta de acceso.
```

EC_VOL_EXIST_NAME

```
public static final long EC_VOL_EXIST_NAME
    Ya existe un volumen con ese nombre.
```

EC_VOL_HAS_FILES

```
public static final long EC_VOL_HAS_FILES
    No se puede eliminar el volumen porque tiene ficheros.
```

EC_VOL_MAXSIZE_EXCEED

```
public static final long EC_VOL_MAXSIZE_EXCEED
    Debe cumplimentar 'Capacidad' con un valor superior a la ocupación actual del volumen.
```

EC_VOL_NOT_EXIST

```
public static final long EC_VOL_NOT_EXIST
    No existe el volumen que se quiere cargar.
```

EC_VOL_NOT_EXIST_FILE

```
public static final long EC_VOL_NOT_EXIST_FILE
    No se ha podido crear el fichero.
```

EC_VOL_REMARKS_EXIST_QUOTES

```
public static final long EC_VOL_REMARKS_EXIST_QUOTES
    No se permite introducir comillas en el texto.
```



Constructores

VolumeErrorCodes

```
public VolumeErrorCodes()
```

ieci.tecdoc.idoc.admin.api.exception

Class VolumeListErrorCodes

```
java.lang.Object
|
|--ieci.tecdoc.idoc.admin.api.exception.VolumeListErrorCodes
```

< [Campos](#) >

```
public class VolumeListErrorCodes
extends java.lang.Object
```

Proporciona los códigos de error específicos del API de administración de lista de volúmenes.

Campos

EC_LISTVOL_ASSOC_ARCH

```
public static final long EC_LISTVOL_ASSOC_ARCH
    No se puede borrar la lista, está asociada a un archivador.
```

EC_LISTVOL_EXIST_NAME

```
public static final long EC_LISTVOL_EXIST_NAME
    Ya existe una lista de volúmenes con ese nombre.
```

EC_LISTVOL_EXIST_VOL

```
public static final long EC_LISTVOL_EXIST_VOL
    Ya existe ese volumen en la lista.
```

EC_LISTVOL_NOT_EXIST

```
public static final long EC_LISTVOL_NOT_EXIST
    No existe la lista de volúmenes que se quiere cargar.
```

EC_LISTVOL_NOT_VOLS



```
public static final long EC_LISTVOL_NOT_VOLS
    La lista no tiene volúmenes.
```

EC_LISTVOL_REMARKS_EXIST_QUOTES

```
public static final long EC_LISTVOL_REMARKS_EXIST_QUOTES
    No se permite introducir comillas en el texto.
```

EC_PREFIX

```
public static final long EC_PREFIX
    Código de error base.
```



Paquete `ieci.tecdoc.idoc.admin.api.system`

Interfaces

[SystemCfg](#)

Proporciona toda la funcionalidad necesaria para configuración del sistema.

`ieci.tecdoc.idoc.admin.api.system`

Interface `SystemCfg`

< [Metodos](#) >

public interface **SystemCfg**

Proporciona toda la funcionalidad necesaria para configuración del sistema.

Metodos

`getRootNameArchs`

```
public java.lang.String getRootNameArchs()
    throws java.lang.Exception
```

Obtien el nombre raíz del árbol de archivadores

Retorno:

El nombre mencionado

Lanza:

`java.lang.Exception` -

`hasFtsConfig`

```
public boolean hasFtsConfig()
    throws java.lang.Exception
```

Obtiene si existe o no configuración de motor documental

Retorno:

true / false

Lanza:

`java.lang.Exception` -



Paquete `ieci.tecdoc.idoc.admin.api.user`

Interfaces

[BasicUser](#)

Maneja el identificador y el nombre de un usuario.

[BasicUsers](#)

[Department](#)

Proporciona toda la funcionalidad necesaria para manejar departamentos de usuarios de invesDoc.

[Group](#)

Proporciona toda la funcionalidad necesaria para manejar grupos invesDoc.

[GroupUserRel](#)

[LdapGroup](#)

Proporciona toda la funcionalidad necesaria para manejar grupos invesDoc relacionados con un servicio de directorio Ldap.

[LdapUser](#)

Proporciona toda la funcionalidad necesaria para manejar usuarios invesDoc relacionados con un servicio de directorio Ldap.

[Permission](#)

Maneja los permisos de un usuario, grupo o departamento.

[Permissions](#)

Mantiene el conjunto de permisos de un usuario, grupo o departamento.

[User](#)

Proporciona toda la funcionalidad necesaria para manejar usuarios invesDoc.

[UserAccess](#)

Maneja los accesos a la gestión de departamentos, grupos y usuarios.

[UserProfile](#)

Maneja los perfiles de un usuario.

[UserProfiles](#)

Mantiene el conjunto de perfiles de un usuario.

Clases

[Departments](#)

Maneja la lista de departamentos invesDoc.

[Groups](#)



Maneja la lista de grupos invesDoc.

[LdapGroups](#)

Maneja la lista de grupos invesDoc relacionados con un servicio de directorio Ldap.

[LdapUsers](#)

Maneja la lista de usuarios invesDoc relacionados con un servicio de directorio Ldap.

[Login](#)

Proporciona funcionalidad de acceso a la administración de invesDoc.

[UserDefs](#)

Proporciona las definiciones de todos los identificadores relativos a la administración de usuarios.

[Users](#)

Maneja la lista de usuarios invesDoc.

ieci.tecdoc.idoc.admin.api.user

Interface BasicUser

< [Metodos](#) >

public interface **BasicUser**

Maneja el identificador y el nombre de un usuario.

Metodos

getId

```
public int getId()
```

Recupera el identificador del usuario.

Retorno:

El identificador mencionado.

getName

```
public java.lang.String getName()
```

Recupera el nombre del usuario.

Retorno:

El nombre mencionado.



ieci.tecdoc.idoc.admin.api.user

Interface BasicUsers

< [Metodos](#) >

public interface **BasicUsers**

Metodos

count

public int **count**()

Devuelve el número de usuarios.

Retorno:

El número de usuarios mencionado.

get

public [BasicUser](#) **get**(int index)

Devuelve un usuario de la lista.

Parametros:

index - Indice del usuario que se desea recuperar.

Retorno:

El usuario mencionado.

ieci.tecdoc.idoc.admin.api.user

Interface Department

< [Metodos](#) >

public interface **Department**

Proporciona toda la funcionalidad necesaria para manejar departamentos de usuarios de invesDoc.

Metodos



delete

```
public void delete()  
    throws java.lang.Exception
```

Elimina el departamento.

Lanza:

java.lang.Exception - Si se produce algún error al eliminar.

getAdminUsers

```
public BasicUsers getAdminUsers()
```

Obtiene la lista de usuarios (identificadores y nombres) que pueden ser administradores de este departamento.

Retorno:

La lista mencionada.

getCreationDate

```
public java.util.Date getCreationDate()
```

Obtiene la fecha de creación del departamento.

Retorno:

La fecha mencionada.

getCreatorId

```
public int getCreatorId()
```

Obtiene el identificador del usuario que ha creado el departamento.

Retorno:

El identificador mencionado.

getDescription

```
public java.lang.String getDescription()
```

Obtiene la descripción del departamento.

Retorno:

El nombre mencionado.



getId

```
public int getId()
```

Obtiene el identificador de departamento.

Retorno:

El identificador mencionado.

getManagerId

```
public int getManagerId()
```

Obtiene el identificador del administrador del departamento.

Retorno:

El identificador mencionado.

getName

```
public java.lang.String getName()
```

Obtiene el nombre del departamento.

Retorno:

El nombre mencionado.

getParentId

```
public int getParentId()
```

Obtiene el identificador del padre del departamento.

Retorno:

El nombre mencionado.

getPermissions

```
public Permissions getPermissions()
```

Devuelve la lista de permisos del departamento.

Retorno:

La lista mencionada.



getType

```
public int getType()
```

Obtiene el identificador del tipo de departamento.

Retorno:

El identificador mencionado.

getUpdateDate

```
public java.util.Date getUpdateDate()
```

Obtiene la fecha de actualización del departamento.

Retorno:

La fecha mencionada.

getUpdaterId

```
public int getUpdaterId()
```

Obtiene el identificador del usuario que ha actualizado el departamento.

Retorno:

El identificador mencionado.

getUsersByDept

```
public Users getUsersByDept(int id)
```

Devuelve la lista de usuarios del departamento.

Parametros:

id - Identificador del departamento.

Retorno:

Los usuarios del departamento.

load

```
public void load(int deptId)  
    throws java.lang.Exception
```

Carga un departamento invesDoc.

Parametros:

deptId - Identificador del departamento.

Lanza:

java.lang.Exception - Si se produce algún error al leer la información del departamento.



load

```
public void load(java.lang.String deptName)
    throws java.lang.Exception
```

Carga un departamento invesDoc.

Parametros:

deptName - Nombre del departamento.

Lanza:

java.lang.Exception - Si se produce algún error al leer la información del departamento.

setDescription

```
public void setDescription(java.lang.String description)
```

Establece la descripción del departamento.

Parametros:

description - La descripción del grupo.

setManagerId

```
public void setManagerId(int managerId)
```

Establece el identificador del administrador del departamento.

Parametros:

managerId - El identificador del administrador.

setName

```
public void setName(java.lang.String name)
```

Establece el nombre del departamento.

Parametros:

name - Nombre del departamento.

setParentId

```
public void setParentId(int parentId)
```

Establece el identificador del padre del departamento.

Parametros:

parentId - El identificador del padre del departamento.



store

```
public void store()
    throws java.lang.Exception
```

Guarda el departamento. Se utiliza tanto para inserciones como para actualizaciones.

Lanza:

java.lang.Exception - Si se produce algún error al guardar. Por ejemplo, el departamento ya existe.

toString

```
public java.lang.String toString()
```

Muestra una representación de los valores de la clase en formato XML.

Retorno:

La representación mencionada.

Reemplaza:

toString in class java.lang.Object

toXML

```
public java.lang.String toXML()
```

Obtiene la información del departamento en formato XML.

Retorno:

La información mencionada.

ieci.tecdoc.idoc.admin.api.user

Class Departments

```
java.lang.Object
|
+--ieci.tecdoc.idoc.admin.api.user.Departments
```

< [Constructores](#) > < [Metodos](#) >

```
public class Departments
    extends java.lang.Object
```

Maneja la lista de departamentos invesDoc.

Constructores



Departments

```
public Departments()
```

Construye un objeto de la clase.

Metodos

count

```
public int count()
```

Devuelve el número de departamentos.

Retorno:

El número de departamento mencionado.

getDepartment

```
public Department getDepartment(int index)
```

Devuelve un departamento de la lista.

Parametros:

index - Índice del departamento que se desea recuperar.

Retorno:

El departamento mencionado.

loadLite

```
public void loadLite()  
    throws java.lang.Exception
```

Carga la lista de departamentos con su información básica.

Lanza:

java.lang.Exception - Si se produce algún error en la carga de los departamentos.

loadLite

```
public void loadLite(int parentId)  
    throws java.lang.Exception
```

Carga la lista de departamentos de un mismo padre con su información básica.

Parametros:

parentId - Identificador de padre

Lanza:

java.lang.Exception - Si se produce algún error en la carga de los departamentos.



toString

```
public java.lang.String toString()
```

Muestra una representación de los valores de la clase en formato XML.

Retorno:

La representación mencionada.

Reemplaza:

toString in class java.lang.Object

toXML

```
public java.lang.String toXML()
```

Obtiene la información de la lista de departamentos en formato XML.

Retorno:

La lista de departamentos mencionada.

ieci.tecdoc.idoc.admin.api.user

Interface Group

< [Metodos](#) >

```
public interface Group
```

Proporciona toda la funcionalidad necesaria para manejar grupos invesDoc.

Metodos

delete

```
public void delete()  
    throws java.lang.Exception
```

Elimina el grupo.

Lanza:

java.lang.Exception - Si se produce algún error al eliminar.



getAdminUsers

```
public BasicUsers getAdminUsers()
```

Obtiene la lista de usuarios (identificadores y nombres) que pueden ser administradores de este grupo.

Retorno:

La lista mencionada.

getCreationDate

```
public java.util.Date getCreationDate()
```

Obtiene la fecha de creación del grupo.

Retorno:

La fecha mencionada.

getCreatorId

```
public int getCreatorId()
```

Obtiene el identificador del usuario que ha creado el grupo.

Retorno:

El identificador mencionado.

getDescription

```
public java.lang.String getDescription()
```

Obtiene la descripción del grupo.

Retorno:

El nombre mencionado.

getId

```
public int getId()
```

Obtiene el identificador de grupo.

Retorno:

El identificador mencionado.



getManagerId

```
public int getManagerId()
```

Obtiene el identificador del administrador del grupo.

Retorno:

El identificador mencionado.

getName

```
public java.lang.String getName()
```

Obtiene el nombre del grupo.

Retorno:

El nombre mencionado.

getPermissions

```
public Permissions getPermissions()
```

Devuelve la lista de permisos del grupo.

Retorno:

La lista mencionada.

getType

```
public int getType()
```

Obtiene el identificador del tipo de grupo.

Retorno:

El identificador mencionado.

getUpdateDate

```
public java.util.Date getUpdateDate()
```

Obtiene la fecha de actualización del grupo.

Retorno:

La fecha mencionada.



getUpdaterId

```
public int getUpdaterId()
```

Obtiene el identificador del usuario que ha actualizado el grupo.

Retorno:

El identificador mencionado.

getUsersByGroup

```
public Users getUsersByGroup(int id)
```

Devuelve la lista de usuarios del grupo. No es necesario cargar antes la clase Group.

Parametros:

id - Identificador del grupo.

Retorno:

Los usuarios del grupo.

load

```
public void load(int groupId)  
throws java.lang.Exception
```

Carga un grupo invesDoc.

Parametros:

groupId - Identificador del grupo.

Lanza:

java.lang.Exception - Si se produce algún error al leer la información del grupo.

load

```
public void load(java.lang.String groupName)  
throws java.lang.Exception
```

Carga un grupo invesDoc.

Parametros:

groupName - Nombre del grupo.

Lanza:

java.lang.Exception - Si se produce algún error al leer la información del grupo.



setDescription

```
public void setDescription(java.lang.String description)
```

Establece la descripción del grupo.

Parametros:

description - La descripción del grupo.

setManagerId

```
public void setManagerId(int managerId)
```

Establece el identificador del administrador del grupo.

Parametros:

managerId - El identificador del administrador.

setName

```
public void setName(java.lang.String name)
```

Establece el nombre del grupo.

Parametros:

name - Nombre del grupo.

store

```
public void store()  
    throws java.lang.Exception
```

Guarda el grupo. Se utiliza tanto para inserciones como para actualizaciones.

Lanza:

java.lang.Exception - Si se produce algún error al guardar. Por ejemplo, el grupo ya existe.

toString

```
public java.lang.String toString()
```

Muestra una representación de los valores de la clase en formato XML.

Retorno:

La representación mencionada.

Reemplaza:

toString in class java.lang.Object



toXML

```
public java.lang.String toXML()
```

Obtiene la información del grupo en formato XML.

Retorno:

La información mencionada.

ieci.tecdoc.idoc.admin.api.user

Interface GroupUserRel

< [Metodos](#) >

```
public interface GroupUserRel
```

Metodos

assignUserToGroup

```
public void assignUserToGroup(int groupId,
                             int userId)
    throws java.lang.Exception
```

Añade el usuario indicado al grupo indicado.

Parametros:

groupId - Identificador de grupo
userId - Identificador de usuario

Lanza:

java.lang.Exception -

deleteGroupUser

```
public void deleteGroupUser(int groupId,
                             int userId)
    throws java.lang.Exception
```

Elimina un usuario de un grupo

Parametros:

groupId - Identificador de grupo
userId - Identificador de usuario

Lanza:

java.lang.Exception -



loadUserGroupIds

```
public java.util.ArrayList loadUserGroupIds(int userId)
```

ieci.tecdoc.idoc.admin.api.user

Class Groups

```
java.lang.Object
|
+--ieci.tecdoc.idoc.admin.api.user.Groups
```

< [Constructores](#) > < [Metodos](#) >

```
public class Groups
extends java.lang.Object
```

Maneja la lista de grupos invesDoc.

Constructores

Groups

```
public Groups()
```

Construye un objeto de la clase.

Metodos

count

```
public int count()
```

Devuelve el número de grupos.

Retorno:

El número de grupo mencionado.

getGroup

```
public Group getGroup(int index)
```

Devuelve un grupo de la lista.

Parametros:

index - Indice del grupo que se desea recuperar.

Retorno:

El grupo mencionado.



loadLite

```
public void loadLite()  
    throws java.lang.Exception
```

Carga la lista de grupos con su información básica.

Lanza:

java.lang.Exception - Si se produce algún error en la carga de los grupos.

toString

```
public java.lang.String toString()
```

Muestra una representación de los valores de la clase en formato XML.

Retorno:

La representación mencionada.

Reemplaza:

toString in class java.lang.Object

toXML

```
public java.lang.String toXML()
```

Obtiene la información de la lista de grupos en formato XML.

Retorno:

La lista de grupo mencionada.

ieci.tecdoc.idoc.admin.api.user

Interface LdapGroup

< [Metodos](#) >

```
public interface LdapGroup
```

Proporciona toda la funcionalidad necesaria para manejar grupos invesDoc relacionados con un servicio de directorio Ldap.

Metodos



createFromLdapGroup

```
public void createFromLdapGroup(java.lang.String root,
                               java.lang.String attr,
                               java.lang.String value)
    throws java.lang.Exception
```

Permite crear un grupo a partir de uno existente en un servidor Ldap.

Parametros:

root - Dn del nodo raíz a partir del cual buscar.
attr - Atributo de búsqueda (típicamente cn).
value - Valor del atributo de búsqueda (típicamente el nombre del grupo).

Retorno:

El grupo que cumple los criterios de búsqueda.

Lanza:

java.lang.Exception - Si se ha producido algún error en la creación (por ejemplo, el grupo no existe en el servidor Ldap).

createGroup

```
public void createGroup(java.lang.String guid,
                       java.lang.String Dn)
    throws java.lang.Exception
```

Permite crear un grupo a partir de uno en un servidor Ldap.

Parametros:

guid - guid del grupo ldap.
Dn - Dn del grupo ldap.

Retorno:

El grupo.

Lanza:

java.lang.Exception - Si se ha producido algún error en la creación

delete

```
public void delete()
    throws java.lang.Exception
```

Elimina el grupo.

Lanza:

java.lang.Exception - Si se produce algún error al eliminar.



getFullName

```
public java.lang.String getFullName()
```

Devuelve el nombre completo del grupo.

Retorno:

El nombre mencionado.

getGuid

```
public java.lang.String getGuid()
```

Devuelve el identificador único del grupo.

Retorno:

El identificador mencionado.

getId

```
public int getId()
```

Devuelve el identificador del grupo.

Retorno:

El identificador mencionado.

getPermissions

```
public Permissions getPermissions()
```

Devuelve la lista de permisos del grupo.

Retorno:

La lista mencionada.

load

```
public void load(int groupId)  
    throws java.lang.Exception
```

Carga un grupo invesDoc.

Parametros:

groupId - Identificador del grupo.

Lanza:

java.lang.Exception - Si se produce algún error al leer la información del grupo.



loadFromGuid

```
public void loadFromGuid(java.lang.String groupGuid)
    throws java.lang.Exception
```

Carga un grupo invesDoc.

Parametros:

groupGuid - Identificador del grupo ldap.

Lanza:

java.lang.Exception - Si se produce algún error al leer la información del grupo.

setFullName

```
public void setFullName(java.lang.String fullName)
```

Establece el nombre completo del grupo.

Parametros:

fullName - El nombre completo del grupo.

store

```
public void store()
    throws java.lang.Exception
```

Guarda el grupo. Se utiliza tanto para inserciones como para actualizaciones.

Lanza:

java.lang.Exception - Si se produce algún error al guardar. Por ejemplo, el grupo ya existe.

toString

```
public java.lang.String toString()
```

Muestra una representación de los valores de la clase en formato XML.

Retorno:

La representación mencionada.

Reemplaza:

toString in class java.lang.Object

toXML

```
public java.lang.String toXML()
```

Obtiene la información del grupo Ldap en formato XML.

Retorno:

La información mencionada.



ieci.tecdoc.idoc.admin.api.user

Class LdapGroups

```
java.lang.Object
|
+--ieci.tecdoc.idoc.admin.api.user.LdapGroups
```

< [Constructores](#) > < [Metodos](#) >

```
public class LdapGroups
extends java.lang.Object
```

Maneja la lista de grupos invesDoc relacionados con un servicio de directorio Ldap.

Constructores

LdapGroups

```
public LdapGroups()
```

Construye un objeto de la clase.

Metodos

count

```
public int count()
```

Devuelve el número de grupos.

Retorno:

El número de grupo mencionado.

getGroup

```
public LdapGroup getGroup(int index)
```

Devuelve un grupo de la lista.

Parametros:

index - Indice del grupo que se desea recuperar.

Retorno:

El grupo mencionado.



loadFull

```
public void loadFull()  
    throws java.lang.Exception
```

Carga la lista de grupos con toda su información.

Lanza:

java.lang.Exception - Si se produce algún error en la carga de los grupos.

loadLite

```
public void loadLite()  
    throws java.lang.Exception
```

Carga la lista de grupos con su información básica.

Lanza:

java.lang.Exception - Si se produce algún error en la carga de los grupos.

toString

```
public java.lang.String toString()
```

Muestra una representación de los valores de la clase en formato XML.

Retorno:

La representación mencionada.

Reemplaza:

toString in class java.lang.Object

toXML

```
public java.lang.String toXML()
```

Obtiene la información de la lista de grupos en formato XML.

Retorno:

La lista de grupo mencionada.

ieci.tecdoc.idoc.admin.api.user

Interface LdapUser

< [Metodos](#) >

```
public interface LdapUser
```

Proporciona toda la funcionalidad necesaria para manejar usuarios invsesDoc relacionados con un servicio de directorio Ldap.



Metodos

createFromLdapUser

```
public void createFromLdapUser(java.lang.String root,
                               java.lang.String attr,
                               java.lang.String value)
    throws java.lang.Exception
```

Permite crear un usuario a partir de uno existente en un servidor Ldap.

Parametros:

root - Dn del nodo raíz a partir del cual buscar.
attr - Atributo de búsqueda (cn).
value - Valor del atributo de búsqueda (el nombre del usuario).

Retorno:

El usuario que cumple los criterios de búsqueda.

Lanza:

java.lang.Exception - Si se ha producido algún error en la creación (por ejemplo, el usuario no existe en el servidor Ldap).

createUser

```
public void createUser(java.lang.String guid,
                       java.lang.String fullName)
    throws java.lang.Exception
```

Permite crear un usuario a partir de uno existente en un servidor Ldap.

Parametros:

guid - guid del usuario en ldap.
fullName - fullname del usuario en ldap

Retorno:

El usuario .

Lanza:

java.lang.Exception - Si se ha producido algún error en la creación

delete

```
public void delete()
    throws java.lang.Exception
```

Elimina el usuario.

Lanza:

java.lang.Exception - Si se produce algún error al eliminar.



getFullName

```
public java.lang.String getFullName()
```

Devuelve el nombre completo del usuario.

Retorno:

El nombre mencionado.

getGuid

```
public java.lang.String getGuid()
```

Devuelve el identificador único del usuario.

Retorno:

El identificador mencionado.

getId

```
public int getId()
```

Devuelve el identificador del usuario.

Retorno:

El identificador mencionado.

getPermissions

```
public Permissions getPermissions()
```

Devuelve la lista de permisos del usuario.

Retorno:

La lista mencionada.

getProfiles

```
public UserProfiles getProfiles()
```

Devuelve la lista de perfiles del usuario.

Retorno:

La lista mencionada.



load

```
public void load(int userId)
    throws java.lang.Exception
```

Carga un usuario.

Parametros:

userId - Identificador del usuario.

Lanza:

java.lang.Exception - Si se produce algún error al leer la información del usuario.

loadFromGuid

```
public void loadFromGuid(java.lang.String userGuid)
    throws java.lang.Exception
```

Carga un usuario.

Parametros:

userGuid - Identificador del usuario ldap.

Lanza:

java.lang.Exception - Si se produce algún error al leer la información del usuario.

store

```
public void store()
    throws java.lang.Exception
```

Guarda el usuario. Se utiliza tanto para inserciones como para actualizaciones.

Lanza:

java.lang.Exception - Si se produce algún error al guardar. Por ejemplo, el usuario ya existe.

toString

```
public java.lang.String toString()
```

Muestra una representación de los valores de la clase en formato XML.

Retorno:

La representación mencionada.

Reemplaza:

toString in class java.lang.Object



toXML

```
public java.lang.String toXML()
```

Obtiene la información del usuario Ldap en formato XML.

Retorno:

La información mencionada.

ieci.tecdoc.idoc.admin.api.user

Class LdapUsers

```
java.lang.Object
|
+--ieci.tecdoc.idoc.admin.api.user.LdapUsers
```

< [Constructores](#) > < [Metodos](#) >

```
public class LdapUsers
extends java.lang.Object
```

Maneja la lista de usuarios invesDoc relacionados con un servicio de directorio Ldap.

Constructores

LdapUsers

```
public LdapUsers()
```

Construye un objeto de la clase.

Metodos

count

```
public int count()
```

Devuelve el número de usuarios.

Retorno:

El número de usuarios mencionado.



getUser

```
public LdapUser getUser(int index)
```

Devuelve un usuario de la lista.

Parametros:

index - Índice del usuario que se desea recuperar.

Retorno:

El usuario mencionado.

loadFull

```
public void loadFull()  
    throws java.lang.Exception
```

Carga la lista de usuarios con toda su información.

Lanza:

java.lang.Exception - Si se produce algún error en la carga de los usuarios.

loadLite

```
public void loadLite()  
    throws java.lang.Exception
```

Carga la lista de usuarios con su información básica.

Lanza:

java.lang.Exception - Si se produce algún error en la carga de los usuarios.

toString

```
public java.lang.String toString()
```

Muestra una representación de los valores de la clase en formato XML.

Retorno:

La representación mencionada.

Reemplaza:

toString in class java.lang.Object

toXML

```
public java.lang.String toXML()
```

Obtiene la información de la lista de usuarios en formato XML.

Retorno:

La lista de usuarios mencionada.



ieci.tecdoc.idoc.admin.api.user

Class Login

```
java.lang.Object
|
+--ieci.tecdoc.idoc.admin.api.user.Login
```

< [Metodos](#) >

```
public class Login
extends java.lang.Object
```

Proporciona funcionalidad de acceso a la administración de invesDoc.

Metodos

doAdmLogin

```
public static ieci.tecdoc.sbo.acs.base.AcsAccessTokenProducts doAdmLogin( java.lang.String name,
                                                                    java.lang.String
password,
                                                                    int numTries)
throws java.lang.Exception
```

Realiza la conexión a las aplicaciones de Administración (Archivos, Usuarios y Volúmenes). Valida los permisos del usuario para utilizar dichas aplicaciones y le proporciona un token de acceso para cada una de ellas.

Parametros:

name - Nombre del usuario.
password - Contraseña.
numTries - Número de intentos incorrectos de usuario-contraseña.

Retorno:

Objeto con array de tokens de acceso.

Lanza:

java.lang.Exception - Si se produce algún error en el login del usuario.



doLogin

```
public static ieci.tecdoc.sbo.acs.base.AcsAccessToken doLogin( java.lang.String name,
                                                            java.lang.String password,
                                                            int productId,
                                                            int numTries)
                                                            throws java.lang.Exception
```

Realiza la conexión a una aplicación. Valida los permisos del usuario para utilizar dicha aplicación y le proporciona un token de acceso.

Parámetros:

name - Nombre del usuario.
 password - Contraseña.
 productId - Identificador de la aplicación que desea hacer login.
 numTries - Número de intentos incorrectos de usuario-contraseña.

Retorno:

El token de acceso.

Lanza:

java.lang.Exception - Si se produce algún error en el login del usuario.

ieci.tecdoc.idoc.admin.api.user

Interface Permission

< [Metodos](#) >

public interface **Permission**

Maneja los permisos de un usuario, grupo o departamento. Cada permiso es una terna compuesta por:

[Identificador de objeto, Identificador de producto, Permiso].

El establecimiento de varios permisos se realiza mediante la unión de todos los valores a través de la sentencia |.

UserDefs

Metodos

getDestination

```
public int getDestination()
```

Devuelve el identificador del destinatario de los permisos.

Retorno:

El destinatario mencionado.



getId

```
public int getId()
```

Devuelve el identificador del objeto.

Retorno:

El identificador mencionado.

getPermission

```
public int getPermission()
```

Devuelve el permiso.

Retorno:

El permiso mencionado.

getProduct

```
public int getProduct()
```

Devuelve el identificador del producto al que se asigna el permiso.

Retorno:

El identificador mencionado.

setPermission

```
public void setPermission(int permission)
```

Establece el permiso.

Parametros:

permission - El permiso mencionado.

toString

```
public java.lang.String toString()
```

Muestra una representación de los valores de la clase en formato XML.

Retorno:

La representación mencionada.

Reemplaza:

toString in class java.lang.Object



toXML

```
public java.lang.String toXML(boolean header)
```

Obtiene la información del permiso en formato XML.

Parametros:

header - Indica si hay que incluir la cabecera xml o no.

Retorno:

La información mencionada.

ieci.tecdoc.idoc.admin.api.user

Interface Permissions

< [Metodos](#) >

```
public interface Permissions
```

Mantiene el conjunto de permisos de un usuario, grupo o departamento.

Metodos

getProductPermission

```
public Permission getProductPermission(int productId)
                                     throws java.lang.Exception
```

Devuelve un permiso de la lista a partir del identificador de producto.

Parametros:

productId - Identificador del producto cuyo permiso se desea.

Retorno:

El permiso mencionado.

Lanza:

java.lang.Exception - Si el permiso para el producto solicitado no existe.

toString

```
public java.lang.String toString()
```

Muestra una representación de los valores de la clase en formato XML.

Retorno:

La representación mencionada.

Reemplaza:

toString in class java.lang.Object



toXML

```
public java.lang.String toXML(boolean header)
```

Obtiene la información de los permisos en formato XML.

Parametros:

header - Indica si hay que incluir la cabecera xml o no.

Retorno:

La información mencionada.

ieci.tecdoc.idoc.admin.api.user

Interface User

< [Metodos](#) >

```
public interface User
```

Proporciona toda la funcionalidad necesaria para manejar usuarios invesDoc.

Metodos

delete

```
public void delete()  
    throws java.lang.Exception
```

Elimina el usuario.

Lanza:

java.lang.Exception - Si se produce algún error al eliminar.

getCreationDate

```
public java.util.Date getCreationDate()
```

Obtiene la fecha de creación del usuario.

Retorno:

La fecha mencionada.



getCreatorId

```
public int getCreatorId()
```

Obtiene el identificador del usuario que ha creado el usuario.

Retorno:

El identificador mencionado.

getDeptId

```
public int getDeptId()
```

Obtiene el identificador del departamento al que pertenece el usuario.

Retorno:

El identificador mencionado.

getDeptsIdMgr

```
public java.util.ArrayList getDeptsIdMgr(int userId)
    throws java.lang.Exception
```

Obtiene los identificadores de los departamentos a los que administra el usuario indicado.

Parametros:

userId - Identificador de usuario

Retorno:

Lista de enteros con los identificadores de departamento

Lanza:

java.lang.Exception - Si se produce algún error.

getDescription

```
public java.lang.String getDescription()
```

Obtiene la descripción del usuario.

Retorno:

El dato mencionado.



getGroupsIdMgr

```
public java.util.ArrayList getGroupsIdMgr(int userId)
    throws java.lang.Exception
```

Otiene los identificadores de los grupos a los que administra el usuario especificado.

Parametros:

userId - Identificador de usuario

Retorno:

Lista de enteros con los identificadores de grupo

Lanza:

java.lang.Exception - Si se produce algún error.

getId

```
public int getId()
```

Obtiene el identificador del usuario.

Retorno:

El identificador mencionado.

getName

```
public java.lang.String getName()
```

Obtiene el nombre del usuario.

Retorno:

El nombre mencionado.

getPermissions

```
public Permissions getPermissions()
```

Obtiene la lista de permisos del usuario.

Retorno:

La lista mencionada.

getProfiles

```
public UserProfiles getProfiles()
```

Obtiene la lista de perfiles del usuario.

Retorno:

La lista mencionada.



getPwdLastUpdTs

```
public double getPwdLastUpdTs()
```

Obtiene la fecha de la última modificación de la contraseña. Horas transcurridas desde 01-01-1971.

Retorno:

La fecha mencionada.

getPwdbc

```
public boolean getPwdbc()
```

Obtiene la obligatoriedad sobre la inicialización de la contraseña.

Retorno:

true si es obligatoria la inicialización de la contraseña. false si no es obligatoria la inicialización de la contraseña.

getPwvpccheck

```
public boolean getPwvpccheck()
```

Obtiene la información de comprobar la caducidad de la contraseña.

Retorno:

true si se comprueba la caducidad de la contraseña. false si no se comprueba.

getState

```
public int getState()
```

Obtiene el estado del usuario.

Retorno:

El dato mencionado.

getUpdateDate

```
public java.util.Date getUpdateDate()
```

Obtiene la fecha de actualización del usuario.

Retorno:

La fecha mencionada.



getUpdaterId

```
public int getUpdaterId()
```

Obtiene el identificador del usuario que ha actualizado el usuario.

Retorno:

El identificador mencionado.

load

```
public void load(int userId)
    throws java.lang.Exception
```

Carga un usuario.

Parametros:

userId - Identificador del usuario.

Lanza:

java.lang.Exception - Si se produce algún error al leer la información del usuario.

load

```
public void load(java.lang.String userName)
    throws java.lang.Exception
```

Carga un usuario.

Parametros:

userName - Nombre del usuario.

Lanza:

java.lang.Exception - Si se produce algún error al leer la información del usuario.

setDeptId

```
public void setDeptId(int deptId)
```

Establece el identificador del departamento del usuario.

Parametros:

deptId - El identificador del departamento.

setDescription

```
public void setDescription(java.lang.String description)
```

Establece la descripción del usuario.

Parametros:

description - La descripción del usuario.



setName

```
public void setName(java.lang.String name)
```

Establece el nombre del usuario.

Parametros:

name - El nombre del usuario.

setPassword

```
public void setPassword(java.lang.String password)
```

Establece la password del usuario.

Parametros:

password - La password del usuario.

setPwdbc

```
public void setPwdbc(boolean pwdbc)
```

Establece la obligatoriedad de la inicialización de la contraseña.

Parametros:

pwdbc - `true` si es obligatoria la inicialización de la contraseña. `false` si no es obligatoria la inicialización de la contraseña.

setPwvpccheck

```
public void setPwvpccheck(boolean pwvpccheck)
```

Establece la información de comprobar la caducidad de la contraseña.

Parametros:

pwvpccheck - `true` si se comprueba la caducidad de la contraseña. `false` si no se comprueba.

setState

```
public void setState(int state)
```

Establece el estado del usuario.

Parametros:

state - El estado del usuario.



store

```
public void store()  
    throws java.lang.Exception
```

Guarda el usuario. Se utiliza tanto para inserciones como para actualizaciones.

Lanza:

java.lang.Exception - Si se produce algún error al guardar. Por ejemplo, el usuario ya existe.

toString

```
public java.lang.String toString()
```

Muestra una representación de los valores de la clase en formato XML.

Retorno:

La representación mencionada.

Reemplaza:

toString in class java.lang.Object

toXML

```
public java.lang.String toXML()
```

Obtiene la información del usuario en formato XML.

Retorno:

La información mencionada.

ieci.tecdoc.idoc.admin.api.user

Interface UserAccess

< [Metodos](#) >

```
public interface UserAccess
```

Maneja los accesos a la gestión de departamentos, grupos y usuarios.

Metodos



userCanAssingUserToGroup

```
public boolean userCanAssingUserToGroup(int connectedUserId,  
                                         int userId,  
                                         int groupId)  
    throws java.lang.Exception
```

Obtiene si el usuario conectado puede asignar usuario a grupo.

Parametros:

connectedUserId - Identificador del usuario conectado.
userId - Identificador del usuario a asignar.
groupId - Identificador del grupo.

Retorno:

true / false

Lanza:

java.lang.Exception - Errores (usuario ya existe).

userCanCreateDept

```
public boolean userCanCreateDept(int connectedUserId,  
                                  int parentDeptId)  
    throws java.lang.Exception
```

Obtiene si el usuario conectado puede crear departamentos.

Parametros:

connectedUserId - Identificador del usuario conectado.
parentDeptId - Identificador del departamento padre.

Retorno:

true / false

Lanza:

java.lang.Exception - Errores

userCanCreateGroup

```
public boolean userCanCreateGroup(int connectedUserId)  
    throws java.lang.Exception
```

Obtiene si el usuario conectado puede crear grupos.

Parametros:

connectedUserId - Identificador del usuario conectado.

Retorno:

true / false

Lanza:

java.lang.Exception - Errores



userCanCreateUser

```
public boolean userCanCreateUser(int connectedUserId,
                                int deptId)
    throws java.lang.Exception
```

Obtiene si el usuario conectado puede crear usuarios

Parametros:

connectedUserId - Identificador del usuario conectado.
deptId - Identificador del departamento padre.

Retorno:

true / false

Lanza:

java.lang.Exception - Errores

userCanDeleteDept

```
public boolean userCanDeleteDept(int connectedUserId,
                                 int DeptId)
    throws java.lang.Exception
```

Obtiene si el usuario conectado puede eliminar el departamento indicado.

Parametros:

connectedUserId - Identificador del usuario conectado.
DeptId - Identificador del departamento.

Retorno:

true / false

Lanza:

java.lang.Exception - Errores

userCanDeleteGroup

```
public boolean userCanDeleteGroup(int connectedUserId,
                                  int GroupId)
    throws java.lang.Exception
```

Obtiene si el usuario conectado puede eliminar el grupo indicado.

Parametros:

connectedUserId - Identificador del usuario conectado.
GroupId - Identificador del grupo.

Retorno:

true / false

Lanza:

java.lang.Exception - Errores



userCanDeleteGroupUser

```
public boolean userCanDeleteGroupUser(int connectedUserId,
                                       int groupId)
    throws java.lang.Exception
```

Obtiene si el usuario conectado puede eliminar usuario de grupo

Parametros:

connectedUserId - Identificador del usuario conectado.
groupId - Identificador del grupo.

Retorno:

true / false

Lanza:

java.lang.Exception - Errores

userCanDeleteUser

```
public boolean userCanDeleteUser(int connectedUserId,
                                  int UserId)
    throws java.lang.Exception
```

Obtiene si el usuario conectado puede eliminar el usuario indicado.

Parametros:

connectedUserId - Identificador del usuario conectado.
UserId - Identificador de usuario

Retorno:

true / false

Lanza:

java.lang.Exception - Errores

userCanEditDept

```
public boolean userCanEditDept(int connectedUserId,
                                int DeptId)
    throws java.lang.Exception
```

Obtiene si el usuario conectado puede modificadr el departamento indicado.

Parametros:

connectedUserId - Identificador del usuario conectado.
DeptId - Identificador de departamento.

Retorno:

true / false

Lanza:

java.lang.Exception - Errores



userCanEditGroup

```
public boolean userCanEditGroup(int connectedUserId,
                                int GroupId)
    throws java.lang.Exception
```

Obtiene si el usuario conectado puede modificar el grupo indicado.

Parametros:

connectedUserId - Identificador de usuario conectado.
GroupId - Identificador del grupo.

Retorno:

true / false

Lanza:

java.lang.Exception - Errores

userCanEditUser

```
public boolean userCanEditUser(int connectedUserId,
                                int UserId)
    throws java.lang.Exception
```

Obtiene si el usuario conectado puede modificar el usuario indicado.

Parametros:

connectedUserId - Identificador del usuario conectado.
UserId - Identificador de usuario

Retorno:

true / false

Lanza:

java.lang.Exception - Errores

userCanMoveDept

```
public boolean userCanMoveDept(int connectedUserId,
                                int DeptId,
                                int dstDeptId)
    throws java.lang.Exception
```

Obtiene si el usuario conectado puede mover el departamento.

Parametros:

connectedUserId - Identificador del usuario conectado.
DeptId - Identificador del departamento.
dstDeptId - Identificador del departamento destino.

Retorno:

true / false

Lanza:

java.lang.Exception - Errores



userCanMoveUser

```
public boolean userCanMoveUser(int connectedUserId,
                               int UserId,
                               int dstDeptId)
    throws java.lang.Exception
```

Obtiene si el usuario conectado puede mover el usuario indicado al departamento indicado.

Parametros:

connectedUserId - Identificador del usuario conectado.
UserId - Identificador de usuario.
dstDeptId - Identificador del departamento destino.

Retorno:

true / false

Lanza:

java.lang.Exception - Errores

userCanViewDept

```
public boolean userCanViewDept(int connectedUserId)
    throws java.lang.Exception
```

Obtiene si el usuario conectado tiene permiso de consulta de departamento.

Parametros:

connectedUserId - Identificador del usuario conectado.

Retorno:

true / false

Lanza:

java.lang.Exception - Errores

userCanViewGroup

```
public boolean userCanViewGroup(int connectedUserId)
    throws java.lang.Exception
```

Obtiene si el usuario conectado puede consultar el grupo.

Parametros:

connectedUserId - Identificador del usuario conectado.

Retorno:

true / false

Lanza:

java.lang.Exception - Errores



userCanViewUser

```
public boolean userCanViewUser(int connectedUserId)
    throws java.lang.Exception
```

Obtiene si el usuario conectado puede consultar usuarios

Parametros:

connectedUserId - Identificador del usuario conectado.

Retorno:

true / false

Lanza:

java.lang.Exception - Errores

ieci.tecdoc.idoc.admin.api.user

Class UserDefs

```
java.lang.Object
|
+--ieci.tecdoc.idoc.admin.api.user.UserDefs
```

< [Campos](#) >

```
public class UserDefs
    extends java.lang.Object
```

Proporciona las definiciones de todos los identificadores relativos a la administración de usuarios.

Campos

PERMISSION_ALL

```
public static final int PERMISSION_ALL
    Todos los permisos.
```

PERMISSION_CREATION

```
public static final int PERMISSION_CREATION
    Permiso de creación.
```

PERMISSION_DELETION

```
public static final int PERMISSION_DELETION
    Permiso de borrado.
```

PERMISSION_NONE



```
public static final int PERMISSION_NONE
    Ningún permiso.
```

PERMISSION_PRINTING

```
public static final int PERMISSION_PRINTING
    Permiso de impresión.
```

PERMISSION_QUERY

```
public static final int PERMISSION_QUERY
    Permiso de búsqueda.
```

PERMISSION_UPDATE

```
public static final int PERMISSION_UPDATE
    Permiso de actualización.
```

PRODUCT_IDOC

```
public static final int PRODUCT_IDOC
    Identificador de producto de la Aplicación de Consulta.
```

PRODUCT_IFLOW

```
public static final int PRODUCT_IFLOW
    Identificador de producto de invesFlow.
```

PRODUCT_ISICRES

```
public static final int PRODUCT_ISICRES
    Identificador de producto de inveSicres.
```

PRODUCT_SYSTEM

```
public static final int PRODUCT_SYSTEM
    Identificador de producto de las Herramientas de Sistema.
```

PRODUCT_USER

```
public static final int PRODUCT_USER
    Identificador de producto del Administrador de Usuarios.
```

PRODUCT_VOLUME



```
public static final int PRODUCT_VOLUME
    Identificador de producto del Administrador de Volúmenes.
```

PROFILE_MANAGER

```
public static final int PROFILE_MANAGER
    Identificador de perfil Administrador.
```

PROFILE_NONE

```
public static final int PROFILE_NONE
    Identificador de perfil Sin Derechos.
```

PROFILE_STANDARD

```
public static final int PROFILE_STANDARD
    Identificador de perfil Estándar.
```

PROFILE_SUPERUSER

```
public static final int PROFILE_SUPERUSER
    Identificador de perfil Superusuario.
```

USER_STAT_DEF

```
public static final int USER_STAT_DEF
    Estado del usuario por definición.
```

USER_STAT_LOCKED

```
public static final int USER_STAT_LOCKED
    Estado del usuario bloqueado.
```

ieci.tecdoc.idoc.admin.api.user

Interface UserProfile

< [Metodos](#) >

```
public interface UserProfile
```

Maneja los perfiles de un usuario. Cada permiso es una terna compuesta por:
[Identificador de usuario, Identificador de Producto, Perfil].

```
    UserDefs
```



Metodos

getProduct

```
public int getProduct()
```

Devuelve el identificador del producto al que se asigna el perfil.

Retorno:

El identificador mencionado.

getProfile

```
public int getProfile()
```

Devuelve el perfil asignado al producto.

Retorno:

El perfil mencionado.

getUserId

```
public int getUserId()
```

Devuelve el identificador del usuario.

Retorno:

El identificador mencionado.

setProfile

```
public void setProfile(int profile)
```

Establece el perfil asignado al producto.

Parametros:

profile - El perfil mencionado.

toString

```
public java.lang.String toString()
```

Muestra una representación de los valores de la clase en formato XML.

Retorno:

La representación mencionada.

Reemplaza:

toString in class java.lang.Object



toXML

```
public java.lang.String toXML(boolean header)
```

Obtiene la información del perfil del usuario en formato XML.

Parametros:

header - Indica si hay que incluir la cabecera xml o no.

Retorno:

La información mencionada.

ieci.tecdoc.idoc.admin.api.user

Interface UserProfiles

< [Metodos](#) >

```
public interface UserProfiles
```

Mantiene el conjunto de perfiles de un usuario.

Metodos

getProductProfile

```
public UserProfile getProductProfile(int productId) throws java.lang.Exception
```

Devuelve un perfil de la lista a partir del identificador de producto.

Parametros:

productId - Identificador del producto cuyo perfil se desea.

Retorno:

El perfil mencionado.

Lanza:

java.lang.Exception - Si el perfil para el producto solicitado no existe.

toString

```
public java.lang.String toString()
```

Muestra una representación de los valores de la clase en formato XML.

Retorno:

La representación mencionada.

Reemplaza:

toString in class java.lang.Object



toXML

```
public java.lang.String toXML(boolean header)
```

Obtiene la información de los perfiles de un usuario en formato XML.

Parametros:

header - Indica si hay que incluir la cabecera xml o no.

Retorno:

La información mencionada.

ieci.tecdoc.idoc.admin.api.user

Class Users

```
java.lang.Object
|
+--ieci.tecdoc.idoc.admin.api.user.Users
```

< [Constructores](#) > < [Metodos](#) >

```
public class Users
extends java.lang.Object
```

Maneja la lista de usuarios invesDoc.

Constructores

Users

```
public Users()
```

Construye un objeto de la clase.

Metodos

count

```
public int count()
```

Devuelve el número de usuarios.

Retorno:

El número de usuarios mencionado.



getUser

```
public User getUser(int index)
```

Devuelve un usuario de la lista.

Parametros:

index - Índice del usuario que se desea recuperar.

Retorno:

El usuario mencionado.

loadByDept

```
public void loadByDept(int deptId)
    throws java.lang.Exception
```

Carga la lista de usuarios con su información básica, según un departamento.

Parametros:

deptId - Identificador del departamento.

Lanza:

java.lang.Exception - Si se produce algún error en la carga de los usuarios.

loadByGroup

```
public void loadByGroup(int groupId)
    throws java.lang.Exception
```

Carga la lista de usuarios con su información básica, según un grupo.

Parametros:

groupId - Identificador del grupo.

Lanza:

java.lang.Exception - Si se produce algún error en la carga de los usuarios.

moveUser

```
public void moveUser(int userId,
    int dstDeptId)
```

mueve el usuario al departamento indicado.

Parametros:

userId - Identificador de usuario.
dstDeptId - Identificador del departamento destino.



toString

```
public java.lang.String toString()
```

Muestra una representación de los valores de la clase en formato XML.

Retorno:

La representación mencionada.

Reemplaza:

toString in class java.lang.Object

toXML

```
public java.lang.String toXML()
```



Paquete ieci.tecdoc.idoc.admin.api.volume

Interfaces

[Repository](#)

Proporciona toda la funcionalidad necesaria para manejar repositorios de volúmenes en invesDoc.

[Volume](#)

Proporciona toda la funcionalidad necesaria para manejar volúmenes en invesDoc.

[VolumeList](#)

Proporciona toda la funcionalidad necesaria para manejar listas de volúmenes en invesDoc.

Clases

[Repositories](#)

Maneja la lista de repositorios de invesDoc.

[VolumeDefs](#)

Proporciona las definiciones de todos los identificadores relativos a la administración de volúmenes.

[VolumeLists](#)

Maneja la lista de la lista de volúmenes de invesDoc.

[Volumes](#)

Maneja la lista de volúmenes de invesDoc.

ieci.tecdoc.idoc.admin.api.volume

Class Repositories

```
java.lang.Object
|
+--ieci.tecdoc.idoc.admin.api.volume.Repositories
```

< [Constructores](#) > < [Metodos](#) >

```
public class Repositories
extends java.lang.Object
```

Maneja la lista de repositorios de invesDoc.

Constructores

Repositories

```
public Repositories()
```

Construye un objeto de la clase.



Metodos

count

```
public int count()
```

Devuelve el número de repositorios.

Retorno:

El número de repositorios mencionado.

getRepository

```
public Repository getRepository(int index)
```

Devuelve un repositorio de la lista.

Parametros:

index - Indice del repositorio que se desea recuperar.

Retorno:

El repositorio mencionado.

load

```
public void load()  
    throws java.lang.Exception
```

Carga la lista de repositorios con su información básica.

Lanza:

java.lang.Exception - Si se produce algún error en la carga de los repositorios.

toString

```
public java.lang.String toString()
```

Muestra una representación de los valores de la clase en formato XML.

Retorno:

La representación mencionada.

Reemplaza:

toString in class java.lang.Object



toXML

```
public java.lang.String toXML()
```

Obtiene la información de la lista de repositorios en formato XML.

Retorno:

La lista de repositorios mencionada.

ieci.tecdoc.idoc.admin.api.volume

Interface Repository

< [Metodos](#) >

```
public interface Repository
```

Proporciona toda la funcionalidad necesaria para manejar repositorios de volúmenes en invesDoc.

Metodos

delete

```
public void delete()  
    throws java.lang.Exception
```

Elimina el repositorio.

Lanza:

java.lang.Exception - Si se produce algún error al eliminar.

getCreationDate

```
public java.util.Date getCreationDate()
```

Obtiene la fecha de creación del repositorio.

Retorno:

La fecha mencionada.

getCreatorId

```
public int getCreatorId()
```

Obtiene el identificador del usuario que ha creado el repositorio.

Retorno:

El identificador mencionado.



getId

```
public int getId()
```

Obtiene el identificador del repositorio.

Retorno:

El identificador mencionado.

getName

```
public java.lang.String getName()
```

Obtiene el nombre del repositorio.

Retorno:

El nombre mencionado.

getOs

```
public int getOs()
```

Obtiene El identificador del sistema operativo del repositorio.

Retorno:

El identificador mencionado.

getPassword

```
public java.lang.String getPassword()
```

Obtiene el password para conectarse por ftp al repositorio.

Retorno:

Nombre mencionado.

getPath

```
public java.lang.String getPath()
```

Obtiene el path completo del repositorio.

Retorno:

El nombre mencionado



getPort

```
public int getPort()
```

Obtiene el puerto ftp del repositorio.

Retorno:

El nombre mencionado.

getRemarks

```
public java.lang.String getRemarks()
```

Obtiene los comentarios del repositorio.

Retorno:

El nombre mencionado.

getServer

```
public java.lang.String getServer()
```

Obtiene el servidor ftp del repositorio.

Retorno:

El nombre mencionado.

getState

```
public int getState()
```

Obtiene el estado del repositorio Ver - VolumeDefs

Retorno:

El estado mencionado

getType

```
public int getType()
```

Obtiene el identificador del tipo de repositorio.

Retorno:

El identificador mencionado.



getUpdateDate

```
public java.util.Date getUpdateDate()
```

Obtiene la fecha de actualización del repositorio.

Retorno:

La fecha mencionada.

getUpdaterId

```
public int getUpdaterId()
```

Obtiene el identificador del usuario que ha actualizado el repositorio.

Retorno:

El identificador mencionado.

getUser

```
public java.lang.String getUser()
```

Obtiene el usuario para conectarse por ftp al repositorio.

Retorno:

El nombre mencionado.

getVolumes

```
public Volumes getVolumes(int id)
```

Obtiene los volúmenes asociados al repositorio. No es necesario cargar el objeto antes.

Parametros:

id - identificador del repositorio

Retorno:

Los datos mencionados.

load

```
public void load(int repositoryId)  
    throws java.lang.Exception
```

Carga un repositorio de invesDoc.

Parametros:

repositoryId - Identificador del repositorio.

Lanza:

java.lang.Exception - Si se produce algún error al leer la información del repositorio.



setName

```
public void setName(java.lang.String name)
```

Establece el nombre del repositorio.

Parametros:

name - El nombre del repositorio.

setOs

```
public void setOs(int os)
```

Establece el sistema operativo del repositorio.

Parametros:

os - El sistema operativo del repositorio.

setPassword

```
public void setPassword(java.lang.String password)
```

Establece el password para conectarse por ftp al repositorio.

Parametros:

password - El password para conectarse por ftp al repositorio.

setPath

```
public void setPath(java.lang.String path)
```

Establece el path completo del repositorio.

Parametros:

path - El path completo del repositorio.

setPort

```
public void setPort(int port)
```

Establece el puerto ftp del repositorio.

Parametros:

port - El puerto ftp del repositorio.



setRemarks

```
public void setRemarks(java.lang.String remarks)
```

Establece los comentarios del repositorio.

Parametros:

remarks - Los comentarios del repositorio

setServer

```
public void setServer(java.lang.String server)
```

Establece el servidor ftp del repositorio.

Parametros:

server - El servidor ftp del repositorio

setState

```
public void setState(int state)
```

Establece el estado del repositorio

Parametros:

state - El estado

setType

```
public void setType(int type)
```

Establece el tipo de repositorio.

Parametros:

type - El tipo de repositorio.

setUser

```
public void setUser(java.lang.String user)
```

Establece el usuario para conectarse por ftp al repositorio.

Parametros:

user - el usuario para conectarse por ftp al repositorio.



store

```
public void store()  
    throws java.lang.Exception
```

Guarda el repositorio. Se utiliza tanto para inserciones como para actualizaciones.

Lanza:

java.lang.Exception - Si se produce algún error al guardar. Por ejemplo, el repositorio ya existe.

toString

```
public java.lang.String toString()
```

Muestra una representación de los valores de la clase en formato XML.

Retorno:

La representación mencionada.

Reemplaza:

toString in class java.lang.Object

toXML

```
public java.lang.String toXML()
```

Obtiene la información del repositorio en formato XML.

Retorno:

La información mencionada.

ieci.tecdoc.idoc.admin.api.volume

Interface Volume

< [Metodos](#) >

```
public interface Volume
```

Proporciona toda la funcionalidad necesaria para manejar volúmenes en invesDoc.

Metodos



delete

```
public void delete()  
    throws java.lang.Exception
```

Elimina el volumen.

Lanza:

java.lang.Exception - Si se produce algún error al eliminar.

getActSize

```
public java.lang.String getActSize()
```

Obtiene la ocupación en curso en bytes.

Retorno:

El valor mencionado.

getCreationDate

```
public java.util.Date getCreationDate()
```

Obtiene la fecha de creación del volumen.

Retorno:

La fecha mencionada.

getCreatorId

```
public int getCreatorId()
```

Obtiene el identificador del usuario que ha creado el volumen.

Retorno:

El identificador mencionado.

getId

```
public int getId()
```

Obtiene el identificador del repositorio.

Retorno:

El identificador mencionado.



getMaxSize

```
public java.lang.String getMaxSize()
```

Obtiene el tamaño máximo del volumen en bytes.

Retorno:

El dato mencionado.

getName

```
public java.lang.String getName()
```

Obtiene el nombre del volumen.

Retorno:

El nombre mencionado.

getNumFiles

```
public int getNumFiles()
```

Obtiene el número de ficheros en curso del volumen.

Retorno:

El valor mencionado.

getPath

```
public java.lang.String getPath()
```

Obtiene la ruta del volumen relativo a su repositorio.

Retorno:

El nombre mencionado.

getRemarks

```
public java.lang.String getRemarks()
```

Obtiene los comentarios del volumen.

Retorno:

El dato mencionado.



getRepId

```
public int getRepId()
```

Obtiene el identificador del repositorio.

Retorno:

El identificador mencionado.

getRepType

```
public int getRepType()
```

Obtiene el identificador del tipo de repositorio donde está almacenado el volumen, puede ser PFS, valor 3 o FTP, valor 1.

Retorno:

El identificador mencionado.

getState

```
public int getState()
```

Obtiene el estado del volumen.

Retorno:

El valor mencionado.

getTemp

```
public int getTemp()
```

Obtiene la información de si el volumen es temporal(1) o no(0).

Retorno:

El valor mencionado.

getUpdateDate

```
public java.util.Date getUpdateDate()
```

Obtiene la fecha de actualización del volumen.

Retorno:

La fecha mencionada.



getUpdaterId

```
public int getUpdaterId()
```

Obtiene el identificador del usuario que ha actualizado el volumen.

Retorno:

El identificador mencionado.

load

```
public void load(int volId)
    throws java.lang.Exception
```

Carga un volumen de invesDoc.

Parametros:

volId - Identificador del volumen.

Lanza:

java.lang.Exception - Si se produce algún error al leer la información del volumen.

loadListIds

```
public void loadListIds(int volId,
    java.util.ArrayList listIds)
    throws java.lang.Exception
```

Obtiene los identificadores de las lista a las que está asociado el volumen

Parametros:

volId - Identificador de volumen
listIds - Array de identificadores de lista

Lanza:

java.lang.Exception -

setMaxSize

```
public void setMaxSize(java.lang.String maxSize)
```

Establece el tamaño máximo del volumen en bytes.

Parametros:

maxSize - El tamaño máximo mencionado en bytes.



setName

```
public void setName(java.lang.String name)
```

Establece el nombre del volumen.

Parametros:

name - El nombre del volumen.

setPath

```
public void setPath(java.lang.String path)
```

Establece la ruta del volumen relativo a su repositorio.

Parametros:

path - La ruta del volumen.

setRemarks

```
public void setRemarks(java.lang.String remarks)
```

Establece los comentarios del volumen.

Parametros:

remarks - Los comentarios del volumen.

setRepId

```
public void setRepId(int repId)
```

Establece el identificador del repositorio.

Parametros:

repId - El identificador del repositorio.

setState

```
public void setState(int state)
```

Establece el estado del volumen.

Parametros:

state - El estado del volumen.



store

```
public void store()
    throws java.lang.Exception
```

Guarda el volumen. Se utiliza tanto para inserciones como para actualizaciones.

Lanza:

java.lang.Exception - Si se produce algún error al guardar. Por ejemplo, el volumen ya existe.

toString

```
public java.lang.String toString()
```

Muestra una representación de los valores de la clase en formato XML.

Retorno:

La representación mencionada.

Reemplaza:

toString in class java.lang.Object

toXML

```
public java.lang.String toXML()
```

Obtiene la información del volumen en formato XML.

Retorno:

La información mencionada.

ieci.tecdoc.idoc.admin.api.volume

Class VolumeDefs

```
java.lang.Object
|
+--ieci.tecdoc.idoc.admin.api.volume.VolumeDefs
```

< [Campos](#) > < [Constructores](#) >

```
public class VolumeDefs
    extends java.lang.Object
```

Proporciona las definiciones de todos los identificadores relativos a la administración de volúmenes.

Campos

FTP_PORT_DEFAULT



```
public static final int FTP_PORT_DEFAULT
    Valor por defecto del puerto para conectarse por FTP.
```

OS_NT

```
public static final int OS_NT
    Identificador del Sistema operativo de repositorio Windows NT.
```

OS_UNIX

```
public static final int OS_UNIX
    Identificador del Sistema operativo de repositorio Unix.
```

OS_WINDOWS

```
public static final int OS_WINDOWS
    Identificador del Sistema operativo de repositorio Windows.
```

REP_STAT_OFFLINE

```
public static final int REP_STAT_OFFLINE
    Identificador del estado de repositorio Disponible.
```

REP_STAT_READONLY

```
public static final int REP_STAT_READONLY
    Identificador del estado de repositorio Sólo lectura.
```

REP_TYPE_FTP

```
public static final int REP_TYPE_FTP
    Identificador del repositorio de tipo FTP.
```

REP_TYPE_PFS

```
public static final int REP_TYPE_PFS
    Identificador del repositorio de tipo PFS.
```

VOL_STAT_FULL

```
public static final int VOL_STAT_FULL
    Identificador del estado de volumen Lleno.
```

VOL_STAT_NOT_READY



```
public static final int VOL_STAT_NOT_READY
    Identificador del estado de volumen Disponible.
```

VOL_STAT_READONLY

```
public static final int VOL_STAT_READONLY
    Identificador del estado de volumen Sólo lectura.
```

Constructores

VolumeDefs

```
public VolumeDefs()
```

ieci.tecdoc.idoc.admin.api.volume

Interface VolumeList

< [Metodos](#) >

```
public interface VolumeList
```

Proporciona toda la funcionalidad necesaria para manejar listas de volúmenes en invesDoc.

Metodos

add

```
public void add(int volId)
    throws java.lang.Exception
```

Añade un volumen a la lista de volúmenes.

Parametros:

volId - El identificador del volumen.

Lanza:

java.lang.Exception - Si se produce algún error al añadir.

delete

```
public void delete()
    throws java.lang.Exception
```

Elimina la lista.

Lanza:

java.lang.Exception - Si se produce algún error al eliminar.



deleteVolume

```
public void deleteVolume(int volId)
    throws java.lang.Exception
```

Elimina un volumen de la lista de volúmenes.

Parametros:

volId - El identificador del volumen.

Lanza:

java.lang.Exception - Si se produce algún error al eliminar.

getCreationDate

```
public java.util.Date getCreationDate()
```

Obtiene la fecha de creación de la lista de volúmenes.

Retorno:

La fecha mencionada.

getCreatorId

```
public int getCreatorId()
```

Obtiene el identificador del usuario que ha creado la lista de volúmenes.

Retorno:

El identificador mencionado.

getFlags

```
public int getFlags()
```

Obtiene los flags de la lista de volúmenes.

Retorno:

El dato mencionado.

getId

```
public int getId()
```

Obtiene el identificador de la lista de volúmenes

Retorno:

El identificador mencionado



getName

```
public java.lang.String getName()
```

Obtiene el nombre de la lista de volúmenes.

Retorno:

El nombre mencionado.

getRemarks

```
public java.lang.String getRemarks()
```

Obtiene los comentarios de la lista de volúmenes.

Retorno:

El dato mencionado.

getState

```
public int getState()
```

Obtiene el estado de la lista de volúmenes.

Retorno:

El dato mencionado.

getUpdateDate

```
public java.util.Date getUpdateDate()
```

Obtiene la fecha de actualización de la lista de volúmenes.

Retorno:

La fecha mencionada.

getUpdaterId

```
public int getUpdaterId()
```

Obtiene el identificador del usuario que ha actualizado la lista de volúmenes.

Retorno:

El identificador mencionado.



getVolumes

```
public Volumes getVolumes(int id)
```

Obtiene una lista de identificadores de volúmenes. No es necesario cargar el objeto antes.

Parametros:

id - Identificador de la lista de volúmenes.

Retorno:

La lista mencionada.

load

```
public void load(int listId)  
    throws java.lang.Exception
```

Carga una lista de volúmenes de invesDoc.

Parametros:

listId - Identificador de la lista.

Lanza:

java.lang.Exception - Si se produce algún error al leer la información de la lista.

setName

```
public void setName(java.lang.String name)
```

Establece el nombre de la lista de volúmenes.

Parametros:

name - Nombre de la lista.

setRemarks

```
public void setRemarks(java.lang.String remarks)
```

Establece los comentarios de la lista de volúmenes.

Parametros:

remarks - Los comentarios de la lista.

setState

```
public void setState(int state)
```

Establece el estado de la lista de volúmenes.

Parametros:

state - El estado de la lista.



store

```
public void store()
    throws java.lang.Exception
```

Guarda la lista. Se utiliza tanto para inserciones como para actualizaciones.

Lanza:

java.lang.Exception - Si se produce algún error al guardar. Por ejemplo, la lista ya existe.

toString

```
public java.lang.String toString()
```

Muestra una representación de los valores de la clase en formato XML.

Retorno:

La representación mencionada.

Reemplaza:

toString in class java.lang.Object

toXML

```
public java.lang.String toXML()
```

Obtiene la información de la lista de volúmenes en formato XML.

Retorno:

La información mencionada.

ieci.tecdoc.idoc.admin.api.volume

Class VolumeLists

```
java.lang.Object
|
+--ieci.tecdoc.idoc.admin.api.volume.VolumeLists
```

< [Constructores](#) > < [Metodos](#) >

```
public class VolumeLists
    extends java.lang.Object
```

Maneja la lista de la lista de volúmenes de invesDoc.

Constructores



VolumeLists

```
public VolumeLists()
```

Construye un objeto de la clase.

Metodos

count

```
public int count()
```

Devuelve el número de lista de volúmenes.

Retorno:

El número de lista de volúmenes mencionado.

getVolumeList

```
public VolumeList getVolumeList(int index)
```

Devuelve una lista de volúmenes de la lista.

Parametros:

index - Índice de la lista de volúmenes que se desea recuperar.

Retorno:

La lista de volúmenes mencionada.

load

```
public void load()  
    throws java.lang.Exception
```

Carga la lista de volúmenes con su información básica.

Lanza:

java.lang.Exception - Si se produce algún error en la carga de las listas de volúmenes.

toString

```
public java.lang.String toString()
```

Muestra una representación de los valores de la clase en formato XML.

Retorno:

La representación mencionada.

Reemplaza:

toString in class java.lang.Object



toXML

```
public java.lang.String toXML()
```

Obtiene la información de la lista de la lista de volúmenes en formato XML.

Retorno:

La lista mencionada.

ieci.tecdoc.idoc.admin.api.volume

Class Volumes

```
java.lang.Object
|
+--ieci.tecdoc.idoc.admin.api.volume.Volumes
```

< [Constructores](#) > < [Metodos](#) >

```
public class Volumes
extends java.lang.Object
```

Maneja la lista de volúmenes de invesDoc.

Constructores

Volumes

```
public Volumes()
```

Construye un objeto de la clase.

Metodos

add

```
public void add(ieci.tecdoc.idoc.admin.internal.VolumeImpl volume)
```

Añade un volumen a la lista.

Parametros:

volume - Volumen que se quiere añadir.



count

```
public int count()
```

Devuelve el número de volúmenes.

Retorno:

El número de volúmenes mencionado.

getVolume

```
public Volume getVolume(int index)
```

Devuelve un volumen de la lista.

Parametros:

index - Índice del volumen que se desea recuperar.

Retorno:

El volumen mencionado.

loadByRep

```
public void loadByRep(int repId)
    throws java.lang.Exception
```

Carga la lista de volúmenes de un repositorio con su información básica.

Lanza:

java.lang.Exception - Si se produce algún error en la carga de los volúmenes.

loadByVolumeList

```
public void loadByVolumeList(int listId)
    throws java.lang.Exception
```

Carga la lista de volúmenes de una lista de volúmenes con su información básica.

Lanza:

java.lang.Exception - Si se produce algún error en la carga de los volúmenes.

remove

```
public void remove(int volId)
```

Elimina un volumen de la lista.

Parametros:

volId - Identificador del volumen a eliminar.



toString

```
public java.lang.String toString()
```

Muestra una representación de los valores de la clase en formato XML.

Retorno:

La representación mencionada.

Reemplaza:

toString in class java.lang.Object

toXML

```
public java.lang.String toXML()
```

Obtiene la información de la lista de volúmenes en formato XML.

Retorno:

La lista de volúmenes mencionada.
