

COM/ProCom - API

Version 2.0

M-team. 2013

Introduction

Purpose

The purpose of this document is to describe the specifications of API of COM as exposed to the GED world.

There is 3 supported worlds :

- *Java*
- *RPG*
- *Microsoft .Net*

JadeR is a SOA (*Services Oriented Architecture*). So the API is exposed as services. The actual standard for multi-platform services is the *Web Services* technology. The API is thus exposed as a set of Web Services.

What's new with ProCom

- CatalogService is fully backward compatible.
- OrchestratorService is now available

Version history

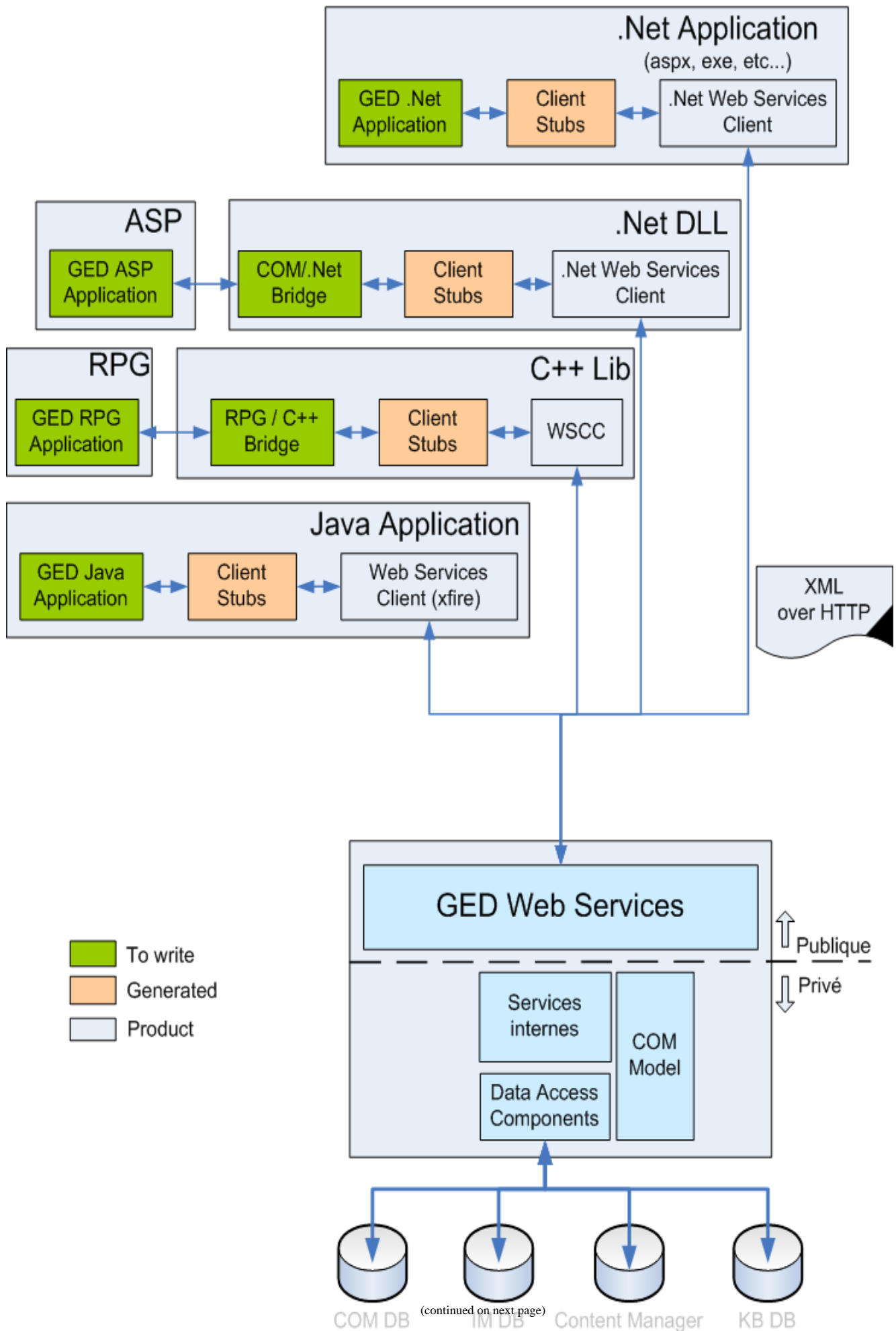
- *Version 2.0*
 - **Improvement** : documentation update
 - **Improvement** : regenerating documentation with new version of APIs
 - **Improvement** : small attachment can be inserted in createEventRequest(use imageBytes)
 - **Improvement** : documents created via WebServices have now their own source (value : 08)
 - **Improvement** : documents created via WebServices can specify the media type (values : paper, fax, ...)
 - **Bug** : for union only, it is now mandatory to specify thirdOrg when specify a member based on NISS
 - **Improvement** : if the workflow flag is set to true but there is no workflow associated with the docType a warning is send in CreateEventReply instead of an error
- *Version 1.9*
 - **Improvement** : parameters are now available to control zoom and toolbars in viewing an image
- *Version 1.8*
 - **New Service** : GedPrintService.printDocument. This service allows to generate and print a document asynchronously (using CSO component)
 - **New Service** : GedPrintService.printSynchronous. This service allows to generate and print a document synchronously (using CSO component)
 - **Improvement** : new attributes in GEDEvent : workflowStatus
- *Version 1.7*
 - **New Service** : CatalogService.findDocTypes. This service allows to search doctypes based on some criteria
 - **Improvement** : new attributes in GEDEvent : thirdExternalNumber, thirdNiss, thirdRiziv, thirdPartyType These are usefull information on the primary third
 - **Improvement** : new attributes in GEDEvent : secThirdExternalNumber, secThirdNiss, secThirdRiziv, secThirdPartyType These are usefull information on the secondary third
 - **Improvement** : New code type to specify a third : "ID". This is the DB id from party
 - **Improvement** : In CatalogService.updateEvent, you can now change the document's image. Image will be added in the versions of the document in DocCenter and will become the default one
 - **Improvement** : You can search on folder reference in CatalogService.findByParameters. A new attribute has been added to FindByParametersRequest : folderRef
 - **Bug** : updateEvent with a new folderRef is now possible

(continued on next page)

- **Bug** : findByParameters on a folderType is now possible
- *Version 1.6*
 - New Service : createEvent is now accessible on AS/400 (GBP547)
 - Improvement : new attribute in CreateEventRequest : imageCatalogKey. Binary content, text notes and graphical notes from the specified event will be copied to the new event
 - Improvement : new attribute in GEDEvent : dcId. It is the internal ID of DocCenter
- *Version 1.5*
 - New Service : OrchestratorWebService.assignTask
 - Improvement : new attributes in GEDEvent : scanBatchNumberYear, secThirdId, secThirdName, secThirdFName, noteLog, scannerId, thirdManagementCenter, freeField
- *Version 1.4*
 - BUG : service CatalogService.getAllDocTypes now returns the doctype's labels
- *Version 1.3*
 - New Service : CatalogService.getEventByScanBatchNumber
 - New Service : CatalogService.getEventByKey

Architecture

The architecture is as follows :



(continued on next page)

Web services are based on files called WSDLs (*Web Services Definition Language*), which are XML files containing all the information relating to services that are available at a particular location on the internet. At their simplest level, WSDLs describe request/response message pairs in detail, and contain everything relevant to the service.

A program must be used to turn the WSDL into a suite of C++/.Net/Java stubs and data objects that you can call and pass information to, and that request information from the server and then wait for the corresponding reply before passing the response data objects back to the client. The stubs hide the internet communication from the application writer. All you need to know is the name of the service, the method it contains, and the structure of any data objects that are passed.

JADE 2 provides WSDL files for the services that are exposed as Web Services. These files are located in this distribution in the following directory : *dist/wsdl*. Moreover, the generation of stubs in .Net is a built in feature of .Net. For RPG, IBM provides a product that can generate stubs in C++ that can be used by RPG programs. And for Java, libraries exists that can generate stubs (here we use *xfire*).

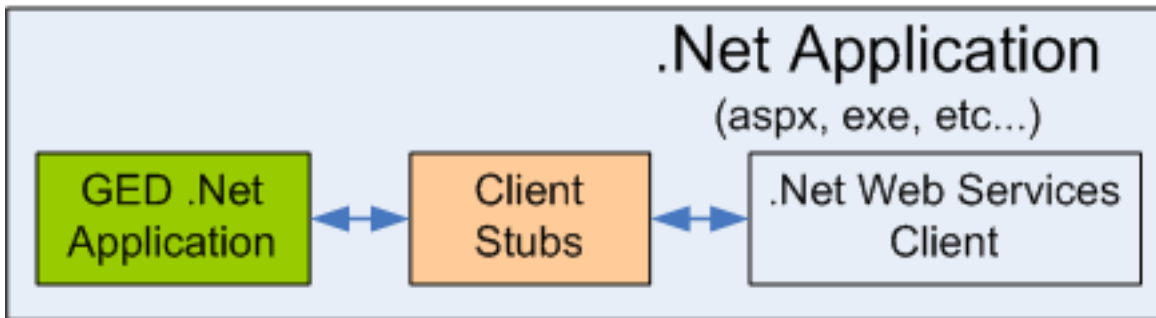
How to use this documentation

This documentation is actually the java documentation of the services. But the following sections will explain how to interpret it in *RPG*, in *Microsoft .Net* and of course in *Java*

The following will introduce the binding in the 3 worlds.

Use API from Microsoft .Net

The following shows the architecture.

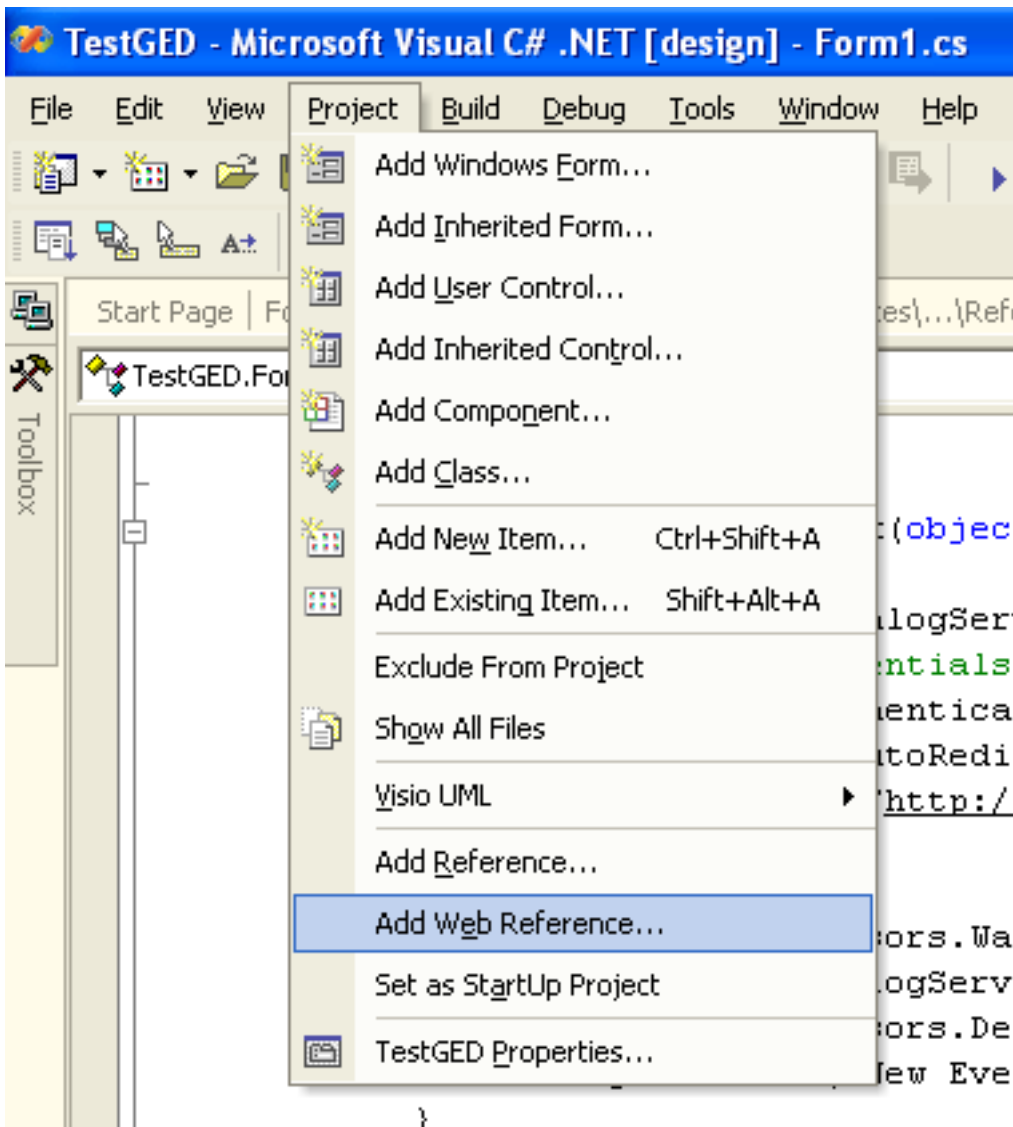


There is two ways to use the APIs from .Net : the manual way or the COMAPI way.

Manual Way

In the manual way, you configure everything yourself : you generate the stubs and then you use the services from the stubs. To generate the stubs, in *Visual Studio .Net*, add a *Web Reference* :

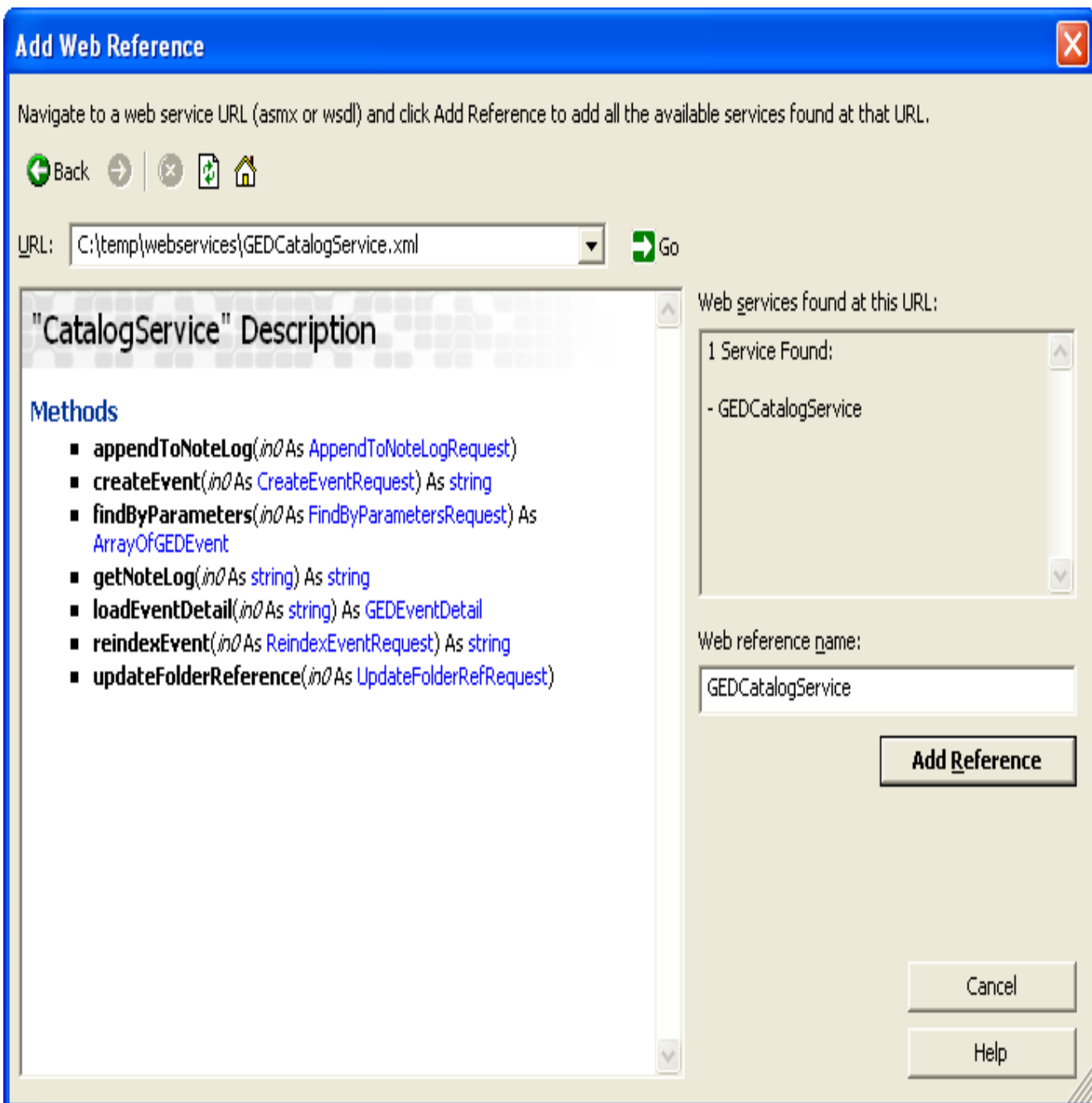
Then locate the WSDL file on your system (*GEDCatalogService.xml*) :



Now you can use the services.

COMAPI Way

However, the simplest way is to use the COMAPI that is provided to you. In this API all the stubs are already generated. Moreover, you have helpers that help you retrieve automatically the connection parameters. Here is an example :



```

ServiceLocator locator = new ServiceLocator("airbus", "561");
CatalogService service = locator.getCatalogService("JADE561", "XXXXXXX", "fr");
try
{
    ReindexEventRequest rqst = new ReindexEventRequest();
    rqst.eventKey = "561-2005-561-0123456789";
    rqst.newDocType = "K00011";
    rqst.newThirdOrg = "561";
    rqst.newThirdCodeType = "EXID";
    rqst.newThirdCodeValue = "123456789";
    string key = service.reindexEvent(rqst);
    MessageBox.Show("New Event : " + key);
}

```



```

catch(System.Web.Services.Protocols.SoapException exception)
{
    COMGEException ex = (COMGEException)COMGEException.getException(exception);
    MessageBox.Show("Code = " + ex.errorCode
        + "\nMessage = " + ex.errorMessage
        + "\nStacktrace = " + ex.errorStackTrace);
}
catch(System.Net.WebException ex)
{
    this.Cursor = Cursors.Default;
    MessageBox.Show("Message=" + ex.Message, "ERROR",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
}

```

The `ServiceLocator` class is used to locate the web services on the system you try to connect to. For instance, it retrieves the active version number on the system. On the example, we want to locate the active version on "airbus" for the "561" environment. The `serviceLocator` provides a method that returns a class that implements all the *COM Catalog services* (i.e. `CatalogService` service = `locator.getCatalogService("JADE561","XXXXXXX","fr")`).

The *COMGEException* class is used to represent an error in the services. It can be initialized from a *SoapException* and provides the following fields :

- **errorCode** : the error code returned by the API
- **errorMessage** : the error message returned by the API
- **errorStackTrace** : technical information concerning the error to send to the Help Desk when there is a problem

How to install the COMAPI DLL

You can find this DLL in this distribution. In the latter, you will find :

- A *docs* directory containing this document
- A *sources* directory containing the sources of the COMAPI DLL
- An *example* directory containing two examples :
 - A simple example in C# : *simple.zip*
 - A Visual Basic script that retrieves the connection information (note that to use this you have to register the DLL see below)
- A *dist/.net* directory containing the following files:
 - *COMAPI.dll* : to use it from .Net, copy it in the path of your application
 - *COMAPI.pdb* : this file contain debug information about the DLL and must be the same directory as the dll.

So to install the DLL for a usage from .Net, you only have to copy it in the path of your application. But if you want to use it from ASP, you have to register it. Look at the *register.cmd* in the *dist* directory.

Use API from ASP

In order to use it from ASP, it is necessary to create COM objects (*Component Object Model* not the Communication project). Fortunately, it is possible to create COM objects in .Net. So we can create COM objects in .Net that use the .Net generated stubs (see above) to communicate with JADE 2.

To do this, the following steps are mandatory :

- Generate a key : `sn -k c:\temp\mycom.snk` for .Net strong name
- Create a class Library project : `MyCOMAPI`
- Modify the *AssemblyInfo.cs* :

```
[assembly: AssemblyTitle("MyCOMAPI")]
[assembly: AssemblyDescription("MyCOM API")]
[assembly: AssemblyConfiguration("")]
[assembly: AssemblyCompany("XXX")]
[assembly: AssemblyProduct("MyCOM")]
[assembly: AssemblyCopyright("XXX")]
[assembly: AssemblyTrademark("")]
[assembly: AssemblyCulture("")]
[assembly: AssemblyDelaySign(false)]
[assembly: AssemblyKeyFile("c:\\temp\\mycom.snk")]
[assembly: AssemblyKeyName("myCom")]
[assembly: System.Runtime.InteropServices.ComVisible(true)]
```

- Create a class : *CatalogService* (necessary ?)
- Add the following attribute to this class : *[System.Runtime.InteropServices.ComVisible(true)]*
- Implement the class (delegate the web services)
- Build you solution
- Register the dll : *regasm COMAPI.dll*
- Put your dll in the GAC : *gacutil /i comapi.dll /f*
- Now you can use it from ASP :
 - *set o = createobject("comapi.CatalogService")*
 - *msgbox o.reindexEvent("toto")*

Note that the ServiceLocator is a COM object so you can use it to retrieve connection informations :

```
dim locator
set locator = createobject("COMAPI.ServiceLocator")
call locator.init("airbus","561")

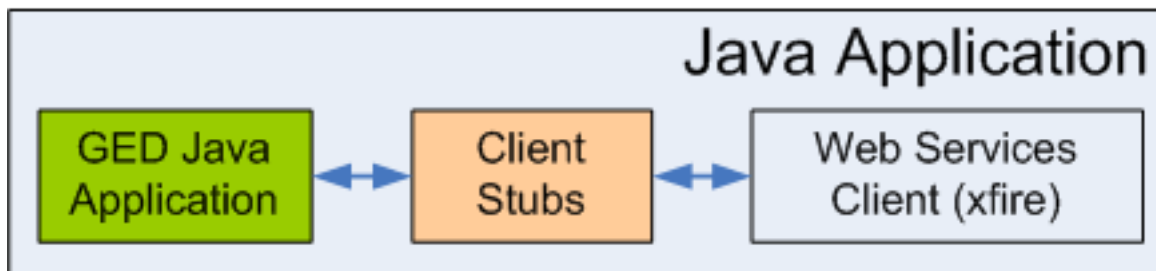
dim url
url = locator.getServiceURL("GEDCatalogService","JADE561","XXXXXXX","fr")
```

How to install the COMAPI DLL

See the same section in the *.Net Section*

Use API from Java

The following shows the architecture.



A jar file is provided to you : *COMAPI.jar*. In this API all the stubs are already generated. Moreover, you have helpers that help you retrieve automatically the connection parameters. Here is an example :

```
try
{
    ServiceLocator locator = new ServiceLocator("airbus","561");
    CatalogService service = locator.getCatalogService("JADE561","XXXXXXX","fr");
    ReindexEventRequest rqst = new ReindexEventRequest();
    rqst.setEventKey("561-2005-561-0123456789");
    rqst.setNewDocType("K00011");
    rqst.setNewThirdOrg("561");
    rqst.setNewThirdCodeType("EXID");
    rqst.setNewThirdCodeValue("123456789");
    String key = service.reindexEvent(rqst);
    System.out.println("New Event : " + key);
}
catch(COMGEException e)
{
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMessage());
    System.out.println(e.getErrorStackTrace());
    fail(e.getErrorMessage());
}
```

The *ServiceLocator* class is used to locate the web services on the system you try to connect to. For instance, it retrieves the active version number on the system. On the example, we want to locate the active version on "airbus" for the "561" environment. The *serviceLocator* provides a method that returns a class that implements all the *COM Catalog services* (i.e. *CatalogService* *service* = *locator.getCatalogService*("JADE561","XXXXXXX","fr")).

The *COMGEException* class is used to represent an error in the services. It can be initialized from a *SoapException* and provides the following fields :

- **errorCode** : the error code returned by the API
- **errorMessage** : the error message returned by the API
- **errorStackTrace** : technical information concerning the error to send to the Help Desk when there is a problem

How to install the COMAPI.jar

You can find this jar file on the IFS of Airbus at the following address : TODO

At this location, you will find :

- A *docs* directory containing this document
- A *sources* directory containing the sources of the *COMAPI.jar*
- An *example* directory containing examples
- A *dist* directory containing the following files:
 - *COMAPI.jar* : the jar file to use in your classpath
 - *libs* directory : this contains all the libraries used by the *COMAPI*. All these JAR files must be in the classpath

So to install the COM API for a usage from Java, you only have to copy *COMAPI.jar* (and all jar files of *libs* directory) in the classpath of your application.

Use API from RPG

M-team provides the following AS/400 programs :

- **CBP536** : This program reindex a document in the catalog. The parameters are :
 - *Param. 1* : sever on which runs Jade2/R (40 positions alpha - fill with blancs to obtain an exact size of 40)
 - *Param. 2* : environment (3 positions alpha)
 - *Param. 3* : TCP port (5 positions alpha). Note that in production it is always 80
 - *Param. 4* : name of a valid Jade II user (10 positions alpha)
 - *Param. 5* : password (10 positions alpha)
 - *Param. 6* : language (2 positions: FR, NL ou DE)
 - *Param. 7* : catalog key of the document (23 positions alpha)
 - *Param. 8* : new document type (6 positions alpha)
 - *Param. 9* : the party code type (4 positions alpha). Possible values:
 - *UID* : Unique ID
 - *EXID* : External number
 - *NISS* : NISS
 - *Param. 10* : party code (13 positions alpha)
 - *Param. 11* : party organization
- **GBP547** : This program creates a document in the catalog. The parameters are :
 - *HOST (IN 255 alpha)* : host
 - *ENV (IN 3 alpha)* : Environment du catalogage
 - *PORTA (IN 5 unsigned 0 decimal)* : Port
 - *USER (IN 10 alpha)* : Utilisateur reconnu dans DocCenter et qui a l'autorisation de cataloguer
 - *PASSWORD (IN 10 alpha)* : Mot de passe de cet utilisateur
 - *LANG (IN 2 alpha)* : Langue de l'utilisateur
 - *DOCTYPE (IN 8 alpha)* : Type de document
 - *SCNBYEAR (IN 4 alpha)* : Année
 - *SCANNBR (IN 13 alpha)* : Numéro d'archivage GED
 - *THIRDORG (IN 3 alpha)* : Organisation du tiers
 - *THIRDCDETY (IN 5 alpha)* : Type de numéro du tiers
 - *THIRDCDVAL (IN 13 alpha)* : Numéro du tiers
 - *THIRDPARTY (IN 50 alpha)* : Type de tiers
 - *SECCORG (IN 3 alpha)* : Organisation du tiers secondaire
 - *SECCDETY (IN 5 alpha)* : Type de numéro du tiers secondaire
 - *SECCDEVAL (IN 13 alpha)* : Numéro du tiers secondaire
 - *SECPARTYP (IN 50 alpha)* :
 - *DOCLG (IN 2 alpha)* : Langue du document
 - *FOLDERTYP (IN 2 alpha)* : Type de dossier
 - *FOLDEREF (IN 20 alpha)* : Réfrece du dossier
 - *NOTELOG (IN 2500 alpha)* : Note
 - *IMAGE (IN 255 alpha)* : Chemin ou se trouve le fichier de l'image
 - *IMGMIMETYP (IN 50 alpha)* : Type de fichier
 - *IMAGEKEY (IN 13 alpha)* : ID de l'image
 - *SCANNERID (IN 21 alpha)* : ID du scanner
 - *THIRDC TRE (IN 10 alpha)* : Centre de gestion
 - *WRKFLAG (IN 5 alpha)* : Flag pour le workflow ('TRUE ' ou 'FALSE')
 - *FREEFIELD (IN 20000 alpha)* : Champs libre
 - *ERROR (OUT 512 alpha)* : Message d'erreur
 - *NEWKEY (OUT 23 alpha)* : Clef de catalogage du document
 - *ERRORCDE (OUT 2 alpha)* : Code Erreur
 - *BUFLEN (OUT 10 INTEGER 0 decimal)* : longueur du buffer de communication avec Jade
 - *BUFLEN (OUT 65535 alpha)* : buffer de communication avec Jade
 - *STACKTRACE (OUT 1024 alpha)* : Message d'erreur
 - *RCVDATE (IN 20 alpha)* : Date de catalogage

Display an image

The images can be displayed in a browser with the following URL :

`http://<server>/<contextPath>/remoting/<environment>/imageView`

where

- `<server>` is the jadeII/jadeR server
- `<contextPath>` is the context path of jadeII/jadeR
- `<environment>` is the environment

with the following parameters :

- *key* : indique la clef de catalogage (Obligatoire).
- *format* : indique si on veut transformer l'image en pdf. Valeurs possible : pdf
- *generate* : indique si on veut générer l'image si elle n'existe pas (pour IM et CSO). Valeurs possibles : yes
- *v1* : indique si on veut visualiser avec viewone. Valeurs possibles : yes
- *basicauth* : permet de by-passer l'écran de login. Valeurs : user:password encodé en base64
- *c_tb_img (true / false)* : Minimise la barre d'outils relative aux contrôles de la vue lors de l'ouverture de ViewOne
- *c_tb_view (true / false)* : Minimise la barre d'outils relative aux contrôles de la vue lors de l'ouverture de ViewOne
- *c_tb_ann (true / false)* : Minimise la barre d'outils relative aux contrôles des annotations lors de l'ouverture de ViewOne
- *h_tb_ann (true / false)* : Masque la barre d'outils relative aux contrôles des annotations lors de l'ouverture de ViewOne
- *zoom (0.01 ... 0.5 ... 5.0)* : Indique le pourcentage d'affichage du document, la valeur par défaut étant de 50%
- *notes_enabled (true / false)* : Indique si les notes DocCenter sont affichées en bas de l'écran de visualisation du document, en dehors de ViewOne
- *ann_enabled (true / false)* : Indique si les annotations ViewOne sont affichées. Si les annotations sont désactivées, elles ne seront pas imprimées.
- *printer (<nom_imprimante>)* : Permet de forcer l'impression vers une imprimante spécifique. Si ce paramètre est défini, l'impression est lancée directement sans passer par le dialogue d'impression si la touche P est pressée ou si l'utilisateur clique sur le bouton d'impression.
- *Scale(best / ftoh / ftow)* : Permet de définir la mise à l'échelle du document affiché. La valeur « fto » permet un zoom sur la largeur tandis que « fto » permet un zoom sur la hauteur. Le paramètre par défaut « best » permet de visionner l'entièreté du document à l'écran.

Package

be.gfdi.com.common.services.ged

This package contains all the services for the *GED*.

It contains the following :

- **CatalogService** contains all the services related to the Catalog :
 - **createEvent** : to index a document
 - **reindexEvent** : to reindex a document
 - **findByParameters** : to find events
 - **loadEventDetail** : to find events
 - **updateFolderReference** : to update a folder reference associated to an event
 - **getAllDocTypes** : to retrieve all doctypes defined in DocCenter
- **OrchestratorWebService** contains all the services related to the Orchestrator :
 - **dispatchEvent** : to start a new process instance
 - **completeTask** : to complete an existing task
 - **findEventRouting** : to retrieve event routing associated with a doctype
 - **updateEventRouting** : to update event routing associated with a doctype
- **GEDEvent** : data structure that represents an event
- **GEDEventDetail** : data structure that represents the details of the event
- **GEDFindParameters** : data structure that contains all the possible parameters used to query the catalog
- **ThirdCodeType** : contains the third code type
- **COMGEDEException** contains all the errors that can happen in the services.

be.gfdi.com.common.services.ged

Class AppendToNoteLogRequest

java.lang.Object

└-be.gfdi.com.common.services.ged.AppendToNoteLogRequest

public class **AppendToNoteLogRequest**
extends Object

Constructor Summary

public	AppendToNoteLogRequest()
--------	--

Method Summary

String	getContent() get the content to add to the notelog
String	getKey() get the key of the event for which the notelog is updated
void	setContent(String content) set the content to add to the notelog
void	setKey(String key) set the key of the event for which the notelog is updated

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

AppendToNoteLogRequest

public **AppendToNoteLogRequest()**

Methods

getContent

public String **getContent()**

get the content to add to the notelog

Returns:

the content to add to the notelog

setContent

```
public void setContent(String content)
```

set the content to add to the notelog

Parameters:

content - the content to add to the notelog

getKey

```
public String getKey()
```

get the key of the event for which the notelog is updated

Returns:

the key of the event for which the notelog is updated

setKey

```
public void setKey(String key)
```

set the key of the event for which the notelog is updated

Parameters:

key - the key of the event for which the notelog is updated

be.gfdi.com.common.services.ged

Class AssignTaskRequest

java.lang.Object

└--be.gfdi.com.common.services.ged.AssignTaskRequest

All Implemented Interfaces:

Serializable

public class **AssignTaskRequest**
extends Object
implements Serializable

Constructor Summary

public	AssignTaskRequest()
--------	-------------------------------------

Method Summary

long	getTaskId()
String	getUserId()
void	setTaskId(long taskId)
void	setUserId(String userId)

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

AssignTaskRequest

public **AssignTaskRequest()**

Methods

getTaskId

public long **getTaskId()**

(continued from last page)

setTaskId

```
public void setTaskId(long taskId)
```

getUserId

```
public String getUserId()
```

setUserId

```
public void setUserId(String userId)
```

be.gfdi.com.common.services.ged Interface CatalogService

public interface **CatalogService**
extends

This contains all the services related to the catalog : find, consult, reindex, etc...

Method Summary

void	appendToNoteLog (AppendToNoteLogRequest request) This service appends a content to the notelog
String	createEvent (CreateEventRequest request) This service index a new event in the catalog
CreateEventReply	createEventWithWarning (CreateEventRequest request) This service index a new event in the catalog and send a possible warning messages
GEDEvent[]	findByParameters (FindByParametersRequest request) This service finds events that corresponds to the specified parameters
GEDDocType[]	findDocTypes (FindDocTypesRequest rqst)
GEDDocType[]	getAllDocTypes () Retrieve all the existing DocType as an array of GEDDocType objects.
GEDFolderType[]	getAllFolderTypes (String[] muts) Retrieve all the existing FolderType as an array of GEDFolderType objects.
GEDEvent	getEventByKey (String key) This service loads an event
GEDEvent	getEventByScanBatchNumber (String scanBatchNumber) This service loads an event based on the "scanbatchnumber".
String	getNoteLog (String key) This service returns the content of the notelog associated to this event
Boolean	isDocumentValidToBeImported (String filePath, Boolean legalDocument) Validates a document to be imported into the Catalog
GEDEventDetail	loadEventDetail (String key) This service load the details of an event (such as the fields values for a CSO)
String	reindexEvent (ReindexEventRequest request) This service "reindex" an event.
String	reindexWithoutNewLine (ReindexEventRequest request) This service "reindexWithoutNewLine" an event.
CreateEventReply	updateEventWithWarning (CreateEventRequest request) This service update an old event in the catalog and send a possible warning messages

void	updateFolderReference (UpdateFolderRefRequest request) This service add (or update) a reference to a folder for a specified event If the event has already a reference of the type specified in argument ("newFolderType") the reference is updated.
void	updateStatusForDelete (String eventKey) updateStatusForDelete method is used to update the status of an existing Event to "X" .
void	updateStatusForDouble (String eventKey) updateStatusForDouble method is used to update the status of an existing Event to "D" .

Methods

reindexEvent

public String **reindexEvent**([ReindexEventRequest](#) request)
throws [COMGEException](#)

This service "reindex" an event. The only supported event class is 'MME' the existing event is updated and a new event is created.

Parameters:

request - request containing all necessary parameters, see ([ReindexEventRequest](#))

Returns:

the key of the new event created (i.e. "561-2006-561-0123456789")

Errors :

- [COMGEException.ERROR_DOCTYPE_DOES_NOT_EXIST_KEY](#)
- [COMGEException.ERROR_DOCTYPE_NOT_MME_KEY](#)
- [COMGEException.ERROR_EVENT_DOES_NOT_EXIST_KEY](#)
- [COMGEException.ERROR_EVENT_NOT_MME_KEY](#)
- [COMGEException.ERROR_INVALID_KEY](#)
- [COMGEException.ERROR_THIRD_CODETYPE_INVALID_KEY](#)
- [COMGEException.ERROR_DIRECTION_DOES_NOT_EXIST](#)
- [COMGEException.ERROR_THIRD_DOES_NOT_EXIST_KEY](#)
- [COMGEException.ERROR_THIRD_CODE_VALUE_FOR_CODETYPE_INVALID_KEY](#)
- [COMGEException.ERROR_THIRD_ORG_INVALID_KEY](#)
- [COMGEException.ERROR_UNKNOWN_KEY](#)

reindexWithoutNewLine

public String **reindexWithoutNewLine**([ReindexEventRequest](#) request)
throws [COMGEException](#)

This service "reindexWithoutNewLine" an event. The only supported event class is 'MME' the existing event is just updated and no new event is created.

Parameters:

request - request containing all necessary parameters, see ([ReindexEventRequest](#))

Returns:

(continued from last page)

the key of the event updated (i.e. "561-2006-561-0123456789")

Errors :

- [COMGEException.ERROR_DOCTYPE_DOES_NOT_EXIST_KEY](#)
- [COMGEException.ERROR_DOCTYPE_NOT_MME_KEY](#)
- [COMGEException.ERROR_EVENT_DOES_NOT_EXIST_KEY](#)
- [COMGEException.ERROR_EVENT_NOT_MME_KEY](#)
- [COMGEException.ERROR_INVALID_KEY](#)
- [COMGEException.ERROR_THIRD_CODETYPE_INVALID_KEY](#)
- [COMGEException.ERROR_DIRECTION_DOES_NOT_EXIST](#)
- [COMGEException.ERROR_THIRD_DOES_NOT_EXIST_KEY](#)
- [COMGEException.ERROR_THIRD_CODE_VALUE_FOR_CODETYPE_INVALID_KEY](#)
- [COMGEException.ERROR_THIRD_ORG_INVALID_KEY](#)
- [COMGEException.ERROR_UNKNOWN_KEY](#)

findByParameters

```
public GEDEvent\[\] findByParameters(FindByParametersRequest request)
    throws COMGEException
```

This service finds events that corresponds to the specified parameters

Parameters:

request - request containing all necessary parameters, see ([FindByParametersRequest](#))

Returns:

list of [GEDEvent](#)

Errors :

- [COMGEException.ERROR_THIRD_CODETYPE_INVALID_KEY](#)
- [COMGEException.ERROR_DIRECTION_DOES_NOT_EXIST](#)
- [COMGEException.ERROR_THIRD_DOES_NOT_EXIST_KEY](#)
- [COMGEException.ERROR_THIRD_CODE_VALUE_FOR_CODETYPE_INVALID_KEY](#)
- [COMGEException.ERROR_THIRD_ORG_INVALID_KEY](#)
- [COMGEException.ERROR_DOCTYPE_DOES_NOT_EXIST_KEY](#)
- [COMGEException.ERROR_DOCTYPE_NOT_MME_KEY](#)
- [COMGEException.ERROR_DOCTYPESUBCLASS_DOES_NOT_EXIST](#)
- [COMGEException.ERROR_FOLDERTYPE_DOES_NOT_EXIST](#)
- [COMGEException.ERROR_DOCLANGUAGE_DOES_NOT_EXIST](#)
- [COMGEException.ERROR_UNKNOWN_KEY](#)

loadEventDetail

```
public GEDEventDetail loadEventDetail(String key)
    throws COMGEException
```

This service load the details of an event (such as the fields values for a CSO)

Parameters:

key - event key

Returns:

(continued from last page)

event details

Errors :

- [COMGEException.ERROR_THIRD_ORGANIZATION_CAN_NOT_BE_NULL](#)
- [COMGEException.ERROR_UNKNOWN_KEY](#)

getEventByKey

```
public GEDEvent getEventByKey(String key)  
    throws COMGEException
```

This service loads an event

Parameters:

key - event key

Returns:

[GEDEvent](#)

Errors :

- [COMGEException.ERROR_UNKNOWN_KEY](#)

getEventByScanBatchNumber

```
public GEDEvent getEventByScanBatchNumber(String scanBatchNumber)  
    throws COMGEException
```

This service loads an event based on the "scanbatchnumber". The format of the scanbatch number is either year + '-' + number of year + number.

Parameters:

scanBatchNumber - ged identification number

Returns:

[GEDEvent](#)

Throws:

[COMGEException](#)

updateFolderReference

```
public void updateFolderReference(UpdateFolderRefRequest request)  
    throws COMGEException
```

This service add (or update) a reference to a folder for a specified event If the event has already a reference of the type specified in argument ("newFolderType") the reference is updated. Otherwise a new reference is added to the document

Parameters:

(continued from last page)

request - request containing all necessary parameters, see ([UpdateFolderRefRequest](#))

Errors :

- [COMGEException.ERROR_EVENT_DOES_NOT_EXIST_KEY](#)
- [COMGEException.ERROR_EVENT_NOT_MME_KEY](#)
- [COMGEException.ERROR_INVALID_KEY](#)
- [COMGEException.ERROR_FOLDERTYPE_DOES_NOT_EXIST](#)
- [COMGEException.ERROR_FOLDER_REFERENCE_INVALID_KEY](#)
- [COMGEException.ERROR_UNKNOWN_KEY](#)

createEvent

```
public String createEvent(CreateEventRequest request)
    throws COMGEException
```

This service index a new event in the catalog

Parameters:

request - request containing all necessary parameters

Returns:

key generated for the new event

Errors :

- [COMGEException.ERROR_THIRD_CODETYPE_INVALID_KEY](#)
- [COMGEException.ERROR_DIRECTION_DOES_NOT_EXIST](#)
- [COMGEException.ERROR_THIRD_DOES_NOT_EXIST_KEY](#)
- [COMGEException.ERROR_THIRD_CODE_VALUE_FOR_CODETYPE_INVALID_KEY](#)
- [COMGEException.ERROR_THIRD_ORG_INVALID_KEY](#)
- [COMGEException.ERROR_DOCTYPE_DOES_NOT_EXIST_KEY](#)
- [COMGEException.ERROR_DOCTYPE_NOT_MME_KEY](#)
- [COMGEException.ERROR_FOLDERTYPE_DOES_NOT_EXIST](#)
- [COMGEException.ERROR_FOLDER_REFERENCE_INVALID_KEY](#)
- [COMGEException.ERROR_IMAGE_KEY_INVALID](#)
- [COMGEException.ERROR_SCANBATCH_NUMBER_INVALID_KEY](#)
- [COMGEException.ERROR_SCANNER_ID_INVALID_KEY](#)
- [COMGEException.ERROR_DOCLANGUAGE_DOES_NOT_EXIST](#)
- [COMGEException.ERROR_UNKNOWN_KEY](#)

createEventWithWarning

```
public CreateEventReply createEventWithWarning(CreateEventRequest request)
    throws COMGEException
```

This service index a new event in the catalog and send a possible warning messages

Parameters:

request - request containing all necessary parameters

Returns:

(continued from last page)

CreateEventReply that contains a key generated for the new event and warning messages

Errors :

- [COMGEException.ERROR_THIRD_CODETYPE_INVALID_KEY](#)
- [COMGEException.ERROR_DIRECTION_DOES_NOT_EXIST](#)
- [COMGEException.ERROR_THIRD_DOES_NOT_EXIST_KEY](#)
- [COMGEException.ERROR_THIRD_CODE_VALUE_FOR_CODETYPE_INVALID_KEY](#)
- [COMGEException.ERROR_THIRD_ORG_INVALID_KEY](#)
- [COMGEException.ERROR_DOCTYPE_DOES_NOT_EXIST_KEY](#)
- [COMGEException.ERROR_DOCTYPE_NOT_MME_KEY](#)
- [COMGEException.ERROR_FOLDERTYPE_DOES_NOT_EXIST](#)
- [COMGEException.ERROR_FOLDER_REFERENCE_INVALID_KEY](#)
- [COMGEException.ERROR_IMAGE_KEY_INVALID](#)
- [COMGEException.ERROR_SCANBATCH_NUMBER_INVALID_KEY](#)
- [COMGEException.ERROR_SCANNER_ID_INVALID_KEY](#)
- [COMGEException.ERROR_DOCLANGUAGE_DOES_NOT_EXIST](#)
- [COMGEException.ERROR_DOCMEDIA_DOES_NOT_EXIST](#)
- [COMGEException.ERROR_DOCSOURCE_DOES_NOT_EXIST](#)
- [COMGEException.ERROR_UNKNOWN_KEY](#)

updateEventWithWarning

```
public CreateEventReply updateEventWithWarning(CreateEventRequest request)
    throws COMGEException
```

This service update an old event in the catalog and send a possible warning messages

Parameters:

request - request containing all necessary parameters

Returns:

CreateEventReply that contains event key and warning messages

Errors :

- [COMGEException.ERROR_THIRD_CODETYPE_INVALID_KEY](#)
- [COMGEException.ERROR_DIRECTION_DOES_NOT_EXIST](#)
- [COMGEException.ERROR_THIRD_DOES_NOT_EXIST_KEY](#)
- [COMGEException.ERROR_THIRD_CODE_VALUE_FOR_CODETYPE_INVALID_KEY](#)
- [COMGEException.ERROR_THIRD_ORG_INVALID_KEY](#)
- [COMGEException.ERROR_DOCTYPE_DOES_NOT_EXIST_KEY](#)
- [COMGEException.ERROR_DOCTYPE_NOT_MME_KEY](#)
- [COMGEException.ERROR_FOLDER_REFERENCE_INVALID_KEY](#)
- [COMGEException.ERROR_IMAGE_KEY_INVALID](#)
- [COMGEException.ERROR_SCANBATCH_NUMBER_INVALID_KEY](#)
- [COMGEException.ERROR_SCANNER_ID_INVALID_KEY](#)
- [COMGEException.ERROR_DOCLANGUAGE_DOES_NOT_EXIST](#)
- [COMGEException.ERROR_DOCMEDIA_DOES_NOT_EXIST](#)
- [COMGEException.ERROR_DOCSOURCE_DOES_NOT_EXIST](#)
- [COMGEException.ERROR_UNKNOWN_KEY](#)

getNoteLog

```
public String getNoteLog(String key)
    throws COMGEException
```

This service returns the content of the notelog associated to this event

(continued from last page)

Parameters:

key - event key (561-2006-561-0000001246 or 56120065610000001246)

Returns:

the content of the notelog associated to this event

appendToNoteLog

```
public void appendToNoteLog(AppendToNoteLogRequest request)
    throws COMGEDEException
```

This service appends a content to the notelog

Parameters:request - request containing all necessary parameters, see ([AppendToNoteLogRequest](#))

getAllFolderTypes

```
public GEDFolderType\[\] getAllFolderTypes(String[] muts)
```

Retrieve all the existing FolderType as an array of GEDFolderType objects.

Returns:

An array containing all the existing FolderType.

getAllDocTypes

```
public GEDDocType\[\] getAllDocTypes()
```

Retrieve all the existing DocType as an array of GEDDocType objects.

Returns:

An array containing all the existing DocType.

findDocTypes

```
public GEDDocType\[\] findDocTypes(FindDocTypesRequest rqst)
```

updateStatusForDelete

```
public void updateStatusForDelete(String eventKey)
    throws COMGEDEException
```

updateStatusForDelete method is used to update the status of an existing Event to "X" . with possibles warnings

Parameters:

eventKey - is String value of the key event if error return warning message

Errors :

- [COMGEDEException.ERROR_UNKNOWN_KEY](#)

(continued from last page)

updateStatusForDouble

```
public void updateStatusForDouble(String eventKey)  
    throws COMGEDEException
```

updateStatusForDouble method is used to update the status of an existing Event to "D" . with possibles warnings

Parameters:

eventKey - is the key event if error return warning message

Errors :

- [COMGEDEException.ERROR_UNKNOWN_KEY](#)

isDocumentValidToBeImported

```
public Boolean isDocumentValidToBeImported(String filePath,  
    Boolean legalDocument)  
    throws COMGEDEException
```

Validates a document to be imported into the Catalog

Parameters:

filePath - The document full file path

legalDocument - The legal nature of the document (true if legal)

Returns:

True if validation is ok

Throws:

[COMGEDEException](#) - If validation failed, with following error codes :

Errors :

- [COMGEDEException.ERROR_DOCUMENT_FORMAT_NOT_ALLOWED_FOR_LEGAL](#)
- [COMGEDEException.ERROR_DOCUMENT_FORMAT_NOT_ALLOWED_FOR_NON_LEGAL](#)
- [COMGEDEException.ERROR_DOCUMENT_PDF_MAX_SIZE_PER_PAGE_EXCEEDED](#)

be.gfdi.com.common.services.ged Class COMGEDEException

```

java.lang.Object
  |
  +- java.lang.Throwable
    |
    +- java.lang.Exception
      |
      +- be.gfdi.com.common.services.ged.COMGEDEException

```

All Implemented Interfaces:

Serializable

```

public class COMGEDEException
extends Exception

```

This this class represents the error generated by the services

Field Summary

public static final	<u>ERROR_ATTACHED_FILE_IS_EMPTY</u> The request attachment is empty Value: 51
public static final	<u>ERROR_ATTACHED_FILE_IS_TOO_LARGE</u> The request attachment has no name Value: 52
public static final	<u>ERROR_BATCH_SCAN_YEAR_NOT_VALID</u> The format of the scanbatchnumber is invalid Value: 40
public static final	<u>ERROR_CANVAS_STATUS_NOT_VALID</u> The canvas status is not valid Value: 37
public static final	<u>ERROR_CONTACT_INFO_TYPE_NOT_VALID</u> The contactinfo type is not valid Value: 38
public static final	<u>ERROR_DIRECTION_DOES_NOT_EXIST</u> event direction does not exist Value: 08
public static final	<u>ERROR_DOCLANGUAGE_DOES_NOT_EXIST</u> docLanguage does not exist Value: 11
public static final	<u>ERROR_DOCMEDIA_DOES_NOT_EXIST</u> The document media does not exist. Value: 48

public static final	ERROR_DOCSOURCE_DOES_NOT_EXIST The document source does not exist. Value: 49
public static final	ERROR_DOCTYPE_DOES_NOT_EXIST_KEY DocType does not exist Value: 04
public static final	ERROR_DOCTYPE_NOT_CSO_KEY DocType is not an 'CSO' Value: 41
public static final	ERROR_DOCTYPE_NOT_MME_KEY DocType is not an 'MME' Value: 05
public static final	ERROR_DOCTYPESUBCLASS_DOES_NOT_EXIST docTypeSubClass does not exist Value: 09
public static final	ERROR_DOCUMENT_DOES_NOT_EXIST_IN_DOCUMENTSDATA The document exist in the documentsdataact table but not in documentsinfo . Value: 32
public static final	ERROR_DOCUMENT_FORMAT_NOT_ALLOWED_FOR_LEGAL Imported document format is not supported for legal archiving Value: 44
public static final	ERROR_DOCUMENT_FORMAT_NOT_ALLOWED_FOR_NON_LEGAL Imported document format is not supported for non-legal archiving Value: 45
public static final	ERROR_DOCUMENT_PDF_MAX_SIZE_PER_PAGE_EXCEEDED Imported PDF document maximal size per page exceeded Value: 46
public static final	ERROR_ENTITY_IS_NOT_CORRECT The entity is not correct Value: 34
public static final	ERROR_ENTITY_IS_NOT_MANAGEMENT_CENTER The entity event is not a management center. Value: 23
public static final	ERROR_ENTITY_TYPE_NOT_VALID The Entity type is not valid type. Value: 36
public static final	ERROR_EVENT_DOES_NOT_EXIST_KEY an event is not found with the specified key Value: 02
public static final	ERROR_EVENT_IMAGE_EXIST Cannot add new image for this event.This event has already one imagekey . Value: 30

public static final	<u>ERROR_EVENT_NOT_MME_KEY</u> the event is not an 'MME' Value: 03
public static final	<u>ERROR_EXECUTION_CODE_NOT_VALID</u> The execution code is not valid Value: 33
public static final	<u>ERROR_FILE_EXTENSION_CODE_CM_DOES_NOT_EXIST</u> FileExtensionCodeCM does not exist Value: 21
public static final	<u>ERROR_FOLDER_REFERENCE_INVALID_KEY</u> The folder Reference value is invalid (the value must be less or equal to 20 char) Value: 14
public static final	<u>ERROR_FOLDERTYPE_DOES_NOT_EXIST</u> folderType does not exist Value: 10
public static final	<u>ERROR_GED_DEFAULT_USER_CODE_PARAMETER_NOT_EXIST</u> The code Parameter GEDDefaultUser for the image path is not correct. Value: 27
public static final	<u>ERROR_IMAGE_FILE_WITHOUT_EXTENSION_TYPE_NOT_POSSIBLE</u> The request must contain imageMimeType to import the imageFile Value: 20
public static final	<u>ERROR_IMAGE_KEY_INVALID</u> The imagekey value is invalid (the value must be equal to 13 char) Value: 15
public static final	<u>ERROR_IMAGE_PATH_SERVER_NOT_CORRECT</u> The server Name in the image path is not correct. Value: 26
public static final	<u>ERROR_IMPORTING_IMAGE_IN_CONTENT_MANGER</u> Error when importing in the content Manager. Value: 28
public static final	<u>ERROR_INVALID_KEY</u> event key is invalid Value: 01
public static final	<u>ERROR_LANGUAGE_NOT_VALID</u> The language is not valid Value: 39
public static final	<u>ERROR_MANAGEMENT_CENTER_DOES_NOT_EXIST</u> The Management center for this event Key does not exist. Value: 24
public static final	<u>ERROR_NO_PROCESS_DEFINED_FOR_THIS_DOCTYPE</u> No process exists for the selected document type. Value: 47

public static final	<u>ERROR_NOTLOG_NOT_EXIST_FOR_EVENT_KEY</u> The notLog for this event Key does not exist. Value: 22
public static final	<u>ERROR_ORG_FOR_THIS_ENTITY_ID_IS_NOT_CORRECT</u> Entity organisation is not correct for this entityId Value: 42
public static final	<u>ERROR_REQUEST_WITH_IMAGE_AND_IMAGEKEY_NOT_POSSIBLE</u> The request containing an imageFile and an imageKey is not possible in the same time Value: 19
public static final	<u>ERROR_SCANBATCH_NUMBER_INVALID_KEY</u> The scanBatchNbr value is invalid (the value must be less or equal 16 char) Value: 16
public static final	<u>ERROR_SCANNER_ID_INVALID_KEY</u> The scanId value is invalid (the value must be less or equal 16 char) Value: 17
public static final	<u>ERROR_SECONDARY_THIRD_TYPE_INVALID</u> The secondary third type value is invalid (possible values : MUTUALITYPERSON, PROVIDER, EMPLOYER and INSTITUTION) Value: 18
public static final	<u>ERROR_THIRD_CODE_VALUE_FOR_CODETYPE_INVALID_KEY</u> The code value for this code Type is invalid (the value must be 13 char for external and unique nbre and 11 for niss nbre) Value: 12
public static final	<u>ERROR_THIRD_CODETYPE_INVALID_KEY</u> third code type is invalid Value: 06
public static final	<u>ERROR_THIRD_DOES_NOT_EXIST_KEY</u> third does not exist Value: 07
public static final	<u>ERROR_THIRD_ORG_INVALID_KEY</u> The organization value is invalid (the value must be 3 char) Value: 13
public static final	<u>ERROR_THIRD_ORGANIZATION_CAN_NOT_BE_NULL</u> The Third Organization is required when the search is performed by Union. Value: 50
public static final	<u>ERROR_THIRD_PARTY_TYPE</u> The Third party type is not correct. Value: 31
public static final	<u>ERROR_TYPE_FOR_THIS_ENTITY_ID_IS_NOT_CORRECT</u> Entity Type is not correct for this entityId Value: 43
public static final	<u>ERROR_UNKNOWN_KEY</u> Unknown error Value: 99

public static final	ERROR_WORKFLOW_NAME_CAN_NOT_BE_NULL The Workflow name can't be null. Value: 25
public static final	IMAGE_DOES_NOT_EXIST_IN_CONTENT_MANGER Image Key does not exist in content manager. Value: 29
public static final	string The entity does not exist Value: 35

Constructor Summary

public	COMGEException()
public	COMGEException(String message)
public	COMGEException(String errorCode, String message)
public	COMGEException(String errorCode, String message, Throwable cause)

Method Summary

String	getErrorCode()
String	getErrorMessage()
String	getErrorStackTrace()
void	setErrorCode(String errorCode)
void	setErrorMessage(String errorMessage)
void	setErrorStackTrace(String errorStackTrace)

Methods inherited from class java.lang.Throwable

fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Fields

ERROR_INVALID_KEY

public static final java.lang.String **ERROR_INVALID_KEY**

(continued from last page)

event key is invalid
Constant value: **01**

ERROR_EVENT_DOES_NOT_EXIST_KEY

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

an event is not found with the specified key
Constant value: **02**

ERROR_EVENT_NOT_MME_KEY

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

the event is not an 'MME'
Constant value: **03**

ERROR_DOCTYPE_DOES_NOT_EXIST_KEY

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

DocType does not exist
Constant value: **04**

ERROR_DOCTYPE_NOT_MME_KEY

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

DocType is not an 'MME'
Constant value: **05**

ERROR_THIRD_CODETYPE_INVALID_KEY

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

third code type is invalid
Constant value: **06**

ERROR_THIRD_DOES_NOT_EXIST_KEY

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

third does not exist
Constant value: **07**

ERROR_DIRECTION_DOES_NOT_EXIST

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

event direction does not exist
Constant value: **08**

ERROR_DOCTYPESUBCLASS_DOES_NOT_EXIST

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

docTypeSubClass does not exist
Constant value: **09**

ERROR_FOLDERTYPE_DOES_NOT_EXIST

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

folderType does not exist
Constant value: **10**

ERROR_DOCLANGUAGE_DOES_NOT_EXIST

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

docLanguage does not exist
Constant value: **11**

ERROR_THIRD_CODE_VALUE_FOR_CODETYPE_INVALID_KEY

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

The code value for this code Type is invalid (the value must be 13 char for external and unique nbre and 11 for niss nbre)
Constant value: **12**

ERROR_THIRD_ORG_INVALID_KEY

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

The organization value is invalid (the value must be 3 char)
Constant value: **13**

ERROR_FOLDER_REFERENCE_INVALID_KEY

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

The folder Reference value is invalid (the value must be less or equal to 20 char)
Constant value: **14**

ERROR_IMAGE_KEY_INVALID

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

The imagekey value is invalid (the value must be equal to 13 char)
Constant value: **15**

ERROR_SCANBATCH_NUMBER_INVALID_KEY

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

The scanBatchNbr value is invalid (the value must be less or equal 16 char)
Constant value: **16**

ERROR_SCANNER_ID_INVALID_KEY

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

The scanId value is invalid (the value must be less or equal 16 char)
Constant value: **17**

(continued from last page)

ERROR_SECONDARY_THIRD_TYPE_INVALID

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

The secondary third type value is invalid (possible values : MUTUALITYPERSON, PROVIDER, EMPLOYER and INSTITUTION)

Constant value: **18**

ERROR_REQUEST_WITH_IMAGE_AND_IMAGEKEY_NOT_POSSIBLE

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

The request containing an imageFile and an imageKey is not possible in the same time

Constant value: **19**

ERROR_IMAGE_FILE_WITHOUT_EXTENSION_TYPE_NOT_POSSIBLE

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

The request must contain imageMimeType to import the imageFile

Constant value: **20**

ERROR_FILE_EXTENSION_CODE_CM_DOES_NOT_EXIST

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

FileExtensionCodeCM does not exist

Constant value: **21**

ERROR_NOTLOG_NOT_EXIST_FOR_EVENT_KEY

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

The notLog for this event Key does not exist.

Constant value: **22**

ERROR_ENTITY_IS_NOT_MANAGEMENT_CENTER

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

The entity event is not a management center.

Constant value: **23**

ERROR_MANAGEMENT_CENTER_DOES_NOT_EXIST

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

The Management center for this event Key does not exist.

Constant value: **24**

ERROR_WORKFLOW_NAME_CAN_NOT_BE_NULL

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

The Workflow name can't be null.

Constant value: **25**

(continued from last page)

ERROR_IMAGE_PATH_SERVER_NOT_CORRECT

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

The server Name in the image path is not correct.
Constant value: **26**

ERROR_GED_DEFAULT_USER_CODE_PARAMETER_NOT_EXIST

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

The code Parameter GEDDefaultUser for the image path is not correct.
Constant value: **27**

ERROR_IMPORTING_IMAGE_IN_CONTENT_MANGER

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

Error when importing in the content Manager.
Constant value: **28**

IMAGE_DOES_NOT_EXIST_IN_CONTENT_MANGER

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

Image Key does not exist in content manager.
Constant value: **29**

ERROR_EVENT_IMAGE_EXIST

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

Cannot add new image for this event.This event has already one imagekey .
Constant value: **30**

ERROR_THIRD_PARTY_TYPE

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

The Third party type is not correct.
Constant value: **31**

ERROR_DOCUMENT_DOES_NOT_EXIST_IN_DOCUMENTSDATA

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

The document exist in the documentsdatacrt table but not in documentsinfo .
Constant value: **32**

ERROR_EXECUTION_CODE_NOT_VALID

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

The execution code is not valid
Constant value: **33**

ERROR_ENTITY_IS_NOT_CORRECT

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

(continued from last page)

The entity is not correct
Constant value: **34**

string

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

The entity does not exist
Constant value: **35**

ERROR_ENTITY_TYPE_NOT_VALID

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

The Entity type is not valid type.
Constant value: **36**

ERROR_CANVAS_STATUS_NOT_VALID

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

The canvas status is not valid
Constant value: **37**

ERROR_CONTACT_INFO_TYPE_NOT_VALID

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

The contactinfo type is not valid
Constant value: **38**

ERROR_LANGUAGE_NOT_VALID

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

The language is not valid
Constant value: **39**

ERROR_BATCH_SCAN_YEAR_NOT_VALID

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

The format of the scanbatchnumber is invalid
Constant value: **40**

ERROR_DOCTYPE_NOT_CSO_KEY

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

DocType is not an 'CSO'
Constant value: **41**

ERROR_ORG_FOR_THIS_ENTITY_ID_IS_NOT_CORRECT

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

Entity organisation is not correct for this entityId
Constant value: **42**

ERROR_TYPE_FOR_THIS_ENTITY_ID_IS_NOT_CORRECT

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

Entity Type is not correct for this entityId
Constant value: **43**

ERROR_DOCUMENT_FORMAT_NOT_ALLOWED_FOR_LEGAL

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

Imported document format is not supported for legal archiving
Constant value: **44**

ERROR_DOCUMENT_FORMAT_NOT_ALLOWED_FOR_NON_LEGAL

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

Imported document format is not supported for non-legal archiving
Constant value: **45**

ERROR_DOCUMENT_PDF_MAX_SIZE_PER_PAGE_EXCEEDED

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

Imported PDF document maximal size per page exceeded
Constant value: **46**

ERROR_NO_PROCESS_DEFINED_FOR_THIS_DOCTYPE

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

No process exists for the selected document type. Message to display in batch import table is : "This doctype has no process associated"
Constant value: **47**

ERROR_DOCMEDIA_DOES_NOT_EXIST

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

The document media does not exist.
Constant value: **48**

ERROR_DOCSOURCE_DOES_NOT_EXIST

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

The document source does not exist.
Constant value: **49**

ERROR_THIRD_ORGANIZATION_CAN_NOT_BE_NULL

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

The Third Organization is required when the search is performed by Union.
Constant value: **50**

(continued from last page)

ERROR_ATTACHED_FILE_IS_EMPTY

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

The request attachment is empty
Constant value: **51**

ERROR_ATTACHED_FILE_IS_TOO_LARGE

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

The request attachment has no name
Constant value: **52**

ERROR_UNKNOWN_KEY

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

Unknown error
Constant value: **99**

Constructors

COMGEDEException

```
public COMGEDEException()
```

COMGEDEException

```
public COMGEDEException(String message)
```

COMGEDEException

```
public COMGEDEException(String errorCode,  
                        String message)
```

COMGEDEException

```
public COMGEDEException(String errorCode,  
                        String message,  
                        Throwable cause)
```

Methods

getErrorCode

```
public String getErrorCode()
```

(continued from last page)

getErrorMessage

```
public String getErrorMessage()
```

getErrorStackTrace

```
public String getErrorStackTrace()
```

setErrorCode

```
public void setErrorCode(String errorCode)
```

setErrorMessage

```
public void setErrorMessage(String errorMessage)
```

setErrorStackTrace

```
public void setErrorStackTrace(String errorStackTrace)
```

be.gfdi.com.common.services.ged

Class COMGEDWarning

java.lang.Object

└─be.gfdi.com.common.services.ged.COMGEDWarning

public abstract class **COMGEDWarning**
extends Object

Field Summary

public static final	CATALOG_KEY_WARNING_SEPARATOR Value:
public static final	WARN_NO_PROCESS_DEFINED_FOR_THIS_DOCTYPE This replace the error code COMGEDException.ERROR_NO_PROCESS_DEFINED_FOR_THIS_DOCTYPE This warning is returned instead. Value: 02
public static final	WARN_THIRD_IS_NOT_MUTUALITY_PERSON Value: 01

Constructor Summary

public	COMGEDWarning ()
--------	----------------------------------

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Fields

CATALOG_KEY_WARNING_SEPARATOR

public static final java.lang.String **CATALOG_KEY_WARNING_SEPARATOR**

Constant value: |

WARN_THIRD_IS_NOT_MUTUALITY_PERSON

public static final java.lang.String **WARN_THIRD_IS_NOT_MUTUALITY_PERSON**

Constant value: **01**

(continued from last page)

WARN_NO_PROCESS_DEFINED_FOR_THIS_DOCTYPE

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

This replace the error code [COMGEException.ERROR_NO_PROCESS_DEFINED_FOR_THIS_DOCTYPE](#) This warning is returned instead.

Constant value: **02**

Constructors

COMGEDWarning

```
public COMGEDWarning()
```

be.gfdi.com.common.services.ged Class CompleteTaskMessageConverter

java.lang.Object

└-be.gfdi.com.common.services.ged.CompleteTaskMessageConverter

public class **CompleteTaskMessageConverter**
extends Object

Constructor Summary

public	CompleteTaskMessageConverter() Default constructor.
--------	--

Method Summary

EventTask	convertRequest (CompleteTaskRequest req)
CompleteTaskResponse	convertResponse (Task task)

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

CompleteTaskMessageConverter

public **CompleteTaskMessageConverter**()

Default constructor.

Methods

convertRequest

public EventTask **convertRequest**([CompleteTaskRequest](#) req)

Parameters:

req

Returns:

eventTask

convertResponse

public [CompleteTaskResponse](#) **convertResponse**(Task task)

(continued from last page)

Parameters:

task

Returns:

CompleteTaskResponse

be.gfdi.com.common.services.ged Class CompleteTaskRequest

java.lang.Object

└-be.gfdi.com.common.services.ged.CompleteTaskRequest

public class **CompleteTaskRequest**
extends Object

Constructor Summary

public	CompleteTaskRequest()
--------	---------------------------------------

Method Summary

Long	getTaskId()
CompleteTaskVariable[]	getTaskVariables()
void	setTaskId(Long taskId)
void	setTaskVariables(CompleteTaskVariable[] taskVariables)

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

CompleteTaskRequest

public **CompleteTaskRequest**()

Methods

getTaskId

public Long **getTaskId**()

setTaskId

public void **setTaskId**(Long taskId)

getTaskVariables

```
public CompleteTaskVariable\[\] getTaskVariables()
```

setTaskVariables

```
public void setTaskVariables(CompleteTaskVariable\[\] taskVariables)
```

be.gfdi.com.common.services.ged

Class CompleteTaskResponse

java.lang.Object

└─be.gfdi.com.common.services.ged.CompleteTaskResponse

public class **CompleteTaskResponse**
extends Object

Constructor Summary

public	CompleteTaskResponse()
--------	--

Method Summary

Long	getProcessInstanceId()
Long	getTaskId()
String	getTaskName()
String	getTaskStatus()
Long	getTaskTokenId()
void	setProcessInstanceId(Long processInstanceId)
void	setTaskId(Long taskId)
void	setTaskName(String taskName)
void	setTaskStatus(String taskStatus)
void	setTaskTokenId(Long taskTokenId)

Methods inherited from class java.lang.Object
<code>equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait</code>

Constructors

CompleteTaskResponse

public **CompleteTaskResponse()**

Methods

getProcessInstanceId

```
public Long getProcessInstanceId()
```

setProcessInstanceId

```
public void setProcessInstanceId(Long processInstanceId)
```

getTaskStatus

```
public String getTaskStatus()
```

setTaskStatus

```
public void setTaskStatus(String taskStatus)
```

getTaskTokenId

```
public Long getTaskTokenId()
```

setTaskTokenId

```
public void setTaskTokenId(Long taskTokenId)
```

getTaskName

```
public String getTaskName()
```

setTaskName

```
public void setTaskName(String taskName)
```

getTaskId

```
public Long getTaskId()
```

(continued from last page)

setTaskId

```
public void setTaskId(Long taskId)
```


be.gfdi.com.common.services.ged Class CompleteTaskVariable

java.lang.Object

└-be.gfdi.com.common.services.ged.CompleteTaskVariable

public class **CompleteTaskVariable**
extends Object

Constructor Summary

public	CompleteTaskVariable()
--------	--

Method Summary

String	getName()
Date	getValueDate()
Double	getValueDouble()
Long	getValueLong()
String	getValueString()
void	setName(String name)
void	setValueDate(Date valueDate)
void	setValueDouble(Double valueDouble)
void	setValueLong(Long valueLong)
void	setValueString(String valueString)

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

CompleteTaskVariable

public **CompleteTaskVariable()**

Methods

getName

```
public String getName()
```

setName

```
public void setName(String name)
```

getValueDate

```
public Date getValueDate()
```

setValueDate

```
public void setValueDate(Date valueDate)
```

getValueDouble

```
public Double getValueDouble()
```

setValueDouble

```
public void setValueDouble(Double valueDouble)
```

getValueLong

```
public Long getValueLong()
```

setValueLong

```
public void setValueLong(Long valueLong)
```

getValueString

```
public String getValueString()
```

(continued from last page)

setValueString

```
public void setValueString(String valueString)
```

be.gfdi.com.common.services.ged Class CreateEventReply

java.lang.Object

└-be.gfdi.com.common.services.ged.CreateEventReply

public class **CreateEventReply**
extends Object

Constructor Summary

public	CreateEventReply()
--------	------------------------------------

Method Summary

String	getCmIndex()
String	getKey()
String	getWarning()
void	setCmIndex(String cmIndex)
void	setKey(String key)
void	setWarning(String warning)

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

CreateEventReply

public **CreateEventReply()**

Methods

getKey

public String **getKey()**

setKey

```
public void setKey(String key)
```

getWarning

```
public String getWarning()
```

setWarning

```
public void setWarning(String warning)
```

getCmIndex

```
public String getCmIndex()
```

setCmIndex

```
public void setCmIndex(String cmIndex)
```

be.gfdi.com.common.services.ged Class CreateEventRequest

java.lang.Object

└─be.gfdi.com.common.services.ged.CreateEventRequest

All Implemented Interfaces:

Serializable

public class **CreateEventRequest**
extends Object
implements Serializable

Constructor Summary

public	CreateEventRequest()
--------	--------------------------------------

Method Summary

String	getCatalogKey() return the eventKey in catalog this parameter is optional
String	getDocLanguage() get the language of the document.
String	getDocMedia() Get the document media This parameter is optional
String	getDocSource() Get the document source This parameter is optional
String	getDocType() get the event document type (for instance K00011) This parameter is mandatory
String	getFolderRef() get the folder reference to which this event is related (for instance 561123456789) This parameter is optional (default to NoRefDossier)
String	getFolderType() get the business to which this event is related (for instance K0) This parameter is optional (default to the folder type associated to the docType)
String	getFreeField()
String	getImage() get the IFS filepath of an image to import in "Content Manager" (for instance /QDTM/IMPORT/561/MyImage.tiff) This parameter is optional
byte[]	getImageBytes() Get the imagBytes This parameter is optional and is exclusive with image and imageKey
String	getImageCatalogKey()

String	<u>getImageKey()</u> get the Content Manager imageKey that has to be linked to this event This requires that the image already exists in Content Manager This parameter is optional and is exclusive with image and imageBytes
String	<u>getImageMimeType()</u> get the image mime type (for instance image/tiff) This parameter is optional (mandatory if an image is specified)
String	<u>getImageName()</u> Get the imageName This parameter is optional (mandatory if imageBytes is set)
String	<u>getNoteLog()</u> get the comment to add to the "Content Manager notelog" This parameter is optional and can only be specified if the parameter "image" is specified.
Date	<u>getReceiveDate()</u> get the catalog (or receive) date (for instance today) This parameter is optional (default to the current system date)
String	<u>getScanBatchNumber()</u> get the sequence number part of the external identifier for the event This parameter is optional
String	<u>getScanBatchNumberYear()</u> get the year part of the external identifier for the event This parameter is optional
Date	<u>getScanDate()</u>
String	<u>getScannerId()</u> get the id of the scanner used to scan the image (used by "Indicateurs de gestion") This parameter is optional
String	<u>getSecThirdCodeType()</u> TBD (list): get the secondary third code type to which the event is associated (one of <u>ThirdCodeType</u>) This parameter is optional
String	<u>getSecThirdCodeValue()</u> TBD (list): get the secondary third code value to which the event is associated This parameter is optional
String	<u>getSecThirdOrg()</u> TBD (list): get the secondary third organization to which the event is associated (i.e 561) This parameter is optional
String	<u>getSecThirdParType()</u> TBD (list): get the secondary third ParType code to which the event is associated This parameter is optional
String	<u>getThirdCodeType()</u> get the third code type to which the event is associated (one of <u>ThirdCodeType</u>) This parameter is mandatory
String	<u>getThirdCodeValue()</u> get the third value type to which the event is associated This parameter is mandatory
String	<u>getThirdManagementCenter()</u> getThirdManagementCenter is the entity that will treat this document this parameter is optional

String	<u>getThirdOrg()</u> get the third organization to which the event is associated (i.e 561) This parameter is mandatory
String	<u>getThirdParType()</u>
String	<u>getWorkflowStatus()</u> get the workflow status (one of <u>WorkflowStatus</u>) This parameter is optional
boolean	<u>isLegalDocument()</u> Information whether imported document is legal or not.
boolean	<u>isWorkflowFlg()</u> return if the workflow must be launched this parameter is optional
void	<u>setCatalogKey(String catalogKey)</u> set the eventKey for the document this parameter is optional
void	<u>setDocLanguage(String docLanguage)</u> set the language of the document This parameter is optional (default to third mother tongue)
void	<u>setDocMedia(String docMedia)</u> Set the document media (one of <u>DocMedia</u>) This parameter is optional
void	<u>setDocSource(String docSource)</u> Set the document source (one of <u>DocSource</u>) This parameter is optional - Default value is <u>DocSource.DOC_SOURCE_WEB_SERVICE</u>
void	<u>setDocType(String docType)</u> set the event document type (for instance K00011) This parameter is mandatory
void	<u>setFolderRef(String folderRef)</u> set the folder reference to which this event is related (for instance 561123456789) This parameter is optional (default to NoRefDossier)
void	<u>setFolderType(String folderType)</u> set the business to which this event is related (for instance K0) This parameter is optional (default to the folder type associated to the docType)
void	<u>setFreeField(String freeField)</u>
void	<u>setImage(String image)</u> set the IFS filepath of an image to import in "Content Manager" (for instance /QDTM/IMPORT/561/MyImage.tiff) This parameter is optional and is exclusive with the imageKey and imageBytes
void	<u>setImageBytes(byte[] imageBytes)</u> Set the imageBytes This parameter is optional and is exclusive with image and imageKey
void	<u>setImageCatalogKey(String imageCatalogKey)</u>
void	<u>setImageKey(String imageKey)</u>
void	<u>setImageMimeType(String imageMimeType)</u> set the image mime type (for instance image/tiff) This parameter is optional (mandatory if an image is specified)
void	<u>setImageName(String imageName)</u> Set the imageName This parameter is optional (mandatory if imageBytes is set)

void	<u>setLegalDocument</u> (boolean legalDocument) Information whether imported document is legal or not.
void	<u>setNoteLog</u> (String noteLog) set the comment to add to the "Content Manager notelog" This parameter is optional and can only be specified if the parameter "image" is specified It is limited to 100 positions
void	<u>setReceiveDate</u> (Date receiveDate) set the catalog (or receive) date (for instance today) This parameter is optional (default to the current system date)
void	<u>setScanBatchNumber</u> (String scanBatchNumber) set the sequence number part of the external identifier for the event This parameter is optional
void	<u>setScanBatchNumberYear</u> (String scanBatchNumberYear) set the year part of the external identifier for the event This parameter is optional
void	<u>setScanDate</u> (Date scanDate)
void	<u>setScannerId</u> (String scannerId) set the id of the scanner used to scan the image (used by "Indicateurs de gestion") This parameter is optional
void	<u>setSecThirdCodeType</u> (String secThirdCodeType) TBD (list): set the secondary third code type to which the event is associated (one of <u>ThirdCodeType</u>) This parameter is optional
void	<u>setSecThirdCodeValue</u> (String secThirdCodeValue) TBD (list): set the secondary third code value to which the event is associated This parameter is optional
void	<u>setSecThirdOrg</u> (String secThirdOrg) TBD (list): set the secondary third organization to which the event is associated (i.e 561) This parameter is optional
void	<u>setSecThirdParType</u> (String secThirdParType) TBD (list): set the secondary third ParType code to which the event is associated (i.e 561) This parameter is optional
void	<u>setThirdCodeType</u> (String thirdCodeType) set the third code type to which the event is associated (one of <u>ThirdCodeType</u>) This parameter is mandatory
void	<u>setThirdCodeValue</u> (String thirdCodeValue) set the third value type to which the event is associated This parameter is mandatory
void	<u>setThirdManagementCenter</u> (String thirdManagementCenter) set the thirdManagementCenter entity that will treat this document this parameter is optional
void	<u>setThirdOrg</u> (String thirdOrg) set the third organization to which the event is associated (i.e 561) This parameter is mandatory
void	<u>setThirdParType</u> (String thirdParType)
void	<u>setWorkflowFlg</u> (boolean workflowFlg) set the workflow flag for this document this parameter is optional

void	setWorkflowStatus (String workflowStatus) Set the workflow status (one of WorkflowStatus) This parameter is optional
String	toString ()

Methods inherited from class `java.lang.Object`

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Constructors

CreateEventRequest

```
public CreateEventRequest()
```

Methods

toString

```
public String toString()
```

getDocLanguage

```
public String getDocLanguage()
```

get the language of the document. This parameter is optional (default to third mother tongue)

setDocLanguage

```
public void setDocLanguage(String docLanguage)
```

set the language of the document This parameter is optional (default to third mother tongue)

getDocType

```
public String getDocType()
```

get the event document type (for instance K00011) This parameter is mandatory

setDocType

```
public void setDocType(String docType)
```

set the event document type (for instance K00011) This parameter is mandatory

getFolderRef

```
public String getFolderRef()
```

get the folder reference to which this event is related (for instance 561123456789) This parameter is optional (default to NoRefDossier)

setFolderRef

```
public void setFolderRef(String folderRef)
```

set the folder reference to which this event is related (for instance 561123456789) This parameter is optional (default to NoRefDossier)

getFolderType

```
public String getFolderType()
```

get the business to which this event is related (for instance K0) This parameter is optional (default to the folder type associated to the docType)

setFolderType

```
public void setFolderType(String folderType)
```

set the business to which this event is related (for instance K0) This parameter is optional (default to the folder type associated to the docType)

getImage

```
public String getImage()
```

get the IFS filepath of an image to import in "Content Manager" (for instance /QDTM/IMPORT/561/MyImage.tiff) This parameter is optional

setImage

```
public void setImage(String image)
```

set the IFS filepath of an image to import in "Content Manager" (for instance /QDTM/IMPORT/561/MyImage.tiff) This parameter is optional and is exclusive with the imageKey and imageBytes

getImageMimeType

```
public String getImageMimeType()
```

get the image mime type (for instance image/tiff) This parameter is optional (mandatory if an image is specified)

setImageMimeType

```
public void setImageMimeType(String imageMimeType)
```

set the image mime type (for instance image/tiff) This parameter is optional (mandatory if an image is specified)

getImageKey

```
public String getImageKey()
```

get the Content Manager imageKey that has to be linked to this event This requires that the image already exists in Content Manager This parameter is optional and is exclusive with image and imageBytes

Returns:

imageKey

(continued from last page)

setImageKey

```
public void setImageKey(String imageKey)
```

getNoteLog

```
public String getNoteLog()
```

get the comment to add to the "Content Manager notelog" This parameter is optional and can only be specified if the parameter "image" is specified. It is limited to 100 positions

setNoteLog

```
public void setNoteLog(String noteLog)
```

set the comment to add to the "Content Manager notelog" This parameter is optional and can only be specified if the parameter "image" is specified It is limited to 100 positions

getReceiveDate

```
public Date getReceiveDate()
```

get the catalog (or receive) date (for instance today) This parameter is optional (default to the current system date)

setReceiveDate

```
public void setReceiveDate(Date receiveDate)
```

set the catalog (or receive) date (for instance today) This parameter is optional (default to the current system date)

getScanBatchNumber

```
public String getScanBatchNumber()
```

get the sequence number part of the external identifier for the event This parameter is optional

setScanBatchNumber

```
public void setScanBatchNumber(String scanBatchNumber)
```

set the sequence number part of the external identifier for the event This parameter is optional

getScanBatchNumberYear

```
public String getScanBatchNumberYear()
```

get the year part of the external identifier for the event This parameter is optional

setScanBatchNumberYear

```
public void setScanBatchNumberYear(String scanBatchNumberYear)
```

set the year part of the external identifier for the event This parameter is optional

(continued from last page)

getScannerId

```
public String getScannerId()
```

get the id of the scanner used to scan the image (used by "Indicateurs de gestion") This parameter is optional

setScannerId

```
public void setScannerId(String scannerId)
```

set the id of the scanner used to scan the image (used by "Indicateurs de gestion") This parameter is optional

getSecThirdCodeType

```
public String getSecThirdCodeType()
```

TBD (list): get the secondary third code type to which the event is associated (one of [ThirdCodeType](#)) This parameter is optional

setSecThirdCodeType

```
public void setSecThirdCodeType(String secThirdCodeType)
```

TBD (list): set the secondary third code type to which the event is associated (one of [ThirdCodeType](#)) This parameter is optional

getSecThirdCodeValue

```
public String getSecThirdCodeValue()
```

TBD (list): get the secondary third code value to which the event is associated This parameter is optional

setSecThirdCodeValue

```
public void setSecThirdCodeValue(String secThirdCodeValue)
```

TBD (list): set the secondary third code value to which the event is associated This parameter is optional

getSecThirdOrg

```
public String getSecThirdOrg()
```

TBD (list): get the secondary third organization to which the event is associated (i.e 561) This parameter is optional

setSecThirdOrg

```
public void setSecThirdOrg(String secThirdOrg)
```

TBD (list): set the secondary third organization to which the event is associated (i.e 561) This parameter is optional

getThirdCodeType

```
public String getThirdCodeType()
```

get the third code type to which the event is associated (one of [ThirdCodeType](#)) This parameter is mandatory

(continued from last page)

setThirdCodeType

```
public void setThirdCodeType(String thirdCodeType)
```

set the third code type to which the event is associated (one of [ThirdCodeType](#)) This parameter is mandatory

getThirdCodeValue

```
public String getThirdCodeValue()
```

get the third value type to which the event is associated This parameter is mandatory

setThirdCodeValue

```
public void setThirdCodeValue(String thirdCodeValue)
```

set the third value type to which the event is associated This parameter is mandatory

getThirdOrg

```
public String getThirdOrg()
```

get the third organization to which the event is associated (i.e 561) This parameter is mandatory

setThirdOrg

```
public void setThirdOrg(String thirdOrg)
```

set the third organization to which the event is associated (i.e 561) This parameter is mandatory

getSecThirdParType

```
public final String getSecThirdParType()
```

TBD (list): get the secondary third ParType code to which the event is associated This parameter is optional

setSecThirdParType

```
public final void setSecThirdParType(String secThirdParType)
```

TBD (list): set the secondary third ParType code to which the event is associated (i.e 561) This parameter is optional

getThirdManagementCenter

```
public String getThirdManagementCenter()
```

getThirdManagementCenter is the entity that will treat this document this parameter is optional

setThirdManagementCenter

```
public void setThirdManagementCenter(String thirdManagementCenter)
```

set the thirdManagementCenter entity that will treat this document this parameter is optional

isWorkflowFlg

```
public boolean isWorkflowFlg()
```

(continued from last page)

return if the workflow must be launched this parameter is optional

setWorkflowFlg

```
public void setWorkflowFlg(boolean workflowFlg)
```

set the workflow flag for this document this parameter is optional

getCatalogKey

```
public String getCatalogKey()
```

return the eventKey in catalog this parameter is optional

setCatalogKey

```
public void setCatalogKey(String catalogKey)
```

set the eventKey for the document this parameter is optional

getFreeField

```
public String getFreeField()
```

setFreeField

```
public void setFreeField(String freeField)
```

getThirdParType

```
public String getThirdParType()
```

setThirdParType

```
public void setThirdParType(String thirdParType)
```

getImageCatalogKey

```
public String getImageCatalogKey()
```

setImageCatalogKey

```
public void setImageCatalogKey(String imageCatalogKey)
```

(continued from last page)

getScanDate

```
public Date getScanDate()
```

setScanDate

```
public void setScanDate(Date scanDate)
```

getWorkflowStatus

```
public String getWorkflowStatus()
```

get the workflow status (one of [WorkflowStatus](#)) This parameter is optional

setWorkflowStatus

```
public void setWorkflowStatus(String workflowStatus)
```

Set the workflow status (one of [WorkflowStatus](#)) This parameter is optional

isLegalDocument

```
public boolean isLegalDocument()
```

Information whether imported document is legal or not. Document validation process will be more restrictive for legal documents (PDF only)

Returns:

the legalDocument

setLegalDocument

```
public void setLegalDocument(boolean legalDocument)
```

Information whether imported document is legal or not. Document validation process will be more restrictive for legal documents (PDF only)

Parameters:

legalDocument - the legalDocument to set

getDocMedia

```
public String getDocMedia()
```

Get the document media This parameter is optional

setDocMedia

```
public void setDocMedia(String docMedia)
```

Set the document media (one of [DocMedia](#)) This parameter is optional

(continued from last page)

getDocSource

```
public String getDocSource()
```

Get the document source This parameter is optional

setDocSource

```
public void setDocSource(String docSource)
```

Set the document source (one of [DocSource](#)) This parameter is optional - Default value is [DocSource.DOC_SOURCE_WEB_SERVICE](#)

getImageBytes

```
public byte[] getImageBytes()
```

Get the imageBytes This parameter is optional and is exclusive with image and imageKey

Returns:

the imageBytes

setImageBytes

```
public void setImageBytes(byte[] imageBytes)
```

Set the imageBytes This parameter is optional and is exclusive with image and imageKey

Parameters:

imageBytes - the imageBytes to set

setImageName

```
public void setImageName(String imageName)
```

Set the imageName This parameter is optional (mandatory if imageBytes is set)

Parameters:

imageName - the imageName to set

getImageName

```
public String getImageName()
```

Get the imageName This parameter is optional (mandatory if imageBytes is set)

Returns:

the imageName

be.gfdi.com.common.services.ged

Class DispatchEventMessageConverter

java.lang.Object

└─be.gfdi.com.common.services.ged.DispatchEventMessageConverter

public class **DispatchEventMessageConverter**
extends Object

Constructor Summary

public	DispatchEventMessageConverter() Default constructor.
--------	---

Method Summary

Event	convertRequest (DispatchEventRequest req)
DispatchEventResponse	convertResponse (Task task)

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

DispatchEventMessageConverter

public **DispatchEventMessageConverter**()

Default constructor.

Methods

convertRequest

public Event **convertRequest**([DispatchEventRequest](#) req)

Parameters:

req

Returns:

Event

convertResponse

public [DispatchEventResponse](#) **convertResponse**(Task task)

(continued from last page)

Parameters:

task

Returns:

DispatchEventResponse

be.gfdi.com.common.services.ged

Class DispatchEventRequest

java.lang.Object

└─be.gfdi.com.common.services.ged.DispatchEventRequest

public class **DispatchEventRequest**
extends Object

Constructor Summary

public	DispatchEventRequest()
--------	--

Method Summary

Date	getCatalogDate()
String	getCatalogImageId()
String	getCatalogKey()
String	getDocTypeCode()
String	getDocTypeLabelFr()
String	getDocTypeLabelNl()
String	getDocTypeSubClass()
String	getDocumentLanguage()
String	getEventExternalId()
String	getOrganization()
String	getProcessDefinitionName()
String	getThirdAgencyId()
String	getThirdAgencyNameDe()
String	getThirdAgencyNameFr()
String	getThirdAgencyNameNl()

String	<u>getThirdAgencyUniqueName()</u>
String	<u>getThirdExternalNbr()</u>
String	<u>getThirdFirstName()</u>
String	<u>getThirdLastName()</u>
String	<u>getThirdManagementCenterId()</u>
String	<u>getThirdManagementCenterLabelDe()</u>
String	<u>getThirdManagementCenterLabelFr()</u>
String	<u>getThirdManagementCenterLabelNl()</u>
String	<u>getThirdManagementCenterUniqueName()</u>
String	<u>getThirdNissNumber()</u>
String	<u>getThirdOrganization()</u>
String	<u>getThirdUniqueNumber()</u>
String	<u>getUser()</u>
<u>DispatchEventVariable</u> <u>[]</u>	<u>getVariables()</u>
boolean	<u>isThirdStaffMember()</u>
void	<u>setCatalogDate</u> (Date catalogDate)
void	<u>setCatalogImageId</u> (String catalogImageId)
void	<u>setCatalogKey</u> (String catalogKey)
void	<u>setDocTypeCode</u> (String docTypeCode)
void	<u>setDocTypeLabelFr</u> (String docTypeLabelFr)
void	<u>setDocTypeLabelNl</u> (String docTypeLabelNl)
void	<u>setDocTypeSubClass</u> (String docTypeSubClass)
void	<u>setDocumentLanguage</u> (String documentLanguage)
void	<u>setEventExternalId</u> (String eventExternalId)

void	<u>setOrganization</u> (String organization)
void	<u>setProcessDefinitionName</u> (String processDefinitionName)
void	<u>setThirdAgencyId</u> (String thirdAgencyId)
void	<u>setThirdAgencyNameDe</u> (String thirdAgencyNameDe)
void	<u>setThirdAgencyNameFr</u> (String thirdAgencyNameFr)
void	<u>setThirdAgencyNameNl</u> (String thirdAgencyNameNl)
void	<u>setThirdAgencyUniqueName</u> (String thirdAgencyUniqueName)
void	<u>setThirdExternalNbr</u> (String thirdExternalNbr)
void	<u>setThirdFirstName</u> (String thirdFirstName)
void	<u>setThirdLastName</u> (String thirdLastName)
void	<u>setThirdManagementCenterId</u> (String thirdManagementCenterId)
void	<u>setThirdManagementCenterLabelDe</u> (String thirdManagementCenterLabelDe)
void	<u>setThirdManagementCenterLabelFr</u> (String thirdManagementCenterLabelFr)
void	<u>setThirdManagementCenterLabelNl</u> (String thirdManagementCenterLabelNl)
void	<u>setThirdManagementCenterUniqueName</u> (String thirdManagementCenterUniqueName)
void	<u>setThirdNissNumber</u> (String thirdNissNumber)
void	<u>setThirdOrganization</u> (String thirdOrganization)
void	<u>setThirdStaffMember</u> (boolean thirdStaffMember)
void	<u>setThirdUniqueNumber</u> (String thirdUniqueNumber)
void	<u>setUser</u> (String user)
void	<u>setVariables</u> (<u>DispatchEventVariable[]</u> variables)

Methods inherited from class `java.lang.Object`

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

(continued from last page)

Constructors

DispatchEventRequest

```
public DispatchEventRequest()
```

Methods

getThirdUniqueNumber

```
public String getThirdUniqueNumber()
```

setThirdUniqueNumber

```
public void setThirdUniqueNumber(String thirdUniqueNumber)
```

getThirdOrganization

```
public String getThirdOrganization()
```

setThirdOrganization

```
public void setThirdOrganization(String thirdOrganization)
```

getThirdNissNumber

```
public String getThirdNissNumber()
```

setThirdNissNumber

```
public void setThirdNissNumber(String thirdNissNumber)
```

getThirdExternalNbr

```
public String getThirdExternalNbr()
```

setThirdExternalNbr

```
public void setThirdExternalNbr(String thirdExternalNbr)
```

isThirdStaffMember

```
public boolean isThirdStaffMember()
```

setThirdStaffMember

```
public void setThirdStaffMember(boolean thirdStaffMember)
```

getThirdLastName

```
public String getThirdLastName()
```

setThirdLastName

```
public void setThirdLastName(String thirdLastName)
```

getThirdFirstName

```
public String getThirdFirstName()
```

setThirdFirstName

```
public void setThirdFirstName(String thirdFirstName)
```

getThirdAgencyId

```
public String getThirdAgencyId()
```

setThirdAgencyId

```
public void setThirdAgencyId(String thirdAgencyId)
```

getThirdAgencyUniqueName

```
public String getThirdAgencyUniqueName()
```

setThirdAgencyUniqueName

```
public void setThirdAgencyUniqueName(String thirdAgencyUniqueName)
```

(continued from last page)

getThirdAgencyNameFr

```
public String getThirdAgencyNameFr()
```

setThirdAgencyNameFr

```
public void setThirdAgencyNameFr(String thirdAgencyNameFr)
```

getThirdAgencyNameNl

```
public String getThirdAgencyNameNl()
```

setThirdAgencyNameNl

```
public void setThirdAgencyNameNl(String thirdAgencyNameNl)
```

getThirdAgencyNameDe

```
public String getThirdAgencyNameDe()
```

setThirdAgencyNameDe

```
public void setThirdAgencyNameDe(String thirdAgencyNameDe)
```

getThirdManagementCenterId

```
public String getThirdManagementCenterId()
```

setThirdManagementCenterId

```
public void setThirdManagementCenterId(String thirdManagementCenterId)
```

getThirdManagementCenterUniqueName

```
public String getThirdManagementCenterUniqueName()
```

(continued from last page)

setThirdManagementCenterUniqueName

```
public void setThirdManagementCenterUniqueName(String thirdManagementCenterUniqueName)
```

getThirdManagementCenterLabelFr

```
public String getThirdManagementCenterLabelFr()
```

setThirdManagementCenterLabelFr

```
public void setThirdManagementCenterLabelFr(String thirdManagementCenterLabelFr)
```

getThirdManagementCenterLabelNl

```
public String getThirdManagementCenterLabelNl()
```

setThirdManagementCenterLabelNl

```
public void setThirdManagementCenterLabelNl(String thirdManagementCenterLabelNl)
```

getThirdManagementCenterLabelDe

```
public String getThirdManagementCenterLabelDe()
```

setThirdManagementCenterLabelDe

```
public void setThirdManagementCenterLabelDe(String thirdManagementCenterLabelDe)
```

getOrganization

```
public String getOrganization()
```

setOrganization

```
public void setOrganization(String organization)
```

getUser

```
public String getUser()
```

(continued from last page)

setUser

```
public void setUser(String user)
```

getDocumentLanguage

```
public String getDocumentLanguage()
```

setDocumentLanguage

```
public void setDocumentLanguage(String documentLanguage)
```

getCatalogKey

```
public String getCatalogKey()
```

setCatalogKey

```
public void setCatalogKey(String catalogKey)
```

getCatalogDate

```
public Date getCatalogDate()
```

setCatalogDate

```
public void setCatalogDate(Date catalogDate)
```

getCatalogImageId

```
public String getCatalogImageId()
```

setCatalogImageId

```
public void setCatalogImageId(String catalogImageId)
```

(continued from last page)

getDocTypeCode

```
public String getDocTypeCode()
```

setDocTypeCode

```
public void setDocTypeCode(String docTypeCode)
```

getDocTypeLabelFr

```
public String getDocTypeLabelFr()
```

setDocTypeLabelFr

```
public void setDocTypeLabelFr(String docTypeLabelFr)
```

getDocTypeLabelNl

```
public String getDocTypeLabelNl()
```

setDocTypeLabelNl

```
public void setDocTypeLabelNl(String docTypeLabelNl)
```

getDocTypeSubClass

```
public String getDocTypeSubClass()
```

setDocTypeSubClass

```
public void setDocTypeSubClass(String docTypeSubClass)
```

getEventExternalId

```
public String getEventExternalId()
```

setEventExternalId

```
public void setEventExternalId(String eventExternalId)
```

(continued from last page)

getProcessDefinitionName

```
public String getProcessDefinitionName()
```

setProcessDefinitionName

```
public void setProcessDefinitionName(String processDefinitionName)
```

getVariables

```
public DispatchEventVariable\[\] getVariables()
```

setVariables

```
public void setVariables(DispatchEventVariable\[\] variables)
```

be.gfdi.com.common.services.ged Class DispatchEventResponse

java.lang.Object

└─be.gfdi.com.common.services.ged.DispatchEventResponse

public class **DispatchEventResponse**
extends Object

Constructor Summary

public	DispatchEventResponse()
--------	---

Method Summary

Long	getProcessInstanceId()
Long	getTaskId()
String	getTaskName()
String	getTaskStatus()
Long	getTaskTokenId()
void	setProcessInstanceId(Long processInstanceId)
void	setTaskId(Long taskId)
void	setTaskName(String taskName)
void	setTaskStatus(String taskStatus)
void	setTaskTokenId(Long taskTokenId)

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

DispatchEventResponse

public **DispatchEventResponse()**

Methods

getProcessInstanceId

```
public Long getProcessInstanceId()
```

setProcessInstanceId

```
public void setProcessInstanceId(Long processInstanceId)
```

getTaskStatus

```
public String getTaskStatus()
```

setTaskStatus

```
public void setTaskStatus(String taskStatus)
```

getTaskTokenId

```
public Long getTaskTokenId()
```

setTaskTokenId

```
public void setTaskTokenId(Long taskTokenId)
```

getTaskName

```
public String getTaskName()
```

setTaskName

```
public void setTaskName(String taskName)
```

getTaskId

```
public Long getTaskId()
```

(continued from last page)

setTaskId

```
public void setTaskId(Long taskId)
```


be.gfdi.com.common.services.ged Class DispatchEventVariable

java.lang.Object

└─be.gfdi.com.common.services.ged.DispatchEventVariable

public class **DispatchEventVariable**
extends Object

Constructor Summary

public	DispatchEventVariable()
--------	---

Method Summary

Date	getDateValue()
java.math.BigDecimal	getDecimalValue()
Long	getLongValue()
String	getName()
String	getStringValue()
void	setDateValue(Date dateValue)
void	setDecimalValue(java.math.BigDecimal decimalValue)
void	setLongValue(Long longValue)
void	setName(String name)
void	setStringValue(String stringValue)

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

DispatchEventVariable

public **DispatchEventVariable()**

Methods

getName

```
public String getName()
```

setName

```
public void setName(String name)
```

getStringValue

```
public String getStringValue()
```

setStringValue

```
public void setStringValue(String stringValue)
```

getDateValue

```
public Date getDateValue()
```

setDateValue

```
public void setDateValue(Date dateValue)
```

getLongValue

```
public Long getLongValue()
```

setLongValue

```
public void setLongValue(Long longValue)
```

getDecimalValue

```
public java.math.BigDecimal getDecimalValue()
```

(continued from last page)

setDecimalValue

```
public void setDecimalValue(java.math.BigDecimal decimalValue)
```

be.gfdi.com.common.services.ged

Class DocMedia

java.lang.Object

└─be.gfdi.com.common.services.ged.DocMedia

public abstract class **DocMedia**
extends Object

Field Summary

public static final	DOC_MEDIA_ELECTRONIC_FLOW Electronic flow media Value: 03
public static final	DOC_MEDIA_EMAIL Email media Value: 04
public static final	DOC_MEDIA_EXCEL MS-Word media Value: 06
public static final	DOC_MEDIA_FAX Fax media Value: 02
public static final	DOC_MEDIA_INFOBOX Infobox media Value: 05
public static final	DOC_MEDIA_PAPER Paper media Value: 01

Constructor Summary

public	DocMedia()
--------	----------------------------

Method Summary

static List	getAllDocMedia() Get the list of all medias
-------------	--

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Fields

(continued from last page)

DOC_MEDIA_PAPER

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

Paper media
Constant value: 01

DOC_MEDIA_FAX

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

Fax media
Constant value: 02

DOC_MEDIA_ELECTRONIC_FLOW

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

Electronic flow media
Constant value: 03

DOC_MEDIA_EMAIL

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

Email media
Constant value: 04

DOC_MEDIA_INFOBOX

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

Infobox media
Constant value: 05

DOC_MEDIA_EXCEL

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

MS-Word media
Constant value: 06

Constructors

DocMedia

```
public DocMedia()
```

Methods

getAllDocMedia

```
public static List getAllDocMedia()
```

Get the list of all medias

be.gfdi.com.common.services.ged

Class DocSource

java.lang.Object

└--be.gfdi.com.common.services.ged.DocSource

public abstract class **DocSource**
extends Object

Field Summary

public static final	DOC_SOURCE_BULK_IMPORT Bulk Import source Value: 03
public static final	DOC_SOURCE_DGC DGC source Value: 02
public static final	DOC_SOURCE_ELECTRONIC_FLOW Electronic Flow source Value: 04
public static final	DOC_SOURCE_IM Integrated Mailing source Value: 05
public static final	DOC_SOURCE_IM_LIGHT Integrated Mailing Light source Value: 06
public static final	DOC_SOURCE_JADE Jade source Value: 01
public static final	DOC_SOURCE_OFD OFD source Value: 07
public static final	DOC_SOURCE_WEB_SERVICE Web Service source Value: 08

Constructor Summary

public	DocSource ()
--------	-------------------------------

Method Summary

static List	getAllDocSource () Get the list of all sources
-------------	--

Methods inherited from class `java.lang.Object``equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Fields

DOC_SOURCE_JADE

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

Jade source
Constant value: **01**

DOC_SOURCE_DGC

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

DGC source
Constant value: **02**

DOC_SOURCE_BULK_IMPORT

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

Bulk Import source
Constant value: **03**

DOC_SOURCE_ELECTRONIC_FLOW

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

Electronic Flow source
Constant value: **04**

DOC_SOURCE_IM

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

Integrated Mailing source
Constant value: **05**

DOC_SOURCE_IM_LIGHT

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

Integrated Mailing Light source
Constant value: **06**

DOC_SOURCE_OFD

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

OFD source
Constant value: **07**

(continued from last page)

DOC_SOURCE_WEB_SERVICE

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

Web Service source
Constant value: 08

Constructors

DocSource

```
public DocSource()
```

Methods

getAllDocSource

```
public static List getAllDocSource()
```

Get the list of all sources

be.gfdi.com.common.services.ged Class FindByParametersRequest

java.lang.Object

└─be.gfdi.com.common.services.ged.FindByParametersRequest

All Implemented Interfaces:

Serializable

public class **FindByParametersRequest**
extends Object
implements Serializable

Parameters used by the CatalogService.findByParameters service

Constructor Summary

public	FindByParametersRequest()
--------	---

Method Summary

Date	getCatalogEndDate() get the end date of the period on wich to search (inclusive).
Date	getCatalogStartDate() get the start date of the period on wich to search (inclusive).
String	getDirection() get the direction.
String	getDocType() get the docType on which to search.
String	getFolderRef()
String	getFolderType() get the folder type (for instance K0).
int	getPageNumber() given the maximum number of result, specify wich page of result (for the first page pageNumber is 1, for the second page pageNumber is 2, etc) This parameter is optional (default is 1)
int	getPageSize() get the the maximum number of result This parameter is mandatory
String	getSubClass() get the subClass on which to search.
String	getThirdCodeType() get the code type used to identify the third (one of ThirdCodeType)

String	<u>getThirdCodeValue()</u> get the actual code that identifies the third
String	<u>getThirdOrg()</u> get the organization of the third
void	<u>setCatalogEndDate()</u> (Date catalogEndDate) set the end date of the period on wich to search (inclusive).
void	<u>setCatalogStartDate()</u> (Date catalogStartDate) set the start date of the period on wich to search (inclusive).
void	<u>setDirection()</u> (String direction) set the direction.
void	<u>setDocType()</u> (String docType) set the docType on which to search.
void	<u>setFolderRef()</u> (String folderRef)
void	<u>setFolderType()</u> (String folderType) set the folder type (for instance K0).
void	<u>setPageNumber()</u> (int pageNumber) given the maximum number of result, specify wich page of result (for the first page pageNumber is 1, for the second page pageNumber is 2, etc) This parameter is mandatory
void	<u>setPageSize()</u> (int pageSize) set the the maximum number of result This parameter is mandatory
void	<u>setSubClass()</u> (String subClass) set the subClass on which to search.
void	<u>setThirdCodeType()</u> (String thirdCodeType) set the code type used to identify the third (one of <u>ThirdCodeType</u>)
void	<u>setThirdCodeValue()</u> (String thirdCodeValue) set the actual code that identifies the third
void	<u>setThirdOrg()</u> (String thirdOrg) set the organization of the third

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

FindByParametersRequest

```
public FindByParametersRequest()
```

Methods

(continued from last page)

getCatalogEndDate

```
public Date getCatalogEndDate()
```

get the end date of the period on wich to search (inclusive). This parameter is optional

setCatalogEndDate

```
public void setCatalogEndDate(Date catalogEndDate)
```

set the end date of the period on wich to search (inclusive). This parameter is optional

getCatalogStartDate

```
public Date getCatalogStartDate()
```

get the start date of the period on wich to search (inclusive). This parameter is optional The actual field that is used is G_CAT.RCVDAT and not DCOBJECT.OB_CRDATE

setCatalogStartDate

```
public void setCatalogStartDate(Date catalogStartDate)
```

set the start date of the period on wich to search (inclusive). The actual field that is used is G_CAT.RCVDAT and not DCOBJECT.OB_CRDATE This parameter is optional

getDocType

```
public String getDocType()
```

get the docType on which to search. This parameter is optional

setDocType

```
public void setDocType(String docType)
```

set the docType on which to search. This parameter is optional

getSubClass

```
public String getSubClass()
```

get the subClass on which to search. This parameter is optional, possible values are (CSO, MME, OFD, IED)

setSubClass

```
public void setSubClass(String subClass)
```

set the subClass on which to search. This parameter is optional. Possible values are (CSO, MME, OFD, IED)

getFolderType

```
public String getFolderType()
```

get the folder type (for instance K0). This parameter is optional

(continued from last page)

setFolderType

```
public void setFolderType(String folderType)
```

set the folder type (for instance K0). This parameter is optional

getDirection

```
public String getDirection()
```

get the direction. This parameter is optional. Possible values are (IN, OUT)

setDirection

```
public void setDirection(String direction)
```

set the direction. This parameter is optional. Possible values are (IN, OUT)

getPageNumber

```
public int getPageNumber()
```

given the maximum number of result, specify wich page of result (for the first page pageNumber is 1, for the second page pageNumber is 2, etc) This parameter is optional (default is 1)

setPageNumber

```
public void setPageNumber(int pageNumber)
```

given the maximum number of result, specify wich page of result (for the first page pageNumber is 1, for the second page pageNumber is 2, etc) This parameter is mandatory

getPageSize

```
public int getPageSize()
```

get the the maximum number of result This parameter is mandatory

setPageSize

```
public void setPageSize(int pageSize)
```

set the the maximum number of result This parameter is mandatory

getThirdCodeType

```
public String getThirdCodeType()
```

get the code type used to identify the third (one of [ThirdCodeType](#))

setThirdCodeType

```
public void setThirdCodeType(String thirdCodeType)
```

set the code type used to identify the third (one of [ThirdCodeType](#))

(continued from last page)

getThirdCodeValue

```
public String getThirdCodeValue()
```

get the actual code that identifies the third

setThirdCodeValue

```
public void setThirdCodeValue(String thirdCodeValue)
```

set the actual code that identifies the third

getThirdOrg

```
public String getThirdOrg()
```

get the organization of the third

setThirdOrg

```
public void setThirdOrg(String thirdOrg)
```

set the organization of the third

getFolderRef

```
public String getFolderRef()
```

setFolderRef

```
public void setFolderRef(String folderRef)
```

be.gfdi.com.common.services.ged Class FindDocTypesRequest

java.lang.Object

└-be.gfdi.com.common.services.ged.FindDocTypesRequest

All Implemented Interfaces:

Serializable

public class **FindDocTypesRequest**
extends Object
implements Serializable

Constructor Summary

public	FindDocTypesRequest()
public	FindDocTypesRequest (String codeStartWith)

Method Summary

String	getCodeStartWith()
void	setCodeStartWith (String codeStartWith)

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

FindDocTypesRequest

public **FindDocTypesRequest**()

FindDocTypesRequest

public **FindDocTypesRequest**(String codeStartWith)

Methods

getCodeStartWith

public String **getCodeStartWith**()

(continued from last page)

setCodeStartWith

```
public void setCodeStartWith(String codeStartWith)
```

be.gfdi.com.common.services.ged Class GEDCanvasStatus

java.lang.Object

└--be.gfdi.com.common.services.ged.GEDCanvasStatus

public abstract class **GEDCanvasStatus**
extends Object

constants for each type of canvas status **PRODUCTION** = new CanvasStatus("P"); **TEST** = new CanvasStatus("T");

Field Summary

public static final	PRODUCTION Value: P
public static final	TEST Value: T

Constructor Summary

public	GEDCanvasStatus()
--------	-----------------------------------

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Fields

PRODUCTION

public static final java.lang.String **PRODUCTION**

Constant value: **P**

TEST

public static final java.lang.String **TEST**

Constant value: **T**

Constructors

GEDCanvasStatus

public **GEDCanvasStatus()**

be.gfdi.com.common.services.ged Class GEDContactInfoType

java.lang.Object

↳ be.gfdi.com.common.services.ged.GEDContactInfoType

public abstract class **GEDContactInfoType**
extends Object

RESIDENCE_CODE = "01"; MAIL_CODE = "02"; PROFESIONNAL_CODE = "03"; HEADQUARTER_CODE = "04";
OFFICE_CODE = "05"; FACTURATION_CODE = "06"; CONTACTPERSON_PARTY_CODE = "08";
CONTACTPERSON_CODE = "09";

Field Summary

public static final	CONTACTPERSON Value: 09
public static final	CONTACTPERSONPARTY Value: 08
public static final	FACTURATION Value: 06
public static final	HEADQUARTER Value: 04
public static final	MAIL Value: 02
public static final	OFFICE Value: 05
public static final	PROFESIONNAL Value: 03
public static final	RESIDENCE Value: 01

Constructor Summary

public	GEDContactInfoType()
--------	--------------------------------------

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

(continued from last page)

Fields

RESIDENCE

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

Constant value: 01

MAIL

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

Constant value: 02

PROFESIONNAL

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

Constant value: 03

HEADQUARTER

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

Constant value: 04

OFFICE

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

Constant value: 05

FACTURATION

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

Constant value: 06

CONTACTPERSONPARTY

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

Constant value: 08

CONTACTPERSON

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

Constant value: 09

(continued from last page)

Constructors

GEDContactInfoType

```
public GEDContactInfoType()
```

be.gfdi.com.common.services.ged

Class GEDDocType

java.lang.Object

└-be.gfdi.com.common.services.ged.GEDDocType

All Implemented Interfaces:

Serializable

public class **GEDDocType**
 extends Object
 implements Serializable

Constructor Summary

public	GEDDocType()
--------	------------------------------

Method Summary

String	getArchiveBoxType()
String	getArchiveMode()
String	getCode()
String	getConfidentialLevel()
String	getDcId()
String	getDgcFilter()
String	getDgcLabelFr()
String	getDgcLabelGe()
String	getDgcLabelNl()
String	getDgcSection()
String	getDirection()
String	getLabelFr()
String	getLabelGe()
String	getLabelNl()

boolean	<u>isDgcCataloguable()</u>
void	<u>setArchiveBoxType</u> (String archiveBoxType)
void	<u>setArchiveMode</u> (String archiveMode)
void	<u>setCode</u> (String code)
void	<u>setConfidentialLevel</u> (String confidentialLevel)
void	<u>setDcId</u> (String dcId)
void	<u>setDgcCataloguable</u> (boolean dgcCataloguable)
void	<u>setDgcFilter</u> (String dgcFilter)
void	<u>setDgcLabelFr</u> (String dgcLabelFr)
void	<u>setDgcLabelGe</u> (String dgcLabelGe)
void	<u>setDgcLabelNl</u> (String dgcLabelNl)
void	<u>setDgcSection</u> (String dgcSection)
void	<u>setDirection</u> (String direction)
void	<u>setLabelFr</u> (String labelFr)
void	<u>setLabelGe</u> (String labelGe)
void	<u>setLabelNl</u> (String labelNl)

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

GEDDocType

public **GEDDocType**()

Methods

(continued from last page)

getDirection

```
public String getDirection()
```

setDirection

```
public void setDirection(String direction)
```

getDgcSection

```
public String getDgcSection()
```

setDgcSection

```
public void setDgcSection(String dgcSection)
```

getDgcFilter

```
public String getDgcFilter()
```

setDgcFilter

```
public void setDgcFilter(String dgcFilter)
```

isDgcCataloguable

```
public boolean isDgcCataloguable()
```

setDgcCataloguable

```
public void setDgcCataloguable(boolean dgcCataloguable)
```

getDgcLabelFr

```
public String getDgcLabelFr()
```

setDgcLabelFr

```
public void setDgcLabelFr(String dgcLabelFr)
```

(continued from last page)

getDgcLabelNl

```
public String getDgcLabelNl()
```

setDgcLabelNl

```
public void setDgcLabelNl(String dgcLabelNl)
```

getDgcLabelGe

```
public String getDgcLabelGe()
```

setDgcLabelGe

```
public void setDgcLabelGe(String dgcLabelGe)
```

getArchiveBoxType

```
public String getArchiveBoxType()
```

setArchiveBoxType

```
public void setArchiveBoxType(String archiveBoxType)
```

getArchiveMode

```
public String getArchiveMode()
```

setArchiveMode

```
public void setArchiveMode(String archiveMode)
```

getCode

```
public String getCode()
```

(continued from last page)

setCode

```
public void setCode(String code)
```

getDcId

```
public String getDcId()
```

setDcId

```
public void setDcId(String dcId)
```

getLabelFr

```
public String getLabelFr()
```

setLabelFr

```
public void setLabelFr(String labelFr)
```

getLabelNl

```
public String getLabelNl()
```

setLabelNl

```
public void setLabelNl(String labelNl)
```

getLabelGe

```
public String getLabelGe()
```

setLabelGe

```
public void setLabelGe(String labelGe)
```

getConfidentialLevel

```
public String getConfidentialLevel()
```

(continued from last page)

setConfidentialLevel

```
public void setConfidentialLevel(String confidentialLevel)
```

be.gfdi.com.common.services.ged Class GEDEntityType

java.lang.Object

└─be.gfdi.com.common.services.ged.GEDEntityType

public abstract class **GEDEntityType**
extends Object

constants for EntityType AGENCY ==> entityType = "Agency"; COWORKER ==> entityType = "Coworker"; DEDICATED ==> entityType = "DedAge"; DEPARTMENT ==> entityType = "Department"; MANAGEMENTCENTER ==> entityType = "ManCen"; MEDICALADVISOR ==> entityType = "MedAdv"; PRIMARYAGENCY ==> entityType = "PriAge"; PRINTQUEUE ==> entityType = "PriQue"; SECONDARYAGENCY ==> entityType = "SecAge"; SERVICE ==> entityType = "Service"; SOCIALSEAT ==> entityType = "SocialSeat"; SURGERY ==> entityType = "SurMedAdv"; TEAM ==> entityType = "Team"; VIRTUALAGENCY ==> entityType = "VirAge"; WORKSTATION ==> entityType = "WorSta";

Field Summary

public static final	AGENCY Value: Agency
public static final	COWORKER Value: Coworker
public static final	DEDICATEDAGENCY Value: DedAge
public static final	DEPARTMENT Value: Department
public static final	MANAGEMENTCENTER Value: ManCen
public static final	MEDICALADVISER Value: MedAdv
public static final	PRIMARYAGENCY Value: PriAge
public static final	PRINTQUEUE Value: PriQue
public static final	SECONDARYAGENCY Value: SecAge
public static final	SERVICE Value: Service

public static final	SOCIALSEAT Value: SocialSeat
public static final	SURGERY Value: SurMedAdv
public static final	TEAM Value: TEAM
public static final	VIRTUALAGENCY Value: VirAge
public static final	WORKSTATION Value: WorSta

Constructor Summary

public	GEDEntityType()
--------	---------------------------------

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Fields

AGENCY

public static final java.lang.String **AGENCY**

Constant value: **Agency**

COWORKER

public static final java.lang.String **COWORKER**

Constant value: **Coworker**

DEDICATEDAGENCY

public static final java.lang.String **DEDICATEDAGENCY**

Constant value: **DedAge**

DEPARTMENT

public static final java.lang.String **DEPARTMENT**

Constant value: **Department**

MANAGEMENTCENTER

public static final java.lang.String **MANAGEMENTCENTER**

Constant value: **ManCen**

MEDICALADVISER

public static final java.lang.String **MEDICALADVISER**

Constant value: **MedAdv**

PRIMARYAGENCY

public static final java.lang.String **PRIMARYAGENCY**

Constant value: **PriAge**

PRINTQUEUE

public static final java.lang.String **PRINTQUEUE**

Constant value: **PriQue**

SECONDARYAGENCY

public static final java.lang.String **SECONDARYAGENCY**

Constant value: **SecAge**

SERVICE

public static final java.lang.String **SERVICE**

Constant value: **Service**

SOCIALSEAT

public static final java.lang.String **SOCIALSEAT**

Constant value: **SocialSeat**

SURGERY

public static final java.lang.String **SURGERY**

Constant value: **SurMedAdv**

(continued from last page)

TEAM

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

Constant value: **TEAM**

VIRTUALAGENCY

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

Constant value: **VirAge**

WORKSTATION

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

Constant value: **WorSta**

Constructors

GEDEntityType

```
public GEDEntityType()
```

be.gfdi.com.common.services.ged Class GEDEvent

java.lang.Object

└─be.gfdi.com.common.services.ged.GEDEvent

All Implemented Interfaces:

Serializable

public class **GEDEvent**
extends Object
implements Serializable

Represents an Event as viewed by the GED applications

Constructor Summary

public	GEDEvent ()
--------	------------------------------

Method Summary

Date	getCatalogDate () Date on which the event has been indexed in the system
String	getComment () Contains all the notes from DocCenter (concatenated) This has the same content as the "notelog" field
int	getContentAvailable () Says if an image is associated to the event (1 for true and 0 for false)
int	getContentCreatable () Specifies that the associated image can be created (1 for true and 0 for false).
String	getDcId ()
String	getDirection () Possible values are IN or OUT.
String	getDocType () Document type such as "K00011"
String	getDocTypeLabel () Document type label such as "Certificat medical"
GEDEventFolderInfo[]	getFolderInfos () The folder infos about a given event.
String	getFolderNumber () Deprecated. Use getFolderInfos() instead Number of the folder associated to the event

String	<u>getFolderType()</u> Deprecated. Use <i>getFolderInfos()</i> instead Type of folder associated to the event (such as "GASM")
String	<u>getFreeField()</u>
String	<u>getKey()</u> Event identifier
String	<u>getLanguage()</u> language of the associated image
String	<u>getMedium()</u> exchange medium (such as "Papier")
String	<u>getNoteLog()</u>
Date	<u>getPrintDate()</u> Date on which the event has been printed (outgoing letter)
String	<u>getScanBatchNumber()</u> The number given by the scanning batch to the event (can be used to archive document)
String	<u>getScanBatchNumberYear()</u>
String	<u>getScannerId()</u>
String	<u>getSecThirdExternalNumber()</u>
String	<u>getSecThirdFName()</u>
String	<u>getSecThirdId()</u>
String	<u>getSecThirdName()</u>
String	<u>getSecThirdNiss()</u>
String	<u>getSecThirdOrg()</u>
String	<u>getSecThirdPartyType()</u>
String	<u>getSecThirdRiziv()</u>
String	<u>getStatus()</u> Status of the event (such as "Catalogue" or "Fautif")
String	<u>getSubClass()</u> This field is no more used in Refoundation.
String	<u>getThirdExternalNumber()</u>
String	<u>getThirdFName()</u> The third first name for the event

String	<u>getThirdId()</u> The third unique number for this event
String	<u>getThirdManagementCenter()</u>
String	<u>getThirdName()</u> The third name for the event
String	<u>getThirdNiss()</u>
String	<u>getThirdOrg()</u>
String	<u>getThirdPartyType()</u>
String	<u>getThirdRiziv()</u>
String	<u>getWorkflowStatus()</u> The workflow status (one of <u>WorkflowStatus</u>)
void	<u>setCatalogDate()</u> (Date catalogDate)
void	<u>setComment()</u> (String comment) Contains all the notes from DocCenter (concatenated) This has the same content as the "notelog" field
void	<u>setContentAvailable()</u> (int contentAvailable)
void	<u>setContentCreatable()</u> (int contentCreatable)
void	<u>setDcId()</u> (String dcId)
void	<u>setDirection()</u> (String direction)
void	<u>setDocType()</u> (String docType)
void	<u>setDocTypeLabel()</u> (String docTypeLabel)
void	<u>setFolderInfos()</u> (<u>GEDEventFolderInfo[]</u> folderInfos) The folder infos about a given event.
void	<u>setFolderNumber()</u> (String folderNumber)
void	<u>setFolderType()</u> (String folderType)
void	<u>setFreeField()</u> (String freeField)
void	<u>setKey()</u> (String key)
void	<u>setLanguage()</u> (String language)

void	<u>setMedium</u> (String medium) The type of medium.
void	<u>setNoteLog</u> (String noteLog)
void	<u>setPrintDate</u> (Date printDate)
void	<u>setScanBatchNumber</u> (String scanBatchNumber)
void	<u>setScanBatchNumberYear</u> (String scanBatchNumberYear)
void	<u>setScannerId</u> (String scannerId)
void	<u>setSecThirdExternalNumber</u> (String secThirdExternalNumber)
void	<u>setSecThirdFName</u> (String secThirdFName)
void	<u>setSecThirdId</u> (String secThirdId)
void	<u>setSecThirdName</u> (String secThirdName)
void	<u>setSecThirdNiss</u> (String secThirdNiss)
void	<u>setSecThirdOrg</u> (String secThirdOrg)
void	<u>setSecThirdPartyType</u> (String secThirdPartyType)
void	<u>setSecThirdRiziv</u> (String secThirdRiziv)
void	<u>setStatus</u> (String status)
void	<u>setSubClass</u> (String subClass)
void	<u>setThirdExternalNumber</u> (String thirdExternalNumber)
void	<u>setThirdFName</u> (String thirdFName)
void	<u>setThirdId</u> (String thirdId)
void	<u>setThirdManagementCenter</u> (String thirdManagementCenter)
void	<u>setThirdName</u> (String thirdName)
void	<u>setThirdNiss</u> (String thirdNiss)
void	<u>setThirdOrg</u> (String thirdOrg)
void	<u>setThirdPartyType</u> (String thirdPartyType)

void	setThirdRiziv (String thirdRiziv)
void	setWorkflowStatus (String workflowStatus) The workflow status (one of WorkflowStatus)

Methods inherited from class java.lang.Object

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Constructors

GEDEvent

```
public GEDEvent()
```

Methods

getCatalogDate

```
public Date getCatalogDate()
```

Date on which the event has been indexed in the system

getComment

```
public String getComment()
```

Contains all the notes from DocCenter (concatenated) This has the same content as the "notelog" field

Returns:

comment

getContentAvailable

```
public int getContentAvailable()
```

Says if an image is associated to the event (1 for true and 0 for false)

getDirection

```
public String getDirection()
```

Possible values are IN or OUT.

getDocTypeLabel

```
public String getDocTypeLabel()
```

Document type label such as "Certificat medical"

(continued from last page)

getFolderNumber

```
public String getFolderNumber()
```

Deprecated. Use *getFolderInfos()* instead Number of the folder associated to the event

getFolderType

```
public String getFolderType()
```

Deprecated. Use *getFolderInfos()* instead Type of folder associated to the event (such as "GASM")

getKey

```
public String getKey()
```

Event identifier

getLanguage

```
public String getLanguage()
```

language of the associated image

getMedium

```
public String getMedium()
```

exchange medium (such as "Papier")

getStatus

```
public String getStatus()
```

Status of the event (such as "Catalogue" or "Fautif")

getSubClass

```
public String getSubClass()
```

This field is no more used in Refoundation.

getContentCreatable

```
public int getContentCreatable()
```

Specifies that the associated image can be created (1 for true and 0 for false). (i.e. it is true for IM event because the image can be generated in PDF)

getScanBatchNumber

```
public String getScanBatchNumber()
```

The number given by the scanning batch to the event (can be used to archive document)

getPrintDate

```
public Date getPrintDate()
```

Date on which the event has been printed (outgoing letter)

getThirdId

```
public final String getThirdId()
```

The third unique number for this event

getThirdFName

```
public final String getThirdFName()
```

The third first name for the event

getThirdName

```
public final String getThirdName()
```

The third name for the event

getDocType

```
public final String getDocType()
```

Document type such as "K00011"

setCatalogDate

```
public void setCatalogDate(Date catalogDate)
```

setComment

```
public void setComment(String comment)
```

Contains all the notes from DocCenter (concatenated) This has the same content as the "notelog" field

Parameters:

comment

setContentAvailable

```
public void setContentAvailable(int contentAvailable)
```

setDirection

```
public void setDirection(String direction)
```

setDocTypeLabel

```
public void setDocTypeLabel(String docTypeLabel)
```

setFolderNumber

```
public void setFolderNumber(String folderNumber)
```

setFolderType

```
public void setFolderType(String folderType)
```

setKey

```
public void setKey(String key)
```

setLanguage

```
public void setLanguage(String language)
```

setMedium

```
public void setMedium(String medium)
```

The type of medium. In @ {see DCDEvent} this is know as support. Possible values: 01 : paper 02 : fax 03 : electronic flow 04 : email 05 : infobox 06 : excel

Parameters:

medium - see possible values above

setStatus

```
public void setStatus(String status)
```

setSubClass

```
public void setSubClass(String subClass)
```

setPrintDate

```
public void setPrintDate(Date printDate)
```

setContentCreatable

```
public void setContentCreatable(int contentCreatable)
```

setThirdFName

```
public final void setThirdFName(String thirdFName)
```

setThirdId

```
public final void setThirdId(String thirdId)
```

setThirdName

```
public final void setThirdName(String thirdName)
```

setScanBatchNumber

```
public void setScanBatchNumber(String scanBatchNumber)
```

setDocType

```
public final void setDocType(String docType)
```

getScanBatchNumberYear

```
public String getScanBatchNumberYear()
```

setScanBatchNumberYear

```
public void setScanBatchNumberYear(String scanBatchNumberYear)
```

getSecThirdId

```
public String getSecThirdId()
```

setSecThirdId

```
public void setSecThirdId(String secThirdId)
```

(continued from last page)

getSecThirdName

```
public String getSecThirdName()
```

setSecThirdName

```
public void setSecThirdName(String secThirdName)
```

getSecThirdFName

```
public String getSecThirdFName()
```

setSecThirdFName

```
public void setSecThirdFName(String secThirdFName)
```

getNoteLog

```
public String getNoteLog()
```

setNoteLog

```
public void setNoteLog(String noteLog)
```

getScannerId

```
public String getScannerId()
```

setScannerId

```
public void setScannerId(String scannerId)
```

getThirdManagementCenter

```
public String getThirdManagementCenter()
```

(continued from last page)

setThirdManagementCenter

```
public void setThirdManagementCenter(String thirdManagementCenter)
```

getFreeField

```
public String getFreeField()
```

setFreeField

```
public void setFreeField(String freeField)
```

getDcId

```
public String getDcId()
```

setDcId

```
public void setDcId(String dcId)
```

getThirdExternalNumber

```
public String getThirdExternalNumber()
```

setThirdExternalNumber

```
public void setThirdExternalNumber(String thirdExternalNumber)
```

getThirdNiss

```
public String getThirdNiss()
```

setThirdNiss

```
public void setThirdNiss(String thirdNiss)
```

getSecThirdExternalNumber

```
public String getSecThirdExternalNumber()
```

(continued from last page)

setSecThirdExternalNumber

```
public void setSecThirdExternalNumber(String secThirdExternalNumber)
```

getSecThirdNiss

```
public String getSecThirdNiss()
```

setSecThirdNiss

```
public void setSecThirdNiss(String secThirdNiss)
```

getThirdRiziv

```
public String getThirdRiziv()
```

setThirdRiziv

```
public void setThirdRiziv(String thirdRiziv)
```

getSecThirdRiziv

```
public String getSecThirdRiziv()
```

setSecThirdRiziv

```
public void setSecThirdRiziv(String secThirdRiziv)
```

getThirdPartyType

```
public String getThirdPartyType()
```

setThirdPartyType

```
public void setThirdPartyType(String thirdPartyType)
```

(continued from last page)

getSecThirdPartyType

```
public String getSecThirdPartyType()
```

setSecThirdPartyType

```
public void setSecThirdPartyType(String secThirdPartyType)
```

getWorkflowStatus

```
public String getWorkflowStatus()
```

The workflow status (one of [WorkflowStatus](#))

setWorkflowStatus

```
public void setWorkflowStatus(String workflowStatus)
```

The workflow status (one of [WorkflowStatus](#))

getFolderInfos

```
public GEDEventFolderInfo\[\] getFolderInfos()
```

The folder infos about a given event. Useful vs folderNumber and folderType when many reference folders exists.

setFolderInfos

```
public void setFolderInfos(GEDEventFolderInfo\[\] folderInfos)
```

The folder infos about a given event. Useful vs folderNumber and folderType when many reference folders exists.

getThirdOrg

```
public String getThirdOrg()
```

Returns:

the thirdOrg

setThirdOrg

```
public void setThirdOrg(String thirdOrg)
```

Parameters:

thirdOrg - the thirdOrg to set

getSecThirdOrg

```
public String getSecThirdOrg()
```

(continued from last page)

Returns:

the secThirdOrg

setSecThirdOrg

```
public void setSecThirdOrg(String secThirdOrg)
```

Parameters:

secThirdOrg - the secThirdOrg to set

be.gfdi.com.common.services.ged Class GEDEventDetail

java.lang.Object

└─be.gfdi.com.common.services.ged.GEDEventDetail

All Implemented Interfaces:

Serializable

public class **GEDEventDetail**
extends Object
implements Serializable

Constructor Summary

public	GEDEventDetail()
--------	----------------------------------

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

GEDEventDetail

public **GEDEventDetail()**

be.gfdi.com.common.services.ged Class GEDEventFolderInfo

java.lang.Object

└─be.gfdi.com.common.services.ged.GEDEventFolderInfo

All Implemented Interfaces:

Serializable

public class **GEDEventFolderInfo**
extends Object
implements Serializable

Represents information about an individual reference folder for a given [GEDEvent](#)

Constructor Summary

public	GEDEventFolderInfo()
--------	--------------------------------------

Method Summary

Date	getEndDate() Represents the particular folder end date
String	getFolderNum() Represents the particular folder name or number
String	getFolderType() Represents the folder type which the event is contained in.
Date	getStartDate() Represents the particular folder start date
void	setEndDate() (Date endDate) Represents the particular folder end date
void	setFolderNum() (String folderNum) Represents the particular folder name or number
void	setFolderType() (String folderType) Represents the folder type which the event is contained in.
void	setStartDate() (Date startDate) Represents the particular folder start date

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

(continued from last page)

GEDEventFolderInfo

```
public GEDEventFolderInfo()
```

Methods

getFolderType

```
public String getFolderType()
```

Represents the folder type which the event is contained in.

setFolderType

```
public void setFolderType(String folderType)
```

Represents the folder type which the event is contained in.

getFolderNum

```
public String getFolderNum()
```

Represents the particular folder name or number

setFolderNum

```
public void setFolderNum(String folderNum)
```

Represents the particular folder name or number

getStartDate

```
public Date getStartDate()
```

Represents the particular folder start date

setStartDate

```
public void setStartDate(Date startDate)
```

Represents the particular folder start date

getEndDate

```
public Date getEndDate()
```

Represents the particular folder end date

setEndDate

```
public void setEndDate(Date endDate)
```

Represents the particular folder end date

be.gfdi.com.common.services.ged Class GEDEventRouting

java.lang.Object

└-be.gfdi.com.common.services.ged.GEDEventRouting

All Implemented Interfaces:

Serializable

public class **GEDEventRouting**
extends Object
implements Serializable

Constructor Summary

public	GEDEventRouting()
--------	-----------------------------------

Method Summary

String	getDocType()
String	getEngine()
Long	getId()
int	getProcessDuration()
String	getProcessName()
void	setDocType(String docType)
void	setEngine(String engine)
void	setId(Long id)
void	setProcessDuration(int processDuration)
void	setProcessName(String processName)

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

(continued from last page)

GEDEventRouting

```
public GEDEventRouting()
```

Methods

getId

```
public Long getId()
```

setId

```
public void setId(Long id)
```

getDocType

```
public String getDocType()
```

setDocType

```
public void setDocType(String docType)
```

getProcessName

```
public String getProcessName()
```

setProcessName

```
public void setProcessName(String processName)
```

getEngine

```
public String getEngine()
```

setEngine

```
public void setEngine(String engine)
```


(continued from last page)

getProcessDuration

```
public int getProcessDuration()
```

setProcessDuration

```
public void setProcessDuration(int processDuration)
```

be.gfdi.com.common.services.ged Class GEDEventRoutingConverter

java.lang.Object

└─be.gfdi.com.common.services.ged.GEDEventRoutingConverter

public class **GEDEventRoutingConverter**
extends Object

Utility class for the conversion of the "public" class "GEDEventRouting".

Constructor Summary

public	GEDEventRoutingConverter()
--------	--

Method Summary

static ProcessCreationDecision	convertIn (GEDEventRouting ged) Converts from the "public" object to the "private" object.
static ProcessCreationDecision	convertIn (GEDEventRouting ged, ProcessCreationDecision pcd) Filled the data of the "private" object with the data of the "public" object.
static GEDEventRouting	convertOut (ProcessCreationDecision pcd) Converts from the "private" object to the "public" one.

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

GEDEventRoutingConverter

public **GEDEventRoutingConverter**()

Methods

convertOut

public static [GEDEventRouting](#) **convertOut**(ProcessCreationDecision pcd)

Converts from the "private" object to the "public" one. If the object passed as argument is 'null', the result will be 'null' also.

Parameters:

pcd - The ProcessCreationDecision object to convert.

Returns:

(continued from last page)

The public object.

convertIn

```
public static ProcessCreationDecision convertIn(GEDEventRouting ged)
```

Converts from the "public" object to the "private" object. If the object passed as argument is 'null', the result will be 'null' also.

Parameters:

ged - The "public" object to convert.

Returns:

The "private" object filled in with the data passed as argument.

convertIn

```
public static ProcessCreationDecision convertIn(GEDEventRouting ged,  
        ProcessCreationDecision pcd)
```

Filled the data of the "private" object with the data of the "public" object. This avoids filling the key data of the object.

Parameters:

ged - The "public" object to convert.

pcd - The "private" object to fill in.

Returns:

The "private" object filled in.

be.gfdi.com.common.services.ged Class GEDEXecutionCode

java.lang.Object

↳ be.gfdi.com.common.services.ged.GEDEXecutionCode

public abstract class **GEDEXecutionCode**
extends Object

Field Summary

public static final	BATCH BATCH = new ExecutionCode("1"); IMMEDIATE = new ExecutionCode("2"); BATCHNOPRINT = new ExecutionCode("3"); IMMEDIATENOPRINT = new ExecutionCode("4"); Value: 1
public static final	BATCHNOPRINT Value: 3
public static final	IMMEDIATE Value: 2
public static final	IMMEDIATENOPRINT Value: 4

Constructor Summary

public	GEDEXecutionCode()
--------	------------------------------------

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Fields

BATCH

public static final java.lang.String **BATCH**

BATCH = new ExecutionCode("1"); IMMEDIATE = new ExecutionCode("2"); BATCHNOPRINT = new ExecutionCode("3"); IMMEDIATENOPRINT = new ExecutionCode("4");
Constant value: **1**

IMMEDIATE

public static final java.lang.String **IMMEDIATE**

(continued from last page)

Constant value: **2**

BATCHNOPRINT

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

Constant value: **3**

IMMEDIATENOPRINT

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

Constant value: **4**

Constructors

GEDExecutionCode

```
public GEDExecutionCode()
```

be.gfdi.com.common.services.ged Class GEDFolderType

java.lang.Object

└-be.gfdi.com.common.services.ged.GEDFolderType

All Implemented Interfaces:

Serializable

public class **GEDFolderType**
extends Object
implements Serializable

Constructor Summary

public	GEDFolderType()
--------	---------------------------------

Method Summary

String	getCode()
String	getLabelFr()
String	getLabelGe()
String	getLabelNl()
String	getOrganization()
void	setCode(String code)
void	setLabelFr(String labelFr)
void	setLabelGe(String labelGe)
void	setLabelNl(String labelNl)
void	setOrganization(String organization)

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

(continued from last page)

GEDFolderType

```
public GEDFolderType()
```

Methods

getCode

```
public String getCode()
```

Returns:

the code

setCode

```
public void setCode(String code)
```

Parameters:

code - the code to set

getLabelFr

```
public String getLabelFr()
```

Returns:

the labelFr

setLabelFr

```
public void setLabelFr(String labelFr)
```

Parameters:

labelFr - the labelFr to set

getLabelNl

```
public String getLabelNl()
```

Returns:

the labelNl

setLabelNl

```
public void setLabelNl(String labelNl)
```

(continued from last page)

Parameters:labelNl - the labelNl to set

getLabelGe

```
public String getLabelGe()
```

Returns:the labelGe

setLabelGe

```
public void setLabelGe(String labelGe)
```

Parameters:labelGe - the labelGe to set

getOrganization

```
public String getOrganization()
```

Returns:the organization

setOrganization

```
public void setOrganization(String organization)
```

Parameters:organization - the organization to set

be.gfdi.com.common.services.ged Class GEDPartyType

java.lang.Object

└-be.gfdi.com.common.services.ged.GEDPartyType

public abstract class **GEDPartyType**
extends Object

constants for PartyTypeCode * INSTITUTION ==> partyCode = "330"; MUTUALITYPERSON ==> partyCode = "201"; PROVIDER ==> partyCode = "310"; EMPLOYER ==> partyCode = "320"; COWORKER ==> partyCode = "101"; MEDICALADVISER ==> partyCode = "102"; SOCIALSEAT ==> partyCode = "110"; MANAGEMENTCENTER ==> partyCode = "111"; GENERALAGENCY ==> partyCode = "120"; PRINCIPALAGENCY ==> partyCode = "121"; SECONDARYAGENCY ==> partyCode = "122"; VIRTUALAGENCY ==> partyCode = "123"; DEDICATEDAGENCY ==> partyCode = "124"; DEPARTMENT ==> partyCode = "131"; SERVICE ==> partyCode = "132"; TEAM ==> partyCode = "133"; MEDICALOFFICE ==> partyCode = "180"; PRINTQUEUE ==> partyCode = "192"; WORKSTATION ==> partyCode = "191";

Field Summary	
public static final	COWORKER Value: COWORKER
public static final	DEDICATEDAGENCY Value: DEDICATEDAGENCY
public static final	DEPARTMENT Value: DEPARTMENT
public static final	EMPLOYER Value: EMPLOYER
public static final	GENERALAGENCY Value: GENERALAGENCY
public static final	INSTITUTION Value: INSTITUTION
public static final	MANAGEMENTCENTER Value: MANAGEMENTCENTER
public static final	MEDICALADVISER Value: MEDICALADVISER
public static final	MEDICALOFFICE Value: MEDICALOFFICE
public static final	MUTUALITYPERSON Value: MUTUALITYPERSON

public static final	PRINCIPALAGENCY Value: PRINCIPALAGENCY
public static final	PRINTQUEUE Value: PRINTQUEUE
public static final	PROVIDER Value: PROVIDER
public static final	SECONDARYAGENCY Value: SECONDARYAGENCY
public static final	SERVICE Value: SERVICE
public static final	SOCIALSEAT Value: SOCIALSEAT
public static final	TEAM Value: TEAM
public static final	VIRTUALAGENCY Value: VIRTUALAGENCY
public static final	WORKSTATION Value: WORKSTATION

Constructor Summary

public	GEDPartyType ()
--------	----------------------------------

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Fields

EMPLOYER

public static final java.lang.String **EMPLOYER**

Constant value: **EMPLOYER**

PROVIDER

public static final java.lang.String **PROVIDER**

Constant value: **PROVIDER**

MUTUALITYPERSON

public static final java.lang.String **MUTUALITYPERSON**

Constant value: **MUTUALITYPERSON**

INSTITUTION

public static final java.lang.String **INSTITUTION**

Constant value: **INSTITUTION**

COWORKER

public static final java.lang.String **COWORKER**

Constant value: **COWORKER**

MEDICALADVISER

public static final java.lang.String **MEDICALADVISER**

Constant value: **MEDICALADVISER**

SOCIALSEAT

public static final java.lang.String **SOCIALSEAT**

Constant value: **SOCIALSEAT**

MANAGEMENTCENTER

public static final java.lang.String **MANAGEMENTCENTER**

Constant value: **MANAGEMENTCENTER**

GENERALAGENCY

public static final java.lang.String **GENERALAGENCY**

Constant value: **GENERALAGENCY**

PRINCIPALAGENCY

public static final java.lang.String **PRINCIPALAGENCY**

Constant value: **PRINCIPALAGENCY**

(continued from last page)

SECONDARYAGENCY

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

Constant value: **SECONDARYAGENCY**

VIRTUALAGENCY

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

Constant value: **VIRTUALAGENCY**

DEDICATEDAGENCY

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

Constant value: **DEDICATEDAGENCY**

DEPARTMENT

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

Constant value: **DEPARTMENT**

TEAM

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

Constant value: **TEAM**

MEDICALOFFICE

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

Constant value: **MEDICALOFFICE**

PRINTQUEUE

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

Constant value: **PRINTQUEUE**

WORKSTATION

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

Constant value: **WORKSTATION**

SERVICE

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

(continued from last page)

Constant value: **SERVICE**

Constructors

GEDPartyType

```
public GEDPartyType()
```

be.gfdi.com.common.services.ged Class GEDPrintDocumentRequest

java.lang.Object

└-be.gfdi.com.common.services.ged.GEDPrintDocumentRequest

All Implemented Interfaces:
Serializable

public class **GEDPrintDocumentRequest**
extends Object
implements Serializable

Constructor Summary

public	GEDPrintDocumentRequest()
--------	---

Method Summary

String	getBenCodeType() get code type from party reference of the beneficiary.
String	getBenCodeValue() get value from party reference of the beneficiary.
String	getBenOrg() get the organisation from the party reference of the beneficiary.
String	getBenParType() get type from party reference of the beneficiary.
String	getBody() get the email body
GEDPrintParameter[]	getBodyParameters() get document print parameters
String	getBox() get the box(RecipientAddress)
String	getBusinessRef() get the business reference
String	getCanvasName() get Canvas name Optional : default to the canvas linked to the document type
String	getCanvasStatus() get the CanvasStatus The possible values for the type are: (one of GEDCanvasStatus)
String	getCity() get the city(RecipientAddress)

String	<u>getCo()</u> get the co (RecipientAddress)
String	<u>getContactInfoTypeCode()</u> get the Contact info type.
Integer	<u>getCopyNumber()</u> get the number of copy
String	<u>getCountry()</u> get the country(RecipientAddress)
String	<u>getDocTypeName()</u> get docTypeName of the document type.
String	<u>getEmailFromAddress()</u> get the address email from
String	<u>getEmailFromName()</u> get the name email from
String	<u>getEmailToAdress()</u> get the address email To
String	<u>getEmailToName()</u> get the name email To
String	<u>getExecutionCode()</u> get the execution code The possible values are: (one of <u>GEDExecutionCode</u>)
String	<u>getFolderRef()</u> get Folder (or dossier) number.
String	<u>getFolderType()</u> get Folder (or dossier) type.
String	<u>getLanguage()</u> get the language Optional : default to a value specified by OutgoingLetterDocType.language (either language of the Holder or language of the organization).
byte[]	<u>getLogo()</u> get the logo email
String	<u>getLogoFilePath()</u> get the email logo file path
String	<u>getMancenId()</u> get the id from the entity reference to the Management Center Optional : default to the management center of the holder
String	<u>getMancenOrg()</u> get organisation from the entity reference to the Management Center Optional : default to the management center of the holder
String	<u>getMancenType()</u> get type from the entity reference to the Management Center Optional : default to the management center of the holder The possible values are: (one of <u>GEDEntityType</u>)

String	<u>getMedAdvId()</u> get the Id from the entity reference to the Medical Adviser Optional
String	<u>getMedAdvOrg()</u> get organisation from entity reference to the Medical Adviser Optional
String	<u>getMedAdvType()</u> get the type from the entity reference to the Medical Adviser Optional The possible values for the type are: (one of <u>GEDEntityType</u>)
String	<u>getNumber()</u> get the number(RecipientAddress)
String	<u>getOwnerCodeType()</u> get code type from party reference of the holder.
String	<u>getOwnerCodeValue()</u> get value from party reference of the holder.
String	<u>getOwnerOrg()</u> get the organisation from the party reference of the holder.
String	<u>getOwnerParType()</u> get type from party reference of the holder.
String	<u>getPayerCodeType()</u> get code type from party reference of the payer.
String	<u>getPayerCodeValue()</u> get value from party reference of the payer.
String	<u>getPayerOrg()</u> get the organisation from the party reference of the payer.
String	<u>getPayerParType()</u> get type from party reference of the payer.
String	<u>getPostalCode()</u> get the postal code (RecipientAddress)
String	<u>getPrinterId()</u> get the identification of the printer in ORG Optional : default to the printer calculated with the "Starting Point Algorithm"
String	<u>getProviderCodeType()</u> get code type from party reference of the provider.
String	<u>getProviderCodeValue()</u> get value from party reference of the provider.
String	<u>getProviderOrg()</u> get the organisation from the party reference of the provider.
String	<u>getProviderParType()</u> get type from party reference of the provider.
String	<u>getRecipient()</u> get the recipient(RecipientAddress)

String	<u>getRecipientCodeType()</u> get code type from party reference of the recipient.
String	<u>getRecipientCodeValue()</u> get value from party reference of the recipient.
String	<u>getRecipientOrg()</u> get the organisation from the party reference of the recipient.
String	<u>getRecipientParType()</u> get type from party reference of the recipient.
String	<u>getRequestCsoId()</u> get the request CSO Id
String	<u>getShipper1Id()</u> get the id from the entity reference to the shipping entity 1 Optional : default to the one calculated by the "Starting Point Algorithm"
String	<u>getShipper1Org()</u> get the organisation from entity reference to the shipping entity 1 Optional : default to the one calculated by the "Starting Point Algorithm"
String	<u>getShipper1Type()</u> get the type from the entity reference to the shipping entity 1 Optional : default to the one calculated by the "Starting Point Algorithm" The possible values are: (one of <u>GEDEntityType()</u>)
String	<u>getShipper2Id()</u> get the id from the entity reference to the shipping entity 2 Optional : default to the one calculated by the "Starting Point Algorithm"
String	<u>getShipper2Org()</u> get the organisation from entity reference to the shipping entity 2 Optional : default to the one calculated by the "Starting Point Algorithm"
String	<u>getShipper2Type()</u> get the type from the entity reference to the shipping entity 2 Optional : default to the one calculated by the "Starting Point Algorithm" The possible values are: (one of <u>GEDEntityType()</u>)
String	<u>getShipperRCId()</u> get the id from the entity reference to the shipping entity RC Optional : default to the one calculated by the "Starting Point Algorithm"
String	<u>getShipperRCOrg()</u> get the organisation from entity reference to the shipping entity RC Optional : default to the one calculated by the "Starting Point Algorithm"
String	<u>getShipperRCType()</u> get the type from the entity reference to the shipping entity RC Optional : default to the one calculated by the "Starting Point Algorithm" The possible values are: (one of <u>GEDEntityType()</u>)
String	<u>getSigner1Id()</u> get the id from the entity reference to the Signer1 entity Optional : default to the one calculated by the "Starting Point Algorithm"
String	<u>getSigner1Org()</u> get the organisation from entity reference to the Signer1 entity Optional : default to the one calculated by the "Starting Point Algorithm"

String	<u>getSigner1Type()</u> get the type from the entity reference to the Signer1 entity Optional : default to the one calculated by the "Starting Point Algorithm" The possible values are: (one of <u>GEDEntityType</u>)
String	<u>getSigner2Id()</u> get the id from the entity reference to the Signer2 entity Optional : default to the one calculated by the "Starting Point Algorithm"
String	<u>getSigner2Org()</u> get the organisation from entity reference to the Signer2 entity Optional : default to the one calculated by the "Starting Point Algorithm"
String	<u>getSigner2Type()</u> get the type from the entity reference to the Signer2 entity Optional : default to the one calculated by the "Starting Point Algorithm" The possible values are: (one of <u>GEDEntityType</u>)
String	<u>getStreet()</u> get the street(RecipientAddress)
String	<u>getSubject()</u> get the name email subject
String	<u>getSurMedAdvId()</u> get the Id from the entity reference to the Medical Adviser Office Optional
String	<u>getSurMedAdvOrg()</u> get the organisation from the entity reference to the Medical Adviser Office Optional
String	<u>getSurMedAdvType()</u> get type from the entity reference to the Medical Adviser Office Optional The possible values are: (one of <u>GEDEntityType</u>)
String	<u>getUserId()</u> get the user id (logon name) Mandatory
String	<u>getVersionNbr()</u> get the Canvas version number Optional : default to the last version (with production status)
String	<u>getWorkingEntityId()</u> get the Working entity id of the user Mandatory
String	<u>getWorkingEntityOrg()</u> get the Working organisation entity of the user Mandatory
String	<u>getWorkingEntityType()</u> get the Working entity type of the user Mandatory
String	<u>getWorkStation()</u> get workstation from the user (hostname of the workstation).
Boolean	<u>isEMail()</u> is the email yes or not
Boolean	<u>isHoldSpool()</u> if the spool is hold Optional : default to the value specified in the printer (in ORG)
Boolean	<u>isRecommend()</u> get Registered Mail.

Boolean	<u>isSaveSpool</u> () if the spool is saved Optional : default to the value specified in the printer (in ORG)
void	<u>setBenCodeType</u> (String benCodeType) set code type to party reference of the beneficiary.
void	<u>setBenCodeValue</u> (String benCodeValue) set value to party reference of the beneficiary.
void	<u>setBenOrg</u> (String benOrg) set the organisation to the party reference of the beneficiary.
void	<u>setBenParType</u> (String benParType) set type to party reference of the payer.
void	<u>setBody</u> (String body) set the email body
void	<u>setBodyParameters</u> (<u>GEDPrintParameter[]</u> bodyParameters) set document print parameters
void	<u>setBox</u> (String box) set the box(RecipientAddress)
void	<u>setBusinessRef</u> (String businessRef) set the business reference
void	<u>setCanvasName</u> (String canvasName) set Canvas name Optional : default to the canvas linked to the document type
void	<u>setCanvasStatus</u> (String canvasStatus) set the CanvasStatus The possible values for the type are: (one of <u>GEDCanvasStatus</u>)
void	<u>setCity</u> (String city) set the city(RecipientAddress)
void	<u>setCo</u> (String co) set the co (RecipientAddress)
void	<u>setContactInfoTypeCode</u> (String contactInfoTypeCode) set the Contact info type.
void	<u>setCopyNumber</u> (Integer copyNumber) set the number of copy
void	<u>setCountry</u> (String country) get the country(RecipientAddress)
void	<u>setDocTypeName</u> (String docTypeName) set docTypeName of the document type.
void	<u>setEmail</u> (Boolean mail) set the email to yes or not
void	<u>setEmailFromAddress</u> (String emailFromAddress) set the address email from
void	<u>setEmailFromName</u> (String emailFromName) set the name email from

void	<u>setEmailToAdress</u> (String emailToAdress) set the address email To
void	<u>setEmailToName</u> (String emailToName) set the name email To
void	<u>setExecutionCode</u> (String executionCode) set the execution code The possible values are: (one of <u>GEDExecutionCode</u>)
void	<u>setFolderRef</u> (String folderRef) set Folder (or dossier) number.
void	<u>setFolderType</u> (String folderType) set Folder (or dossier) type.
void	<u>setHoldSpool</u> (Boolean holdSpool) set the spool to hold Optional : default to the value specified in the printer (in ORG)
void	<u>setLanguage</u> (String language) set the language
void	<u>setLogo</u> (byte[] logo) set the logo email
void	<u>setLogoFilePath</u> (String logoFilePath) set the email logo fil path
void	<u>setMancenId</u> (String mancenId) set the id to the entity reference to the Management Center Optional : default to the management center of the holder
void	<u>setMancenOrg</u> (String mancenOrg) set the organisation to the entity reference to the Management Center Optional : default to the management center of the holder
void	<u>setMancenType</u> (String mancenType) set the type to entity reference to the Management Center Optional : default to the management center of the holder The possible values are: (one of <u>GEDEntityType</u>)
void	<u>setMedAdvId</u> (String medAdvId) set the id to entity reference to the Medical Adviser Optional
void	<u>setMedAdvOrg</u> (String medAdvOrg) set organisation to the entity reference to the Medical Adviser Optional
void	<u>setMedAdvType</u> (String medAdvType) set the type for entity reference to the Medical Adviser Optional The possible values for the type are: (one of <u>GEDEntityType</u>)
void	<u>setNumber</u> (String number) set the number(RecipientAddress)
void	<u>setOwnerCodeType</u> (String ownerCodeType) set code type to party reference of the holder.
void	<u>setOwnerCodeValue</u> (String ownerCodeValue) set value to party reference of the holder.
void	<u>setOwnerOrg</u> (String ownerOrg) set the organisation to the party reference of the holder.

void	<u>setOwnerParType</u> (String ownerParType) set type to party reference of the holder.
void	<u>setPayerCodeType</u> (String payerCodeType) set code type to party reference of the payer.
void	<u>setPayerCodeValue</u> (String payerCodeValue) set value to party reference of the payer.
void	<u>setPayerOrg</u> (String payerOrg) set the organisation to the party reference of the payer.
void	<u>setPayerParType</u> (String payerParType) set type to party reference of the payer.
void	<u>setPostalCode</u> (String postalCode) set the postal code(RecipientAddress)
void	<u>setPrinterId</u> (String printerId) set the identification of the printer in ORG Optional : default to the printer calculated with the "Starting Point Algorithm"
void	<u>setProviderCodeType</u> (String providerCodeType) set code type to party reference of the provider.
void	<u>setProviderCodeValue</u> (String providerCodeValue) set value to party reference of the provider.
void	<u>setProviderOrg</u> (String providerOrg) set the organisation to the party reference of the provider.
void	<u>setProviderParType</u> (String providerParType) set type to party reference of the provider.
void	<u>setRecipient</u> (String recipient) set the recipient(RecipientAddress)
void	<u>setRecipientCodeType</u> (String recipientCodeType) set codeType to the party reference of the recipient.
void	<u>setRecipientCodeValue</u> (String recipientCodeValue) set value to the party reference of the recipient.
void	<u>setRecipientOrg</u> (String recipientOrg) set the organisation to the party reference of the recipient.
void	<u>setRecipientParType</u> (String recipientParType) set type to the party reference of the recipient.
void	<u>setRecommend</u> (Boolean recommend) set Registered Mail.
void	<u>setRequestCsoId</u> (String requestCsoId) set the request CSO id
void	<u>setSaveSpool</u> (Boolean saveSpool) set the spool to saved Optional : default to the value specified in the printer (in ORG)

void	<u>setShipper1Id</u> (String shipper1Id) set the id to entity reference to the shipping entity 1 Optional : default to the one calculated by the "Starting Point Algorithm"
void	<u>setShipper1Org</u> (String shipper1Org) set the organisation to entity reference to the shipping entity 1 Optional : default to the one calculated by the "Starting Point Algorithm"
void	<u>setShipper1Type</u> (String shipper1Type) set the type to entity reference to the shipping entity 1 Optional : default to the one calculated by the "Starting Point Algorithm" The possible values are: (one of <u>GEDEntityType</u>)
void	<u>setShipper2Id</u> (String shipper2Id) set the id to entity reference to the shipping entity 2 Optional : default to the one calculated by the "Starting Point Algorithm"
void	<u>setShipper2Org</u> (String shipper2Org) set the organisation to entity reference to the shipping entity 2 Optional : default to the one calculated by the "Starting Point Algorithm"
void	<u>setShipper2Type</u> (String shipper2Type) set the type to entity reference to the shipping entity 2 Optional : default to the one calculated by the "Starting Point Algorithm" The possible values are: (one of <u>GEDEntityType</u>)
void	<u>setShipperRCId</u> (String shipperRCId) set the id to entity reference to the shipping entity RC Optional : default to the one calculated by the "Starting Point Algorithm"
void	<u>setShipperRCOrg</u> (String shipperRCOrg) set the organisation to entity reference to the shipping entity RC Optional : default to the one calculated by the "Starting Point Algorithm"
void	<u>setShipperRCType</u> (String shipperRCType) set the type to entity reference to the shipping entity RC Optional : default to the one calculated by the "Starting Point Algorithm" The possible values are: (one of <u>GEDEntityType</u>)
void	<u>setSigner1Id</u> (String signer1Id) set the id to entity reference to the Signer1 entity Optional : default to the one calculated by the "Starting Point Algorithm"
void	<u>setSigner1Org</u> (String signer1Org) set the organisation to entity reference to the Signer1 entity Optional : default to the one calculated by the "Starting Point Algorithm"
void	<u>setSigner1Type</u> (String signer1Type) set the type to entity reference to the Signer1 entity Optional : default to the one calculated by the "Starting Point Algorithm" The possible values are: (one of <u>GEDEntityType</u>)
void	<u>setSigner2Id</u> (String signer2Id) set the id to entity reference to the Signer2 entity Optional : default to the one calculated by the "Starting Point Algorithm"
void	<u>setSigner2Org</u> (String signer2Org) set the organisation to entity reference to the Signer2 entity Optional : default to the one calculated by the "Starting Point Algorithm"
void	<u>setSigner2Type</u> (String signer2Type) set the type to entity reference to the Signer2 entity Optional : default to the one calculated by the "Starting Point Algorithm" The possible values are: (one of <u>GEDEntityType</u>)

void	<u>setStreet</u> (String street) set the street(RecipientAddress)
void	<u>setSubject</u> (String subject) set the email subject
void	<u>setSurMedAdvId</u> (String surMedAdvId) set the Id to the entity reference to the Medical Adviser Office Optional
void	<u>setSurMedAdvOrg</u> (String surMedAdvOrg) set the organisation to the entity reference to the Medical Adviser Office Optional
void	<u>setSurMedAdvType</u> (String surMedAdvType) set type to the entity reference to the Medical Adviser Office Optional The possible values are: (one of <u>GEDEntityType</u>)
void	<u>setUserId</u> (String userId) set the user id (logon name) Mandatory
void	<u>setVersionNbr</u> (String versionNbr) set Canvas version number Optional : default to the last version (with production status)
void	<u>setWorkingEntity</u> (String workingEntityId) set the Working entity id of the user Mandatory
void	<u>setWorkingEntityOrg</u> (String workingEntityOrg) set the Working organisation entity of the user Mandatory
void	<u>setWorkingEntityType</u> (String workingEntityType) set the Working entity type of the user Mandatory
void	<u>setWorkStation</u> (String workStation) set workstation to the user (hostname of the workstation).

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

GEDPrintDocumentRequest

```
public GEDPrintDocumentRequest()
```

Methods

getDocTypeName

```
public String getDocTypeName()
```

get docTypeName of the document type. Mandatory

(continued from last page)

setDocTypeName

```
public void setDocTypeName(String docTypeName)
```

set docTypeName of the document type. Mandatory

getRecipientOrg

```
public String getRecipientOrg()
```

get the organisation from the party reference of the recipient. Mandatory

setRecipientOrg

```
public void setRecipientOrg(String recipientOrg)
```

set the organisation to the party reference of the recipient. Mandatory

getRecipientCodeType

```
public String getRecipientCodeType()
```

get code type from party reference of the recipient. Mandatory The possible values for the type are: (one of [ThirdCodeType](#))

setRecipientCodeType

```
public void setRecipientCodeType(String recipientCodeType)
```

set codeType to the party reference of the recipient. Mandatory The possible values for the type are: (one of [ThirdCodeType](#))

getRecipientCodeValue

```
public String getRecipientCodeValue()
```

get value from party reference of the recipient. Mandatory

setRecipientCodeValue

```
public void setRecipientCodeValue(String recipientCodeValue)
```

set value to the party reference of the recipient. Mandatory

getRecipientParType

```
public String getRecipientParType()
```

get type from party reference of the recipient. Mandatory The possible values for the type are: (one of [GEDPartyType](#))

setRecipientParType

```
public void setRecipientParType(String recipientParType)
```

set type to the party reference of the recipient. Mandatory The possible values for the type are: (one of [GEDPartyType](#))

(continued from last page)

getOwnerOrg

```
public String getOwnerOrg()
```

get the organisation from the party reference of the holder. Mandatory

setOwnerOrg

```
public void setOwnerOrg(String ownerOrg)
```

set the organisation to the party reference of the holder. Mandatory

getOwnerCodeType

```
public String getOwnerCodeType()
```

get code type from party reference of the holder. Mandatory The possible values for the type are: (one of [ThirdCodeType](#))

setOwnerCodeType

```
public void setOwnerCodeType(String ownerCodeType)
```

set code type to party reference of the holder. Mandatory The possible values for the type are: (one of [ThirdCodeType](#))

getOwnerCodeValue

```
public String getOwnerCodeValue()
```

get value from party reference of the holder. Mandatory

setOwnerCodeValue

```
public void setOwnerCodeValue(String ownerCodeValue)
```

set value to party reference of the holder. Mandatory

getOwnerParType

```
public String getOwnerParType()
```

get type from party reference of the holder. Mandatory The possible values for the type are: (one of [GEDPartyType](#))

setOwnerParType

```
public void setOwnerParType(String ownerParType)
```

set type to party reference of the holder. Mandatory The possible values for the type are: (one of [GEDPartyType](#))

getFolderRef

```
public String getFolderRef()
```

get Folder (or dossier) number. Optional : default to FolderRef.DEFAULT_REFERENCE

setFolderRef

```
public void setFolderRef(String folderRef)
```

(continued from last page)

set Folder (or dossier) number. Optional : default to FolderRef.DEFAULT_REFERENCE

getUserId

```
public String getUserId()
```

get the user id (logon name) Mandatory

setUserId

```
public void setUserId(String userId)
```

set the user id (logon name) Mandatory

getWorkStation

```
public String getWorkStation()
```

get workstation from the user (hostname of the workstation). Mandatory

setWorkStation

```
public void setWorkStation(String workStation)
```

set workstation to the user (hostname of the workstation). Mandatory

getWorkingEntityOrg

```
public String getWorkingEntityOrg()
```

get the Working organisation entity of the user Mandatory

setWorkingEntityOrg

```
public void setWorkingEntityOrg(String workingEntityOrg)
```

set the Working organisation entity of the user Mandatory

getWorkingEntityId

```
public String getWorkingEntityId()
```

get the Working entity id of the user Mandatory

setWorkingEntity

```
public void setWorkingEntity(String workingEntityId)
```

set the Working entity id of the user Mandatory

getWorkingEntityType

```
public String getWorkingEntityType()
```

get the Working entity type of the user Mandatory

(continued from last page)

setWorkingEntityType

```
public void setWorkingEntityType(String workingEntityType)
```

set the Working entity type of the user Mandatory

getCopyNumber

```
public Integer getCopyNumber()
```

get the number of copy

setCopyNumber

```
public void setCopyNumber(Integer copyNumber)
```

set the number of copy

getLanguage

```
public String getLanguage()
```

get the language Optional : default to a value specified by `OutgoingLetterDocType.language` (either language of the Holder or language of the organization).

setLanguage

```
public void setLanguage(String language)
```

set the language

getBodyParameters

```
public GEDPrintParameter\[\] getBodyParameters()
```

get document print parameters

setBodyParameters

```
public void setBodyParameters(GEDPrintParameter\[\] bodyParameters)
```

set document print parameters

getBenOrg

```
public String getBenOrg()
```

get the organisation from the party reference of the beneficiary. Mandatory

setBenOrg

```
public void setBenOrg(String benOrg)
```

set the organisation to the party reference of the beneficiary. Mandatory

getBenCodeType

```
public String getBenCodeType()
```

(continued from last page)

get code type from party reference of the beneficiary. Mandatory The possible values for the type are: (one of [ThirdCodeType](#))

setBenCodeType

```
public void setBenCodeType(String benCodeType)
```

set code type to party reference of the beneficiary. Mandatory The possible values for the type are: (one of [ThirdCodeType](#))

getBenCodeValue

```
public String getBenCodeValue()
```

get value from party reference of the beneficiary. Mandatory

setBenCodeValue

```
public void setBenCodeValue(String benCodeValue)
```

set value to party reference of the beneficiary. Mandatory

getBenParType

```
public String getBenParType()
```

get type from party reference of the beneficiary. Mandatory The possible values for the type are: (one of [GEDPartyType](#))

setBenParType

```
public void setBenParType(String benParType)
```

set type to party reference of the payer. Mandatory The possible values for the type are: (one of [GEDPartyType](#))

getPayerOrg

```
public String getPayerOrg()
```

get the organisation from the party reference of the payer. Mandatory

setPayerOrg

```
public void setPayerOrg(String payerOrg)
```

set the organisation to the party reference of the payer. Mandatory

getPayerCodeType

```
public String getPayerCodeType()
```

get code type from party reference of the payer. Mandatory The possible values for the type are: (one of [ThirdCodeType](#))

setPayerCodeType

```
public void setPayerCodeType(String payerCodeType)
```

set code type to party reference of the payer. Mandatory The possible values for the type are: (one of [ThirdCodeType](#))

getPayerCodeValue

```
public String getPayerCodeValue()
```

get value from party reference of the payer. Mandatory

setPayerCodeValue

```
public void setPayerCodeValue(String payerCodeValue)
```

set value to party reference of the payer. Mandatory

getPayerParType

```
public String getPayerParType()
```

get type from party reference of the payer. Mandatory The possible values for the type are: (one of [GEDPartyType](#))

setPayerParType

```
public void setPayerParType(String payerParType)
```

set type to party reference of the payer. Mandatory The possible values for the type are: (one of [GEDPartyType](#))

getProviderOrg

```
public String getProviderOrg()
```

get the organisation from the party reference of the provider. Mandatory

setProviderOrg

```
public void setProviderOrg(String providerOrg)
```

set the organisation to the party reference of the provider. Mandatory

getProviderCodeType

```
public String getProviderCodeType()
```

get code type from party reference of the provider. Mandatory The possible values for the type are: (one of [ThirdCodeType](#))

setProviderCodeType

```
public void setProviderCodeType(String providerCodeType)
```

set code type to party reference of the provider. Mandatory The possible values for the type are: (one of [ThirdCodeType](#))

getProviderCodeValue

```
public String getProviderCodeValue()
```

get value from party reference of the provider. Mandatory

(continued from last page)

setProviderCodeValue

```
public void setProviderCodeValue(String providerCodeValue)
```

set value to party reference of the provider. Mandatory

getProviderParType

```
public String getProviderParType()
```

get type from party reference of the provider. Mandatory The possible values for the type are: (one of [GEDPartyType](#))

setProviderParType

```
public void setProviderParType(String providerParType)
```

set type to party reference of the provider. Mandatory The possible values for the type are: (one of [GEDPartyType](#))

getFolderType

```
public String getFolderType()
```

get Folder (or dossier) type. Optional : default to `OutgoingLetterDocType.getFolderType()`

setFolderType

```
public void setFolderType(String folderType)
```

set Folder (or dossier) type. Optional : default to `OutgoingLetterDocType.getFolderType()`

getMedAdvOrg

```
public String getMedAdvOrg()
```

get organisation from entity reference to the Medical Adviser Optional

setMedAdvOrg

```
public void setMedAdvOrg(String medAdvOrg)
```

set organisation to the entity reference to the Medical Adviser Optional

getMedAdvId

```
public String getMedAdvId()
```

get the Id from the entity reference to the Medical Adviser Optional

setMedAdvId

```
public void setMedAdvId(String medAdvId)
```

set the id to entity reference to the Medical Adviser Optional

getMedAdvType

```
public String getMedAdvType()
```

(continued from last page)

get the type from the entity reference to the Medical Adviser Optional The possible values for the type are: (one of [GEDEntityType](#))

setMedAdvType

```
public void setMedAdvType(String medAdvType)
```

set the type for entity reference to the Medical Adviser Optional The possible values for the type are: (one of [GEDEntityType](#))

getSurMedAdvOrg

```
public String getSurMedAdvOrg()
```

get the organisation from the entity reference to the Medical Adviser Office Optional

setSurMedAdvOrg

```
public void setSurMedAdvOrg(String surMedAdvOrg)
```

set the organisation to the entity reference to the Medical Adviser Office Optional

getSurMedAdvId

```
public String getSurMedAdvId()
```

get the Id from the entity reference to the Medical Adviser Office Optional

setSurMedAdvId

```
public void setSurMedAdvId(String surMedAdvId)
```

set the Id to the entity reference to the Medical Adviser Office Optional

getSurMedAdvType

```
public String getSurMedAdvType()
```

get type from the entity reference to the Medical Adviser Office Optional The possible values are: (one of [GEDEntityType](#))

setSurMedAdvType

```
public void setSurMedAdvType(String surMedAdvType)
```

set type to the entity reference to the Medical Adviser Office Optional The possible values are: (one of [GEDEntityType](#))

getMancenOrg

```
public String getMancenOrg()
```

get organisation from the entity reference to the Management Center Optional : default to the management center of the holder

setMancenOrg

```
public void setMancenOrg(String mancenOrg)
```

(continued from last page)

set the organisation to the entity reference to the Management Center Optional : default to the management center of the holder

getMancenId

```
public String getMancenId()
```

get the id from the entity reference to the Management Center Optional : default to the management center of the holder

setMancenId

```
public void setMancenId(String mancenId)
```

set the id to the entity reference to the Management Center Optional : default to the management center of the holder

getMancenType

```
public String getMancenType()
```

get type from the entity reference to the Management Center Optional : default to the management center of the holder The possible values are: (one of [GEDEntityType](#))

setMancenType

```
public void setMancenType(String mancenType)
```

set the type to entity reference to the Management Center Optional : default to the management center of the holder The possible values are: (one of [GEDEntityType](#))

getCanvasName

```
public String getCanvasName()
```

get Canvas name Optional : default to the canvas linked to the document type

setCanvasName

```
public void setCanvasName(String canvasName)
```

set Canvas name Optional : default to the canvas linked to the document type

getVersionNbr

```
public String getVersionNbr()
```

get the Canvas version number Optional : default to the last version (with production status)

setVersionNbr

```
public void setVersionNbr(String versionNbr)
```

set Canvas version number Optional : default to the last version (with production status)

getContactInfoTypeCode

```
public String getContactInfoTypeCode()
```

get the Contact info type. It is used to retrieve the address in Party Optional : default to the one specified in the document type The possibles values are : (one of [GEDContactInfoType](#))

setContactInfoTypeCode

```
public void setContactInfoTypeCode(String contactInfoTypeCode)
```

set the Contact info type. It is used to retrieve the address in Party Optional : default to the one specified in the document type The possibles values are : (one of [GEDContactInfoType](#))

getRecipient

```
public String getRecipient()
```

get the recipient(RecipientAddress)

setRecipient

```
public void setRecipient(String recipient)
```

set the recipient(RecipientAddress)

getCo

```
public String getCo()
```

get the co (RecipientAddress)

setCo

```
public void setCo(String co)
```

set the co (RecipientAddress)

getStreet

```
public String getStreet()
```

get the street(RecipientAddress)

setStreet

```
public void setStreet(String street)
```

set the street(RecipientAddress)

getNumber

```
public String getNumber()
```

get the number(RecipientAddress)

setNumber

```
public void setNumber(String number)
```

set the number(RecipientAddress)

(continued from last page)

getBox

```
public String getBox()
```

get the box(RecipientAddress)

setBox

```
public void setBox(String box)
```

set the box(RecipientAddress)

getPostalCode

```
public String getPostalCode()
```

get the postal code (RecipientAddress)

setPostalCode

```
public void setPostalCode(String postalCode)
```

set the postal code(RecipientAddress)

getCity

```
public String getCity()
```

get the city(RecipientAddress)

setCity

```
public void setCity(String city)
```

set the city(RecipientAddress)

getCountry

```
public String getCountry()
```

get the country(RecipientAddress)

setCountry

```
public void setCountry(String country)
```

get the country(RecipientAddress)

getPrinterId

```
public String getPrinterId()
```

get the identification of the printer in ORG Optional : default to the printer calculated with the "Starting Point Algorithm"

setPrinterId

```
public void setPrinterId(String printerId)
```

(continued from last page)

set the identification of the printer in ORG Optional : default to the printer calculated with the "Starting Point Algorithm"

isHoldSpool

```
public Boolean isHoldSpool()
```

if the spool is hold Optional : default to the value specified in the printer (in ORG)

setHoldSpool

```
public void setHoldSpool(Boolean holdSpool)
```

set the spool to hold Optional : default to the value specified in the printer (in ORG)

isSaveSpool

```
public Boolean isSaveSpool()
```

if the spool is saved Optional : default to the value specified in the printer (in ORG)

setSaveSpool

```
public void setSaveSpool(Boolean saveSpool)
```

set the spool to saved Optional : default to the value specified in the printer (in ORG)

getEmailFromName

```
public String getEmailFromName()
```

get the name email from

setEmailFromName

```
public void setEmailFromName(String emailFromName)
```

set the name email from

getEmailFromAddress

```
public String getEmailFromAddress()
```

get the address email from

setEmailFromAddress

```
public void setEmailFromAddress(String emailFromAddress)
```

set the address email from

getEmailToName

```
public String getEmailToName()
```

get the name email To

(continued from last page)

setEmailToName

```
public void setEmailToName(String emailToName)
```

set the name email To

getEmailToAdress

```
public String getEmailToAdress()
```

get the address email To

setEmailToAdress

```
public void setEmailToAdress(String emailToAdress)
```

set the address email To

getSubject

```
public String getSubject()
```

get the name email subject

setSubject

```
public void setSubject(String subject)
```

set the email subject

getBody

```
public String getBody()
```

get the email body

setBody

```
public void setBody(String body)
```

set the email body

getLogoFilePath

```
public String getLogoFilePath()
```

get the email logo file path

setLogoFilePath

```
public void setLogoFilePath(String logoFilePath)
```

set the email logo fil path

getLogo

```
public byte[] getLogo()
```

(continued from last page)

get the logo email

setLogo

```
public void setLogo(byte[] logo)
```

set the logo email

getExecutionCode

```
public String getExecutionCode()
```

get the execution code The possible values are: (one of [GEDExecutionCode](#))

setExecutionCode

```
public void setExecutionCode(String executionCode)
```

set the execution code The possible values are: (one of [GEDExecutionCode](#))

getShipper1Org

```
public String getShipper1Org()
```

get the organisation from entity reference to the shipping entity 1 Optional : default to the one calculated by the "Starting Point Algorithm"

setShipper1Org

```
public void setShipper1Org(String shipper1Org)
```

set the organisation to entity reference to the shipping entity 1 Optional : default to the one calculated by the "Starting Point Algorithm"

getShipper1Id

```
public String getShipper1Id()
```

get the id from the entity reference to the shipping entity 1 Optional : default to the one calculated by the "Starting Point Algorithm"

setShipper1Id

```
public void setShipper1Id(String shipper1Id)
```

set the id to entity reference to the shipping entity 1 Optional : default to the one calculated by the "Starting Point Algorithm"

getShipper1Type

```
public String getShipper1Type()
```

get the type from the entity reference to the shipping entity 1 Optional : default to the one calculated by the "Starting Point Algorithm" The possible values are: (one of [GEDEntityType](#))

setShipper1Type

```
public void setShipper1Type(String shipper1Type)
```

(continued from last page)

set the type to entity reference to the shipping entity 1 Optional : default to the one calculated by the "Starting Point Algorithm" The possible values are: (one of [GEDEntityType](#))

getShipper2Org

```
public String getShipper2Org()
```

get the organisation from entity reference to the shipping entity 2 Optional : default to the one calculated by the "Starting Point Algorithm"

setShipper2Org

```
public void setShipper2Org(String shipper2Org)
```

set the organisation to entity reference to the shipping entity 2 Optional : default to the one calculated by the "Starting Point Algorithm"

getShipper2Id

```
public String getShipper2Id()
```

get the id from the entity reference to the shipping entity 2 Optional : default to the one calculated by the "Starting Point Algorithm"

setShipper2Id

```
public void setShipper2Id(String shipper2Id)
```

set the id to entity reference to the shipping entity 2 Optional : default to the one calculated by the "Starting Point Algorithm"

getShipper2Type

```
public String getShipper2Type()
```

get the type from the entity reference to the shipping entity 2 Optional : default to the one calculated by the "Starting Point Algorithm" The possible values are: (one of [GEDEntityType](#))

setShipper2Type

```
public void setShipper2Type(String shipper2Type)
```

set the type to entity reference to the shipping entity 2 Optional : default to the one calculated by the "Starting Point Algorithm" The possible values are: (one of [GEDEntityType](#))

getShipperRCOrg

```
public String getShipperRCOrg()
```

get the organisation from entity reference to the shipping entity RC Optional : default to the one calculated by the "Starting Point Algorithm"

setShipperRCOrg

```
public void setShipperRCOrg(String shipperRCOrg)
```

set the organisation to entity reference to the shipping entity RC Optional : default to the one calculated by the "Starting Point Algorithm"

(continued from last page)

getShipperRCId

```
public String getShipperRCId()
```

get the id from the entity reference to the shipping entity RC Optional : default to the one calculated by the "Starting Point Algorithm"

setShipperRCId

```
public void setShipperRCId(String shipperRCId)
```

set the id to entity reference to the shipping entity RC Optional : default to the one calculated by the "Starting Point Algorithm"

getShipperRCType

```
public String getShipperRCType()
```

get the type from the entity reference to the shipping entity RC Optional : default to the one calculated by the "Starting Point Algorithm" The possible values are: (one of [GEDEntityType](#))

setShipperRCType

```
public void setShipperRCType(String shipperRCType)
```

set the type to entity reference to the shipping entity RC Optional : default to the one calculated by the "Starting Point Algorithm" The possible values are: (one of [GEDEntityType](#))

getSigner1Org

```
public String getSigner1Org()
```

get the organisation from entity reference to the Signer1 entity Optional : default to the one calculated by the "Starting Point Algorithm"

setSigner1Org

```
public void setSigner1Org(String signer1Org)
```

set the organisation to entity reference to the Signer1 entity Optional : default to the one calculated by the "Starting Point Algorithm"

getSigner1Id

```
public String getSigner1Id()
```

get the id from the entity reference to the Signer1 entity Optional : default to the one calculated by the "Starting Point Algorithm"

setSigner1Id

```
public void setSigner1Id(String signer1Id)
```

set the id to entity reference to the Signer1 entity Optional : default to the one calculated by the "Starting Point Algorithm"

getSigner1Type

```
public String getSigner1Type()
```

(continued from last page)

get the type from the entity reference to the Signer1 entity Optional : default to the one calculated by the "Starting Point Algorithm" The possible values are: (one of [GEDEntityType](#))

setSigner1Type

```
public void setSigner1Type(String signer1Type)
```

set the type to entity reference to the Signer1 entity Optional : default to the one calculated by the "Starting Point Algorithm" The possible values are: (one of [GEDEntityType](#))

getSigner2Org

```
public String getSigner2Org()
```

get the organisation from entity reference to the Signer2 entity Optional : default to the one calculated by the "Starting Point Algorithm"

setSigner2Org

```
public void setSigner2Org(String signer2Org)
```

set the organisation to entity reference to the Signer2 entity Optional : default to the one calculated by the "Starting Point Algorithm"

getSigner2Id

```
public String getSigner2Id()
```

get the id from the entity reference to the Signer2 entity Optional : default to the one calculated by the "Starting Point Algorithm"

setSigner2Id

```
public void setSigner2Id(String signer2Id)
```

set the id to entity reference to the Signer2 entity Optional : default to the one calculated by the "Starting Point Algorithm"

getSigner2Type

```
public String getSigner2Type()
```

get the type from the entity reference to the Signer2 entity Optional : default to the one calculated by the "Starting Point Algorithm" The possible values are: (one of [GEDEntityType](#))

setSigner2Type

```
public void setSigner2Type(String signer2Type)
```

set the type to entity reference to the Signer2 entity Optional : default to the one calculated by the "Starting Point Algorithm" The possible values are: (one of [GEDEntityType](#))

isEMail

```
public Boolean isEMail()
```

is the email yes or not

(continued from last page)

setEMail

```
public void setEMail(Boolean mail)
```

set the email to yes or not

getCanvasStatus

```
public String getCanvasStatus()
```

get the CanvasStatus The possible values for the type are: (one of [GEDCanvasStatus](#))

setCanvasStatus

```
public void setCanvasStatus(String canvasStatus)
```

set the CanvasStatus The possible values for the type are: (one of [GEDCanvasStatus](#))

getBusinessRef

```
public String getBusinessRef()
```

get the business reference

setBusinessRef

```
public void setBusinessRef(String businessRef)
```

set the business reference

getRequestCsoId

```
public String getRequestCsoId()
```

get the request CSO Id

Returns:

requestCsoId

setRequestCsoId

```
public void setRequestCsoId(String requestCsoId)
```

set the request CSO id

Parameters:

requestCsoId

setRecommend

```
public void setRecommend(Boolean recommend)
```

set Registered Mail. Optional : default to OutgoingLetterDocType.recommend

isRecommend

```
public Boolean isRecommend()
```

(continued from last page)

get Registered Mail. Optional : default to OutgoingLetterDocType.recommend

be.gfdi.com.common.services.ged Class GEDPrintParameter

java.lang.Object

└--be.gfdi.com.common.services.ged.GEDPrintParameter

All Implemented Interfaces:

Serializable

public class **GEDPrintParameter**
extends Object
implements Serializable

Constructor Summary

public	GEDPrintParameter()
--------	-------------------------------------

Method Summary

String	getParameter()
String	getValue()
void	setParameter(String parameter)
void	setValue(String value)

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

GEDPrintParameter

public **GEDPrintParameter()**

Methods

setParameter

public void **setParameter**(String parameter)

(continued from last page)

getParameter

```
public String getParameter()
```

setValue

```
public void setValue(String value)
```

getValue

```
public String getValue()
```

be.gfdi.com.common.services.ged Interface GedPrintService

public interface **GedPrintService**
extends

Method Summary

String	printDocument (GEDPrintDocumentRequest request) Initiates the asynchronous printing of a new document.
GEDEvent	printSynchronous (GEDPrintDocumentRequest request) Prints a new document synchronously.

Methods

printDocument

```
public String printDocument(GEDPrintDocumentRequest request)  
    throws COMGEDEException
```

Initiates the asynchronous printing of a new document. The document is created in the catalog with status 'P' and put in the jade2 printing message queue.

Parameters:

request - The document that has to be printed

Returns:

The generated ticket number

printSynchronous

```
public GEDEvent printSynchronous(GEDPrintDocumentRequest request)  
    throws COMGEDEException
```

Prints a new document synchronously. If everything goes well, the document is created in the catalog with the status 'C'. When the document is flagged as eMail and eMail will be sent and an additional document will be added to the catalog, which is linked to the first one. This means it is possible that one call will result in multiple documents in the catalog. The first one, corresponding to the actual print will be returned. If there is an error, a COMGEDEException is thrown and no document is created in the catalog.

Parameters:

request - The document that has to be printed

be.gfdi.com.common.services.ged Interface OrchestratorWebService

public interface **OrchestratorWebService**
extends

Method Summary

void	assignTask (AssignTaskRequest request) Assign a task to a user
CompleteTaskResponse	completeTask (CompleteTaskRequest request) Completes a running task
DispatchEventResponse	dispatchEvent (DispatchEventRequest request) Start a new process instance based on the info from the DispatchEventRequest
GEDEventRouting	findEventRouting (String docType) Returns the EventRouting information according to the docType passed as argument.
boolean	updateEventRouting (GEDEventRouting gedEventRouting) Create the event routing for the specified doc type OR update it with the passed data if it already exists.

Methods

dispatchEvent

public [DispatchEventResponse](#) **dispatchEvent**([DispatchEventRequest](#) request)
throws [COMGEException](#)

Start a new process instance based on the info from the DispatchEventRequest

Parameters:
request

Returns:
DispatchEventResponse

completeTask

public [CompleteTaskResponse](#) **completeTask**([CompleteTaskRequest](#) request)
throws [COMGEException](#)

Completes a running task

Parameters:
request

Returns:
CompleteTaskResponse

findEventRouting

```
public GEDEventRouting findEventRouting(String docType)
    throws COMGEDEException
```

Returns the EventRouting information according to the docType passed as argument.

Parameters:

docType - The docType to search for.

Returns:

The GEDEventRouting object containing info relative to the docType passed as argument.

updateEventRouting

```
public boolean updateEventRouting(GEDEventRouting gedEventRouting)
    throws COMGEDEException
```

Create the event routing for the specified doc type OR update it with the passed data if it already exists.

Parameters:

gedEventRouting - The GEDEventRouting object to create OR update.

Returns:

True if the object already existed AND False if the object did not yet exist and has just been created.

assignTask

```
public void assignTask(AssignTaskRequest request)
    throws COMGEDEException
```

Assign a task to a user

Parameters:

request - the request containing the taskId and the userId

Throws:

[COMGEDEException](#)

be.gfdi.com.common.services.ged

Class PartyType

java.lang.Object

└--be.gfdi.com.common.services.ged.PartyType

public abstract class **PartyType**
extends Object

constants for each type of party

Field Summary

public static final	EMPLOYER THIRD EMPLOYER Value: EMPLOYER
public static final	INSTITUTION THIRD INSTITUTION Value: INSTITUTION
public static final	MUTUALITYPERSON THIRD MUTUALITYPERSON Value: MUTUALITYPERSON
public static final	PROVIDER THIRD PROVIDER Value: PROVIDER

Constructor Summary

public	PartyType()
--------	-----------------------------

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Fields

EMPLOYER

public static final java.lang.String **EMPLOYER**

THIRD EMPLOYER
Constant value: **EMPLOYER**

PROVIDER

public static final java.lang.String **PROVIDER**

THIRD PROVIDER

(continued from last page)

Constant value: **PROVIDER**

MUTUALITYPERSON

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

THIRD MUTUALITYPERSON

Constant value: **MUTUALITYPERSON**

INSTITUTION

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

THIRD INSTITUTION

Constant value: **INSTITUTION**

Constructors

PartyType

```
public PartyType()
```

be.gfdi.com.common.services.ged

Class ReindexEventRequest

java.lang.Object

└─be.gfdi.com.common.services.ged.ReindexEventRequest

All Implemented Interfaces:

Serializable

public class **ReindexEventRequest**

extends Object

implements Serializable

Request to the [CatalogService.reindexEvent\(ReindexEventRequest\)](#) service

Constructor Summary

public	ReindexEventRequest()
--------	---------------------------------------

Method Summary

String	getEventKey() get the the key of the original event in the format "561-2006-561-0123456789" or "56120069990123456789" This parameter is mandatory
String	getNewDocType() get the new doctype (i.e.
String	getNewThirdCodeType() get the code type used to identify the new third (one of ThirdCodeType) This parameter is mandatory
String	getNewThirdCodeValue() get the actual code that identifies the new third This parameter is mandatory
String	getNewThirdOrg() get the organization of the new third This parameter is mandatory
boolean	isWorkflowFlag() return if the workflow must be launched this parameter is optional
void	setEventKey(String eventKey) set the key of the original event in the format "561-2006-561-0123456789" or "56120069990123456789" This parameter is mandatory
void	setNewDocType(String newDocType) set the new doctype (i.e.
void	setNewThirdCodeType(String newThirdCodeType) set the code type used to identify the new third (one of ThirdCodeType) This parameter is mandatory
void	setNewThirdCodeValue(String newThirdCodeValue) set the actual code that identifies the new third This parameter is mandatory

void	setNewThirdOrg (String newThirdOrg) set the organization of the new third This parameter is mandatory
void	setWorkflowFlag (boolean workflowFlag) set the workflow flag for this document this parameter is optional

Methods inherited from class `java.lang.Object`

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Constructors

ReindexEventRequest

```
public ReindexEventRequest()
```

Methods

getEventKey

```
public String getEventKey()
```

get the the key of the original event in the format "561-2006-561-0123456789" or "56120069990123456789" This parameter is mandatory

setEventKey

```
public void setEventKey(String eventKey)
```

set the key of the original event in the format "561-2006-561-0123456789" or "56120069990123456789" This parameter is mandatory

getNewDocType

```
public String getNewDocType()
```

get the new doctype (i.e. K00011) This parameter is mandatory

setNewDocType

```
public void setNewDocType(String newDocType)
```

set the new doctype (i.e. K00011) This parameter is mandatory

getNewThirdCodeType

```
public String getNewThirdCodeType()
```

get the code type used to identify the new third (one of [ThirdCodeType](#)) This parameter is mandatory

setNewThirdCodeType

```
public void setNewThirdCodeType(String newThirdCodeType)
```

(continued from last page)

set the code type used to identify the new third (one of [ThirdCodeType](#)) This parameter is mandatory

getNewThirdCodeValue

```
public String getNewThirdCodeValue()
```

get the actual code that identifies the new third This parameter is mandatory

setNewThirdCodeValue

```
public void setNewThirdCodeValue(String newThirdCodeValue)
```

set the actual code that identifies the new third This parameter is mandatory

getNewThirdOrg

```
public String getNewThirdOrg()
```

get the organization of the new third This parameter is mandatory

setNewThirdOrg

```
public void setNewThirdOrg(String newThirdOrg)
```

set the organization of the new third This parameter is mandatory

isWorkflowFlag

```
public boolean isWorkflowFlag()
```

return if the workflow must be launched this parameter is optional

setWorkflowFlag

```
public void setWorkflowFlag(boolean workflowFlag)
```

set the workflow flag for this document this parameter is optional

be.gfdi.com.common.services.ged

Class ThirdCodeType

java.lang.Object

└-be.gfdi.com.common.services.ged.ThirdCodeType

public abstract class **ThirdCodeType**
extends Object

All constants use to reference a third

Field Summary

public static final	ENTERPRISE_NUMBER ENTERPRISE_NUMBER Value: ENTID
public static final	EXTERNAL_NUMBER External Number Value: EXID
public static final	ID Party ID Value: ID
public static final	INTERNAL_NUMBER INTERNAL_NUMBER Value: INID
public static final	NISS_NUMBER Niss number Value: NISS
public static final	RIZIV_NUMBER RIZIV_NUMBER Value: RIZIV
public static final	RSZ_NUMBER RSZ_NUMBER Value: RSZ
public static final	UNIQUE_NUMBER Unique Number Value: UID

Constructor Summary

public	ThirdCodeType()
--------	---------------------------------

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Fields

ID

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

Party ID
Constant value: **ID**

UNIQUE_NUMBER

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

Unique Number
Constant value: **UID**

EXTERNAL_NUMBER

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

External Number
Constant value: **EXID**

NISS_NUMBER

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

Niss number
Constant value: **NISS**

RSZ_NUMBER

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

RSZ_NUMBER
Constant value: **RSZ**

RIZIV_NUMBER

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

RIZIV_NUMBER
Constant value: **RIZIV**

ENTERPRISE_NUMBER

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

ENTERPRISE_NUMBER
Constant value: **ENTID**

INTERNAL_NUMBER

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

INTERNAL_NUMBER
Constant value: **INID**

Constructors

ThirdCodeType

```
public ThirdCodeType()
```

be.gfdi.com.common.services.ged

Class UpdateFolderRefRequest

java.lang.Object

↳ be.gfdi.com.common.services.ged.UpdateFolderRefRequest

All Implemented Interfaces:

Serializable

public class **UpdateFolderRefRequest**

extends Object

implements Serializable

Constructor Summary

public	UpdateFolderRefRequest ()
--------	--

Method Summary

String	getKey () get the event key This parameter is mandatory
String	getNewFolderRef () get the new folder type (for instance K0) This parameter is mandatory
Date	getNewFolderRefEndDate () get the new folder reference end date This parameter is optional
Date	getNewFolderRefStartDate () get the new folder reference start date This parameter is optional
String	getNewFolderType () get the new folder reference (for instance 561123456789) This parameter is mandatory
void	setKey (String key) set the event key This parameter is mandatory
void	setNewFolderRef (String newFolderRef) set the new folder type (for instance K0) This parameter is mandatory
void	setNewFolderRefEndDate (Date newFolderRefEndDate) set the new folder reference end date This parameter is optional
void	setNewFolderRefStartDate (Date newFolderRefStartDate) set the new folder reference start date This parameter is optional
void	setNewFolderType (String newFolderType) set the new folder reference (for instance 561123456789) This parameter is mandatory

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

UpdateFolderRefRequest

```
public UpdateFolderRefRequest()
```

Methods

getKey

```
public String getKey()
```

get the event key This parameter is mandatory

setKey

```
public void setKey(String key)
```

set the event key This parameter is mandatory

getNewFolderRef

```
public String getNewFolderRef()
```

get the new folder type (for instance K0) This parameter is mandatory

setNewFolderRef

```
public void setNewFolderRef(String newFolderRef)
```

set the new folder type (for instance K0) This parameter is mandatory

getNewFolderType

```
public String getNewFolderType()
```

get the new folder reference (for instance 561123456789) This parameter is mandatory

setNewFolderType

```
public void setNewFolderType(String newFolderType)
```

set the new folder reference (for instance 561123456789) This parameter is mandatory

getNewFolderRefStartDate

```
public Date getNewFolderRefStartDate()
```

get the new folder reference start date This parameter is optional

setNewFolderRefStartDate

```
public void setNewFolderRefStartDate(Date newFolderRefStartDate)
```

(continued from last page)

set the new folder reference start date This parameter is optional

getNewFolderRefEndDate

```
public Date getNewFolderRefEndDate()
```

get the new folder reference end date This parameter is optional

setNewFolderRefEndDate

```
public void setNewFolderRefEndDate(Date newFolderRefEndDate)
```

set the new folder reference end date This parameter is optional

be.gfdi.com.common.services.ged

Class WorkflowStatus

java.lang.Object

└─be.gfdi.com.common.services.ged.WorkflowStatus

public abstract class **WorkflowStatus**
extends Object

Field Summary

public static final	CANCELLED Workflow cancelled Value: 05
public static final	COMPLETED Workflow completed Value: 06
public static final	NO_WORKFLOW No workflow associated Value: 01
public static final	NOT_STARTED Workflow not started Value: 03
public static final	RUNNING Workflow is running Value: 02
public static final	SUSPENDED Workflow suspended Value: 04

Constructor Summary

public	WorkflowStatus()
--------	----------------------------------

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Fields

NO_WORKFLOW

public static final java.lang.String **NO_WORKFLOW**

(continued from last page)

No workflow associated
Constant value: **01**

RUNNING

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

Workflow is running
Constant value: **02**

NOT_STARTED

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

Workflow not started
Constant value: **03**

SUSPENDED

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

Workflow suspended
Constant value: **04**

CANCELLED

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

Workflow cancelled
Constant value: **05**

COMPLETED

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

Workflow completed
Constant value: **06**

Constructors

WorkflowStatus

```
public WorkflowStatus()
```

Index

A

AGENCY 107
appendToNoteLog 25
AppendToNoteLogRequest 15
assignTask 175
AssignTaskRequest 17

B

BATCH 132
BATCHNOPRINT 133

C

CANCELLED 188
CATALOG_KEY_WARNING_SEPARATOR 40
COMGEException 38
COMGEDWarning 41
COMPLETED 188
completeTask 174
CompleteTaskMessageConverter 42
CompleteTaskRequest 44
CompleteTaskResponse 46
CompleteTaskVariable 49
CONTACTPERSON 98
CONTACTPERSONPARTY 98
convertIn 131
convertOut 130
convertRequest 42, 66
convertResponse 42, 66
COWORKER 107, 139
createEvent 23
CreateEventReply 52
CreateEventRequest 58
createEventWithWarning 23

D

DEDICATEDAGENCY 107, 140
DEPARTMENT 107, 140
dispatchEvent 174
DispatchEventMessageConverter 66

DispatchEventRequest 70
DispatchEventResponse 78
DispatchEventVariable 81
DOC_MEDIA_ELECTRONIC_FLOW 85
DOC_MEDIA_EMAIL 85
DOC_MEDIA_EXCEL 85
DOC_MEDIA_FAX 85
DOC_MEDIA_INFOBOX 85
DOC_MEDIA_PAPER 84
DOC_SOURCE_BULK_IMPORT 87
DOC_SOURCE_DGC 87
DOC_SOURCE_ELECTRONIC_FLOW 87
DOC_SOURCE_IM 87
DOC_SOURCE_IM_LIGHT 87
DOC_SOURCE_JADE 87
DOC_SOURCE_OFD 87
DOC_SOURCE_WEB_SERVICE 87
DocMedia 85
DocSource 88

E

EMPLOYER 138, 176
ENTERPRISE_NUMBER 182
ERROR_ATTACHED_FILE_IS_EMPTY 37
ERROR_ATTACHED_FILE_IS_TOO_LARGE 38
ERROR_BATCH_SCAN_YEAR_NOT_VALID 36
ERROR_CANVAS_STATUS_NOT_VALID 36
ERROR_CONTACT_INFO_TYPE_NOT_VALID 36
ERROR_DIRECTION_DOES_NOT_EXIST 32
ERROR_DOCLANGUAGE_DOES_NOT_EXIST 33
ERROR_DOCMEDIA_DOES_NOT_EXIST 37
ERROR_DOCSOURCE_DOES_NOT_EXIST 37
ERROR_DOCTYPE_DOES_NOT_EXIST_KEY 32
ERROR_DOCTYPE_NOT_CSO_KEY 36
ERROR_DOCTYPE_NOT_MME_KEY 32
ERROR_DOCTYPESUBCLASS_DOES_NOT_EXIST 32
ERROR_DOCUMENT_DOES_NOT_EXIST_IN_DOCUMENTSDATA 35
ERROR_DOCUMENT_FORMAT_NOT_ALLOWED_FOR_LEGAL 37
ERROR_DOCUMENT_FORMAT_NOT_ALLOWED_FOR_NON_LEGAL 37
ERROR_DOCUMENT_PDF_MAX_SIZE_PER_PAGE_EXCEEDED 37

- ERROR_ENTITY_IS_NOT_CORRECT 35
ERROR_ENTITY_IS_NOT_MANAGEMENT_CENTER 34
ERROR_ENTITY_TYPE_NOT_VALID 36
ERROR_EVENT_DOES_NOT_EXIST_KEY 32
ERROR_EVENT_IMAGE_EXIST 35
ERROR_EVENT_NOT_MME_KEY 32
ERROR_EXECUTION_CODE_NOT_VALID 35
ERROR_FILE_EXTENSION_CODE_CM_DOES_NOT_EXIST 34
ERROR_FOLDER_REFERENCE_INVALID_KEY 33
ERROR_FOLDERTYPE_DOES_NOT_EXIST 33
ERROR_GED_DEFAULT_USER_CODE_PARAMETER_NOT_EXIST 35
ERROR_IMAGE_FILE_WITHOUT_EXTENSION_TYPE_NOT_POSSIBLE 34
ERROR_IMAGE_KEY_INVALID 33
ERROR_IMAGE_PATH_SERVER_NOT_CORRECT 34
ERROR_IMPORTING_IMAGE_IN_CONTENT_MANAGER 35
ERROR_INVALID_KEY 31
ERROR_LANGUAGE_NOT_VALID 36
ERROR_MANAGEMENT_CENTER_DOES_NOT_EXIST 34
ERROR_NO_PROCESS_DEFINED_FOR_THIS_DOCTYPE 37
ERROR_NOTLOG_NOT_EXIST_FOR_EVENT_KEY 34
ERROR_ORG_FOR_THIS_ENTITY_ID_IS_NOT_CORRECT 36
ERROR_REQUEST_WITH_IMAGE_AND_IMAGEKEY_NOT_POSSIBLE 34
ERROR_SCANBATCH_NUMBER_INVALID_KEY 33
ERROR_SCANNER_ID_INVALID_KEY 33
ERROR_SECONDARY_THIRD_TYPE_INVALID 33
ERROR_THIRD_CODE_VALUE_FOR_CODETYPE_INVALID_KEY 33
ERROR_THIRD_CODETYPE_INVALID_KEY 32
ERROR_THIRD_DOES_NOT_EXIST_KEY 32
ERROR_THIRD_ORG_INVALID_KEY 33
ERROR_THIRD_ORGANIZATION_CAN_NOT_BE_NULL 37
ERROR_THIRD_PARTY_TYPE 35
ERROR_TYPE_FOR_THIS_ENTITY_ID_IS_NOT_CORRECT 37
ERROR_UNKNOWN_KEY 38
ERROR_WORKFLOW_NAME_CAN_NOT_BE_NULL 34
EXTERNAL_NUMBER 182

F

FACTURATION 98
findByParameters 21
FindByParametersRequest 90
findDocTypes 25
FindDocTypesRequest 94
findEventRouting 175

G

GEDCanvasStatus 96
GEDContactInfoType 98
GEDDocType 101
GEDEntityType 109
GEDEvent 114
GEDEventDetail 124
GEDEventFolderInfo 125
GEDEventRouting 127
GEDEventRoutingConverter 130
GEDExecutionCode 133
GEDFolderType 134
GEDPartyType 141
GEDPrintDocumentRequest 151
GEDPrintParameter 171
GENERALAGENCY 139
getAllDocMedia 85
getAllDocSource 88
getAllDocTypes 25
getAllFolderTypes 25
getArchiveBoxType 103
getArchiveMode 103
getBenCodeType 155
getBenCodeValue 156
getBenOrg 155
getBenParType 156
getBody 164
getBodyParameters 155
getBox 161
getBusinessRef 169
getCanvasName 160
getCanvasStatus 169
getCatalogDate 75, 114

getCatalogEndDate 90
getCatalogImageId 75
getCatalogKey 63, 75
getCatalogStartDate 91
getCity 162
getCmIndex 53
getCo 161
getCode 103, 135
getCodeStartWith 94
getComment 114
getConfidentialLevel 104
getContactInfoTypeCode 160
getContent 15
getContentAvailable 114
getContentCreatable 115
getCopyNumber 155
getCountry 162
getDateValue 82
getDcId 104, 120
getDecimalValue 82
getDgcFilter 102
getDgcLabelFr 102
getDgcLabelGe 103
getDgcLabelNI 103
getDgcSection 102
getDirection 92, 101, 114
getDocLanguage 58
getDocMedia 64
getDocSource 64
getDocType 58, 91, 116, 128
getDocTypeCode 75
getDocTypeLabel 114
getDocTypeLabelFr 76
getDocTypeLabelNI 76
getDocTypeName 151
getDocTypeSubClass 76
getDocumentLanguage 75
getEmailFromAddress 163
getEmailFromName 163
getEmailToAdress 164
getEmailToName 163
getEndDate 126
getEngine 128
getErrorCode 38
getErrorMessage 38
getErrorStackTrace 39
getEventByKey 22
getEventByScanBatchNumber 22
getEventExternalId 76
getEventKey 179
getExecutionCode 165
getFolderInfos 122
getFolderNum 126
getFolderNumber 114
getFolderRef 58, 93, 153
getFolderType 59, 91, 115, 126, 158
getFreeField 63, 120
getId 128
getImage 59
getImageBytes 65
getImageCatalogKey 63
getImageKey 59
getImageMimeType 59
getImageName 65
getKey 16, 52, 115, 185
getLabelFr 104, 135
getLabelGe 104, 136
getLabelNI 104, 135
getLanguage 115, 155
getLogo 164
getLogoFilePath 164
getLongValue 82
getMancenId 160
getMancenOrg 159
getMancenType 160
getMedAdvId 158
getMedAdvOrg 158
getMedAdvType 158
getMedium 115
getName 49, 81
getNewDocType 179
getNewFolderRef 185
getNewFolderRefEndDate 186
getNewFolderRefStartDate 185
getNewFolderType 185
getNewThirdCodeType 179
getNewThirdCodeValue 180
getNewThirdOrg 180
getNoteLog 24, 60, 119
getNumber 161

getOrganization 74, 136
getOwnerCodeType 153
getOwnerCodeValue 153
getOwnerOrg 152
getOwnerParType 153
getPageNumber 92
getPageSize 92
getParameter 171
getPayerCodeType 156
getPayerCodeValue 157
getPayerOrg 156
getPayerParType 157
getPostalCode 162
getPrintDate 116
getPrinterId 162
getProcessDefinitionName 77
getProcessDuration 128
getProcessInstanceId 46, 78
getProcessName 128
getProviderCodeType 157
getProviderCodeValue 157
getProviderOrg 157
getProviderParType 158
getReceiveDate 60
getRecipient 161
getRecipientCodeType 152
getRecipientCodeValue 152
getRecipientOrg 152
getRecipientParType 152
getRequestCsoId 169
getScanBatchNumber 60, 115
getScanBatchNumberYear 60, 118
getScanDate 63
getScannerId 60, 119
getSecThirdCodeType 61
getSecThirdCodeValue 61
getSecThirdExternalNumber 120
getSecThirdFName 119
getSecThirdId 118
getSecThirdName 119
getSecThirdNiss 121
getSecThirdOrg 61, 122
getSecThirdParType 62
getSecThirdPartyType 121
getSecThirdRiziv 121
getShipper1Id 165
getShipper1Org 165
getShipper1Type 165
getShipper2Id 166
getShipper2Org 166
getShipper2Type 166
getShipperRCId 166
getShipperRCOrg 166
getShipperRCType 167
getSigner1Id 167
getSigner1Org 167
getSigner1Type 167
getSigner2Id 168
getSigner2Org 168
getSigner2Type 168
getStartDate 126
getStatus 115
getStreet 161
getStringValue 82
getSubClass 91, 115
getSubject 164
getSurMedAdvId 159
getSurMedAdvOrg 159
getSurMedAdvType 159
getTaskId 17, 44, 47, 79
getTaskName 47, 79
getTaskStatus 47, 79
getTaskTokenId 47, 79
getTaskVariables 45
getThirdAgencyId 72
getThirdAgencyNameDe 73
getThirdAgencyNameFr 73
getThirdAgencyNameNl 73
getThirdAgencyUniqueName 72
getThirdCodeType 61, 92
getThirdCodeValue 62, 92
getThirdExternalNbr 71
getThirdExternalNumber 120
getThirdFirstName 72
getThirdFName 116
getThirdId 116
getThirdLastName 72
getThirdManagementCenter 62, 119
getThirdManagementCenterId 73
getThirdManagementCenterLabelDe 74

getThirdManagementCenterLabelFr 74
getThirdManagementCenterLabelNI 74
getThirdManagementCenterUniqueName 73
getThirdName 116
getThirdNiss 120
getThirdNissNumber 71
getThirdOrg 62, 93, 122
getThirdOrganization 71
getThirdParType 63
getThirdPartyType 121
getThirdRiziv 121
getThirdUniqueNumber 71
getUser 74
getUserId 18, 154
getValue 172
getValueDate 50
getValueDouble 50
getValueLong 50
getValueString 50
getVariables 77
getVersionNbr 160
getWarning 53
getWorkflowStatus 64, 122
getWorkingEntityId 154
getWorkingEntityOrg 154
getWorkingEntityType 154
getWorkStation 154

H

HEADQUARTER 98

I

ID 182
IMAGE_DOES_NOT_EXIST_IN_CONTENT_MANGER 35
IMMEDIATE 132
IMMEDIATENOPRINT 133
INSTITUTION 139, 177
INTERNAL_NUMBER 182
isDgcCataloguable 102
isDocumentValidToBeImported 26
isEMail 168
isHoldSpool 163
isLegalDocument 64

isRecommend 169
isSaveSpool 163
isThirdStaffMember 72
isWorkflowFlag 180
isWorkflowFlg 62

L

loadEventDetail 21

M

MAIL 98
MANAGEMENTCENTER 108, 139
MEDICALADVISER 108, 139
MEDICALOFFICE 140
MUTUALITYPERSON 139, 177

N

NISS_NUMBER 182
NO_WORKFLOW 187
NOT_STARTED 188

O

OFFICE 98

P

PartyType 177
PRIMARYAGENCY 108
PRINCIPALAGENCY 139
printDocument 173
PRINTQUEUE 108, 140
printSynchronous 173
PRODUCTION 96
PROFESIONNAL 98
PROVIDER 138, 176

R

reindexEvent 20
ReindexEventRequest 179
reindexWithoutNewLine 20

RESIDENCE 97
RIZIV_NUMBER 182
RSZ_NUMBER 182
RUNNING 188

S

SECONDARYAGENCY 108, 139
SERVICE 108, 140
setArchiveBoxType 103
setArchiveMode 103
setBenCodeType 156
setBenCodeValue 156
setBenOrg 155
setBenParType 156
setBody 164
setBodyParameters 155
setBox 162
setBusinessRef 169
setCanvasName 160
setCanvasStatus 169
setCatalogDate 75, 116
setCatalogEndDate 91
setCatalogImageId 75
setCatalogKey 63, 75
setCatalogStartDate 91
setCity 162
setCmIndex 53
setCo 161
setCode 103, 135
setCodeStartWith 95
setComment 116
setConfidentialLevel 105
setContactInfoTypeCode 161
setContent 16
setContentAvailable 116
setContentCreatable 118
setCopyNumber 155
setCountry 162
setDateValue 82
setDcId 104, 120
setDecimalValue 82
setDgcCataloguable 102
setDgcFilter 102
setDgcLabelFr 102

setDgcLabelGe 103
setDgcLabelNI 103
setDgcSection 102
setDirection 92, 102, 116
setDocLanguage 58
setDocMedia 64
setDocSource 65
setDocType 58, 91, 118, 128
setDocTypeCode 76
setDocTypeLabel 117
setDocTypeLabelFr 76
setDocTypeLabelNI 76
setDocTypeName 151
setDocTypeSubClass 76
setDocumentLanguage 75
setEMail 168
setEmailFromAddress 163
setEmailFromName 163
setEmailToAdress 164
setEmailToName 163
setEndDate 126
setEngine 128
setErrorCode 39
setErrorMessage 39
setErrorStackTrace 39
setEventExternalId 76
setEventKey 179
setExecutionCode 165
setFolderInfos 122
setFolderNum 126
setFolderNumber 117
setFolderRef 59, 93, 153
setFolderType 59, 91, 117, 126, 158
setFreeField 63, 120
setHoldSpool 163
setId 128
setImage 59
setImageBytes 65
setImageCatalogKey 63
setImageKey 59
setImageMimeType 59
setImageName 65
setKey 16, 53, 117, 185
setLabelFr 104, 135
setLabelGe 104, 136

setLabelNI 104, 135
setLanguage 117, 155
setLegalDocument 64
setLogo 165
setLogoFilePath 164
setLongValue 82
setMancenId 160
setMancenOrg 159
setMancenType 160
setMedAdvId 158
setMedAdvOrg 158
setMedAdvType 159
setMedium 117
setName 50, 82
setNewDocType 179
setNewFolderRef 185
setNewFolderRefEndDate 186
setNewFolderRefStartDate 185
setNewFolderType 185
setNewThirdCodeType 179
setNewThirdCodeValue 180
setNewThirdOrg 180
setNoteLog 60, 119
setNumber 161
setOrganization 74, 136
setOwnerCodeType 153
setOwnerCodeValue 153
setOwnerOrg 153
setOwnerParType 153
setPageNumber 92
setPageSize 92
setParameter 171
setPayerCodeType 156
setPayerCodeValue 157
setPayerOrg 156
setPayerParType 157
setPostalCode 162
setPrintDate 117
setPrinterId 162
setProcessDefinitionName 77
setProcessDuration 129
setProcessInstanceId 47, 79
setProcessName 128
setProviderCodeType 157
setProviderCodeValue 157
setProviderOrg 157
setProviderParType 158
setReceiveDate 60
setRecipient 161
setRecipientCodeType 152
setRecipientCodeValue 152
setRecipientOrg 152
setRecipientParType 152
setRecommend 169
setRequestCsoId 169
setSaveSpool 163
setScanBatchNumber 60, 118
setScanBatchNumberYear 60, 118
setScanDate 64
setScannerId 61, 119
setSecThirdCodeType 61
setSecThirdCodeValue 61
setSecThirdExternalNumber 121
setSecThirdFName 119
setSecThirdId 118
setSecThirdName 119
setSecThirdNiss 121
setSecThirdOrg 61, 123
setSecThirdParType 62
setSecThirdPartyType 122
setSecThirdRiziv 121
setShipper1Id 165
setShipper1Org 165
setShipper1Type 165
setShipper2Id 166
setShipper2Org 166
setShipper2Type 166
setShipperRCId 167
setShipperRCOrg 166
setShipperRCType 167
setSigner1Id 167
setSigner1Org 167
setSigner1Type 168
setSigner2Id 168
setSigner2Org 168
setSigner2Type 168
setStartDate 126
setStatus 117
setStreet 161
setStringValue 82

setSubClass 91, 117
setSubject 164
setSurMedAdvId 159
setSurMedAdvOrg 159
setSurMedAdvType 159
setTaskId 17, 44, 47, 79
setTaskName 47, 79
setTaskStatus 47, 79
setTaskTokenId 47, 79
setTaskVariables 45
setThirdAgencyId 72
setThirdAgencyNameDe 73
setThirdAgencyNameFr 73
setThirdAgencyNameNl 73
setThirdAgencyUniqueName 72
setThirdCodeType 61, 92
setThirdCodeValue 62, 93
setThirdExternalNbr 71
setThirdExternalNumber 120
setThirdFirstName 72
setThirdFName 118
setThirdId 118
setThirdLastName 72
setThirdManagementCenter 62, 119
setThirdManagementCenterId 73
setThirdManagementCenterLabelDe 74
setThirdManagementCenterLabelFr 74
setThirdManagementCenterLabelNl 74
setThirdManagementCenterUniqueName 73
setThirdName 118
setThirdNiss 120
setThirdNissNumber 71
setThirdOrg 62, 93, 122
setThirdOrganization 71
setThirdParType 63
setThirdPartyType 121
setThirdRiziv 121
setThirdStaffMember 72
setThirdUniqueNumber 71
setUser 75
setUserId 18, 154
setValue 172
setValueDate 50
setValueDouble 50
setValueLong 50

setValueString 50
setVariables 77
setVersionNbr 160
setWarning 53
setWorkflowFlag 180
setWorkflowFlg 63
setWorkflowStatus 64, 122
setWorkingEntity 154
setWorkingEntityOrg 154
setWorkingEntityType 154
setWorkStation 154
SOCIALSEAT 108, 139
string 36
SURGERY 108
SUSPENDED 188

T

TEAM 108, 140
TEST 96
ThirdCodeType 182
toString 58

U

UNIQUE_NUMBER 182
updateEventRouting 175
updateEventWithWarning 24
updateFolderReference 22
UpdateFolderRefRequest 185
updateStatusForDelete 25
updateStatusForDouble 25

V

VIRTUALAGENCY 109, 140

W

WARN_NO_PROCESS_DEFINED_FOR_THIS_DOCTYPE 40
WARN_THIRD_IS_NOT_MUTUALITY_PERSON 40
WorkflowStatus 188
WORKSTATION 109, 140