Gestion des dossiers GED

GedDossierConsult

GedDossierConsult permet d'initialiser la demande de consultation d'un dossier GED en tenant compte des habilitations de l'utilisateur sur l'armoire et le dossier.

Les arguments « Sd » et « Ssd » permettent de définir le classement pour un positionnement automatique à l'ouverture en consultation.

```
char* GedDossierConsult(
  char *Armoire,
  char *User,
  char *Langue,
  char *Validite,
  char *Sd,
  char *Ssd,
  long IdDT,
  string RefDoc
  );
```

Paramètres

Armoire : Nom physique de l'armoire

User : Login de l'utilisateur

Langue : Langage de l'interface de consultation (fr ou en)

Validite : Durée de validité (en jours) du lien généré (0 : usage unique par défaut)

Sd : Nom du sous-dossier de classement (facultatif)

Ssd : Nom du sous-sous-dossier de classement (facultatif)

IdDT : Identifiant Multigest d'un dossier de travail pour positionnement (facultatif)

RefDoc : Référence du document

Codes retour

Cette fonction retourne un Numéro de session ou un code erreur

Le numéro de session retourné par cette fonction permet de construire l'url de consultation du dossier.
A la consultation de ce dossier, le numéro de session généré est détruit pour des raisons de sécurité.

URL : http://[Serveur]:[Port]/consultation/consultation_redirect.php?session=

Exemple (C++)

```
// Demande de consultation du DOSSIERGED=12005 de l'armoire MDPH
// Les droits de l'utilisateurs ADMIN seront pris en compte
GedInitConnection([SERVEURGED],[PORT],0);
GedAddChampRecherche(« DOSSIERGED », »12005″);
strcpy(NoSession,GedDossierConsult(« MDPH », »ADMIN », »fr », »0″));
```

GedDossierExist

GedDossierExist permet de vérifier l'existence d'un dossier dans la GED.

La recherche d'un dossier se fait sur des critères permettant de construire une clause « WHERE ». Ces critères peuvent être multiples (ex : numéro client + nom du client)

```
long GedDossierExist(
  char *Armoire,
  char *User
  );
```

Paramètres

Armoire : Nom physique de l'armoire

User : Login de l'utilisateur

Codes retour

Si le dossier existe, la fonction renvoie le numéro du dossier GED (numéro Multigest unique) que nous appellerons « ID Multigest ». Cet « ID Multigest » sera nécessaire à d'autres fonctions de l'API.

Si le dossier n'existe pas, un code d'erreur le signalera.

Exemple (C++)

```
// Test d'existence du DOSSIERGED=12005 de l'armoire MDPH
// Les droits de l'utilisateurs ADMIN seront pris en compte
GedInitConnection([SERVEURGED],[PORT],0);
GedAddChampRecherche(« DOSSIERGED », »12005″);
IdMgest = GedDossierExist(« MDPH », »ADMIN »);
```

GedDossierCreate

GedDossierCreate permet de créer un dossier GED (« INSERT ») en correspondance avec l'armoire définie dans l'administration de Multigest. La

création d’un dossier GED tient compte de tous les paramètres définis pour l’armoire : champs obligatoires ou uniques, création automatique des dossiers.

```
long GedDossierCreate(
  char *Armoire,
  char *User,
  char *Force
  );
```

Paramètres

Armoire : Nom physique de l’armoire

User : Login de l’utilisateur

Force : Force la création du dossier (0 par défaut)

Codes retour

Cette fonction retourne 0 ou un code erreur.

Exemple (C++)

```
// Création d'un dossier GED dans l'armoire COURRIER
GedInitConnection([SERVEURGED],[PORT],0);
GedAddChampRequete(« NUMERO », »27225″);
GedAddChampRequete(« EXPEDITEUR », »ERIC ARCHIVAGE »);
GedAddChampRequete(« DATE », »2010-02-28″);
GedAddChampRequete(« OBJET », »Documentation technique »);
RetValue = GedDossierCreate(« COURRIER », »PIERRE », »0″);
```

GedDossierUpdate

GedDossierUpdate permet de modifier un dossier GED (« UPDATE ») en correspondance avec l’armoire définie dans l’administration de Multigest.

```
long GedDossierUpdate(
  char *Armoire,
  char *User
  );
```

Paramètres

Armoire : Nom physique de l’armoire

User : Login de l’utilisateur

Codes retour

Cette fonction retourne 0 ou un code erreur.

Exemple (C++)

```
// Modification d'un dossier GED dans l'armoire COURRIER
// Modification du champ OBJET
GedInitConnection([SERVEURGED],[PORT],0);
GedAddChampRecherche(« NUMERO », »27225″);
GedAddChampRequete(« OBJET », »Documentation technique API »);
RetValue = GedDossierUpdate(« COURRIER », »PIERRE »);
```

GedDossierDelete

GedDossierDelete permet de supprimer un dossier GED (« DELETE »). Le dossier sera supprimé que s'il est vide (aucun document).

```
int GedDossierDelete(
  char *Armoire,
  char *User,
  char *Force
  );
```

Paramètres

Armoire : Nom physique de l'armoire

User : Login de l'utilisateur

Force : Force la suppression du dossier GED

Codes retour

Cette fonction retourne 0 ou un code erreur.

Exemple (C++)

```
// Suppression d'un dossier GED dans l'armoire COURRIER
GedInitConnection([SERVEURGED],[PORT],0);
GedAddChampRecherche(« NUMERO », »27225″);
RetValue = GedDossierDelete(« COURRIER », »PIERRE »,1);
```

GedGenererNumeroClassement

GedGenererNumeroClassement permet de générer un numéro de classement pour les mails ou pour créer des codes à barres nominatifs.

Ces codes à barres peuvent être générés dans plusieurs formats :

- Format A4 (séparateur ERIC) si l'argument « Format » est égale à la

valeur « PDF »
- Format image si l'argument « Format » prend une des valeurs « BMP », « JPG », « PNG ».

```
int GedGenererNumeroClassement(
  char *Armoire,
  char *User,
  char *Libelle1,
  char *Libelle2,
  char *Sd,
  char *Ssd,
  char *NomFile,
  char *Format,
  long *Largeur,
  long *Hauteur
  );
```

Paramètres

Armoire : Nom physique de l'armoire

User : Login de l'utilisateur

Libelle1 : Libellé de description 1 qui apparaitra sur le séparateur généré.

Libelle2 : Libellé de description 2 qui apparaitra sur le séparateur généré.

Sd : Nom du sous-dossier de classement

Ssd : Nom du sous-sous-dossier de classement

NomFile : Nom du fichier dans la GED (par défaut : « Nouveau fichier »)

Format : Format d'impression du code à barres (« PDF » par défaut)

Largeur : Largeur de l'image (pixels) code à barres (si « Format » est « JPG, « BMP », « PNG »)

Hauteur : Hauteur de l'image (pixels) code à barres (si « Format » est « JPG, « BMP », « PNG »)

Codes retour

Numéro de session (18 caractères) ou code erreur

Exemple (C++)

```
// Génération d'un numéro de classement pour
// – Armoire RSA – Généré par PIERRE
// – DOSSIERRSA = 89652
// – Classement = Demande/Formulaire
// – Libellé 1 du séparateur : Dossier de Jean Dupont
```

```
// – Libellé 2 du séparateur : Formulaire de demande
GedInitConnection([SERVEURGED],[PORT],0);
GedAddChampRecherche(« DOSSIERRSA », »89652″);
NoSep = GedGenereNumeroClassement(« RSA »,
                                  « PIERRE »,
                                  « Dossier de jean Dupont »,
                                  « Formulaire de demande »,
                                  « Demande »,
                                  «  »,
                                  « Formulaire »);
```

GedPopupScan

GedPopupScan permet d'ouvrir le popup de numérisation de Multigest Webserveur. Une option permettra de lancer la numérisation dès l'ouverture du popup. Le document numérisé sera automatiquement classé dans le dossier à la validation de la numérisation et le popup sera fermé.

```
Char* GedPopupScan(
  char *Armoire,
  char *User,
  char *Sd,
  char *Ssd,
  char *NomFile,
  int AutoStart,
  char *RefDoc,
  char *TypeDoc
  );
```

Paramètres

Armoire : Nom physique de l'armoire

User : Login de l'utilisateur

Sd : Nom du sous-dossier de classement (facultatif)

Ssd : Nom du sous-sous-dossier de classement (facultatif)

NomFile : Nom du fichier dans la GED (par défaut : « Nouveau fichier »)

AutoStart : Démarrage automatique du scanner

RefDoc : Référence du document (donnée par l'application métier)

TypeDoc : Autre référence du document (donnée par l'application métier)

Codes retour

Numéro de session (18 caractères) ou code erreur

Exemple (C++)

```
// Appel du popup de scan
// – Armoire COURRIER
// – NUMEROCHRONO = 111545
// – Classement = Courrier entrant
// – lancement de la numérisation à l'ouverture du popup
// – Référence du document : REF545
GedInitConnection([SERVEURGED],[PORT],0);
GedAddChampRecherche(« NUMEROCHRONO », »111545″);
NoSession = GedPopupScan( « COURRIER »,
                          « JACQUES »,
                          «  »,
                          «  »,
                          « Courrier entrant »,
                          1,
                          « REF545 »,
                          «  »);
```

GedImporterDocument

GedImporterDocument permet d'importer un document (à son format d'origine) dans la ged depuis n'importe quel poste client.

```
Char* GedImporterDocument (
  char *Armoire,
  char *User,
  char *FileSource,
  char *Sd,
  char *Ssd,
  char *NomFile,
  int Convert,
  int Decoupage,
  int PdfA,
  char RefDoc,
  char TypeDoc,
  char IdDT,
  char Suddixe,
  int ModeDiffere
  );
```

Paramètres

Armoire : Nom physique de l'armoire

User : Login de l'utilisateur

FileSource : Chemin complet du fichier source

Sd : Nom du sous-dossier de classement (facultatif)

Ssd : Nom du sous-sous-dossier de classement (facultatif)

NomFile : Nom du fichier dans la GED (par défaut : « Nouveau fichier »)

Convert : Conversion au format PDF (0/1) : 0 par défaut

Decoupage : Découpage sur balise « GED= » (0/1): 0 par défaut

PdfA : 0 par défaut (argument obsolète)

RefDoc : Référence du document (donnée par l'application métier seulement par les exe mode 1)

TypeDoc : Autre référence du document (donnée par l'application métier)

IdDT : Numéro Multigest du dossier de travail (0 = pas de dossier de travail)

Suffixe : Suffixe à ajouter lors de l'affichage (Libellé personnalisé)

ModeDiffere : Traitement différé (-1: import immédiat, 0: import dès que possible, 1: pas d'import, sert pour éditer le document avant intégration)

Codes retour

Numéro de session (18 caractères) ou Id système du document si ModeDiffere=-1 ou code erreur

Mode UID activé

Ce paragraphe surcharge les indications données précédemment.

- Codes retour :

Numéro de session (18 caractères) ou UID du document si ModeDiffere=-1 ou code erreur

Exemple (C++)

```
// Import d'un document
// – Armoire COURRIER
// – NUMEROCHRONO = 111545
// – Classement = Réponses/Recommandé
// – Conversion doc vers pdf
// – Référence du document : REF546
GedInitConnection([SERVEURGED],[PORT],0);
GedAddChampRecherche(« NUMEROCHRONO », »111545″);
NoSession = GedImporterDocument( « COURRIER »,
                                 « JACQUES »,
                                 « c:\Mes documents\Réponse REF546.doc »,
                                 « Réponses »,
                                 «  »,
```

```
« Recommandé »,
1,
0,
0,
« REF546 »,
«  »);
```

GedImporterDocumentStream

GedImporterDocumentStream permet d’importer un document (à son format d’origine) dans la ged depuis n’importe quel poste client.

```
Char* GedImporterDocumentStream (
  char *Armoire,
  char *User,
  char *FileSourceStream,
  char *Sd,
  char *Ssd,
  char *NomFile,
  char *Ext,
  int Convert,
  int Decoupage,
  int PdfA,
  char RefDoc,
  char TypeDoc,
  char IdDT,
  char Suddixe,
  int ModeDiffere
  );
```

Paramètres

Armoire : Nom physique de l’armoire

User : Login de l’utilisateur

FileSource : Fichier source au format base64

Sd : Nom du sous-dossier de classement

Ssd : Nom du sous-sous-dossier de classement

NomFile : Nom du fichier dans la GED

Ext : Extension du fichier dans la GED

Convert : Conversion au format PDF (0/1) : 0 par défaut

Decoupage : Découpage sur balise « GED= » (0/1): 0 par défaut

PdfA : 0 par défaut (argument obsolète)

RefDoc : Référence du document (donnée par l’application métier)

TypeDoc : Autre référence du document (donnée par l’application métier)

IdDT : Numéro Multigest du dossier de travail (0 = pas de dossier de travail)

Suffixe : Suffixe à ajouter lors de l’affichage (Libellé personnalisé)

ModeDiffere : Traitement différé (-1: import immédiat, 0: import dès que possible, 1: pas d’import, sert pour éditer le document avant intégration)

Codes retour

Numéro de session (18 caractères) ou IdDoc si ModeDiffere=-1 ou code erreur

Mode UID activé

Ce paragraphe surcharge les indications données précédemment.

- Codes retour :

Numéro de session (18 caractères) ou UID si ModeDiffere=-1 ou code erreur

Exemple (PHP)

```
// Import d'un document
// – Armoire COURRIER
// – NUMEROCHRONO = 111545
// – Classement = Réponses/Recommandé
// – Conversion doc vers pdf
// – Référence du document : REF546
$ApiGed = new
SoapClient('http://srvGED/Interconnexion/SoapService.php?wsdl');
$ApiGed->GedAddChampRecherche(« NUMEROCHRONO », »111545″);
NoSession = $ApiGed->GedImporterDocument( « COURRIER »,
                                          « JACQUES »,
                                          base64_encode(file_get_contents(«
c:\Mes documents\Réponse REF546.doc »)),
                                          « Réponses »,
                                          «  »,
                                          « Recommandé »,
                                          « doc »,
                                          1,
                                          0,
                                          0,
                                          « REF546 »,
                                          «  »);
```

GedDossierLinkedFolders

GedDossierLinkedFolders retourne la liste des dossiers liés du dossier GED

passé en paramètre en fonction des habilitations de l'utilisateur transmis aussi en paramètre. Cette liste est regroupée par lien inter-armoire.

```
Char* GedDossierLinkedFolders (
  char *armoire,
  char *login,
  char *idDossier
  );
```

Paramètres

Armoire : Nom physique de l'armoire

User : Login de l'utilisateur

idDossier : Identifiant système du dossier GED

Codes retour

Chaine XML ou code erreur :

- Chaîne XML – Exemple :

```
<?xml version=« 1.0 » encoding=« UTF-8 »?>
<items>
        <item name=« EFALIA – 21263 »>
                <gedFolderId>44813</gedFolderId>
                <cabinetId>2</cabinetId>
                <cabinetName>FACTURES</cabinetName>
                <linkId>9</linkId>
                <linkName>Lien 1</linkName>
        </item>
        <item name=« APPIC – 23659 « >
                <gedFolderId>54636</gedFolderId>
                <cabinetId>1</cabinetId>
                <cabinetName>DEVIS</cabinetName>
                <linkId>12</linkId>
                <linkName>Lien 2</linkName>
        </item>
</items>
```

- Code erreur : -1, -2, -4, -15, -28 – Explications voir Annexe – Liste des codes retour

Exemple (php/SOAP)

```
$ApiGed = new
SoapClient('http://srvGED/Interconnexion/SoapService.php?wsdl');
                          $result =
$ApiGed->GedDossierLinkedFolders('CLIENTS', 'admin', 56326);
```

```
echo $result;
```

GedDossierTreeContent

GedDossierTreeContent retourne la liste arborescente des documents du dossier GED passé en paramètre en fonction des habilitations de l'utilisateur transmis aussi en paramètre.

```
Char* GedDossierTreeContent (
char *armoire,
char *login,
int *idDossier,
int getDocMetadatas,
int useUserSortSettings
);
```

Paramètres

Armoire : Nom physique de l'armoire

User : Login de l'utilisateur

idDossier : Identifiant système du dossier GED

getDocMetadatas :

- 1 : les metadatas des documents et dossiers de travail seront retournés dans la chaîne XML
- 0 : Valeur par défaut

useUserSortSettings

- Si 0 (valeur par défaut) : arborescence triée par défaut (ordre alphabétique ascendant)
- Si 1 : arborescence triée en fonction du paramétrage de l'utilisateur s'il en existe un

Codes retour

Chaine XML ou code erreur :

- Chaîne XML – Exemple :

```
<?xml version=« 1.0 » encoding=« UTF-8 »?>
<items>
    <item name=« Dossier de demande 123456 » text=« Dossier de demande 123456
» type=« DT »>
        <idd>190</idd>
        <iddt>43</iddt>
        <is_readable>1</is_readable>
```

```
        <is_alterable>1</is_alterable>
        <is_printable>1</is_printable>
        <is_addable>1</is_addable>
        <is_removable>1</is_removable>
        <metadatas>
            <metadata name=« NUM »>654321</metadata>
        </metadatas>
        <item name=« 02 – Factures » type=« SD »>
            <idd>65</idd>
            <iddt>43</iddt>
            <sd>02 – Factures</sd>
            <ssd></ssd>
            <is_readable>1</is_readable>
            <is_alterable>1</is_alterable>
            <is_printable>1</is_printable>
            <is_addable>1</is_addable>
            <is_removable>1</is_removable>
            <item name=« Facture_001.pdf » type=« F »>
                <iddoc>954<iddoc>
                <iddt>43</iddt>
                <uid>6566</uid>
                <idd>326</idd>
                <sd>02 – Factures </sd>
                <ssd></ssd>
                <file>Facture_001.pdf</file>
                <ext>pdf</ext>
                <is_readable>1</is_readable>
                <is_alterable>1</is_alterable>
                <is_printable>1</is_printable>
                <is_addable>1</is_addable>
                <is_removable>1</is_removable>
                <size>6169</size>
                <usercrea>6</usercrea>
                <datecrea>25/01/2012 17:31:41</datecrea>
                <userrev></userrev>
                <daterev></daterev>
                <typedoc></typedoc>
                <sid>1</sid>
                <suffixe></suffixe>
                <refdoc></refdoc>
                <metadatas>
                    <metadata name=« NUM »>123456</metadata>
                </metadatas>
            </item>
        </item>
    </item>
    <item name=« 01 – Administratif » text=« 01 – Administratif » TYPE=« SD
»>
        <idd>75</idd>
        <iddt>0</iddt>
        <sd>01 – Administratif</sd>
```

```
        <ssd></ssd>
        <is_readable>1</is_readable>
        <is_alterable>1</is_alterable>
        <is_printable>1</is_printable>
        <is_addable>1</is_addable>
        <is_removable>1</is_removable>
        <item name="A – Identité" text="A – Identité" type ="SSD">
            <idd>89</idd>
            <iddt>0</iddt>
            <sd>A – Identité</sd>
            <ssd></ssd>
            <is_readable>1</is_readable>
            <is_alterable>1</is_alterable>
            <is_printable>1</is_printable>
            <is_addable>1</is_addable>
            <is_removable>1</is_removable>
            <item name=« Carte identité.pdf » text=« Carte identité.pdf –
01/07/2019 – Entré en GED le 02/07/2019 » type=« F »>
                <iddoc>954<iddoc>
                <iddt>0</iddt>
                <uid>6566</uid>
                <idd>326</idd>
                <sd>02 – Factures </sd>
                <ssd></ssd>
                <file>Facture_001.pdf</file>
                <ext>pdf</ext>
                <typedoc></typedoc>
                <is_readable>1</is_readable>
                <is_alterable>1</is_alterable>
                <is_printable>1</is_printable>
                <is_addable>1</is_addable>
                <is_removable>1</is_removable>
                <size>82169</size>
                <usercrea>2</usercrea>
                <datecrea>10/10/2014 09:33:36</datecrea>
                <userrev></userrev>
                <daterev></daterev>
                <sid>1</sid>
                <suffixe></suffixe>
                <refdoc></refdoc>
                <metadatas>
                    <metadata name=« DATE_DELIVRANCE »>01/07/2019</metadata>
                    <metadata name=« DATE_FIN »>01/07/2029</metadata>
                </metadatas>
            </item>
        </item>
    </item>
</items>
```

Explications sur les balises et attributs du XML :

- Balise <*item*> : Nœud correspondant à un élément du plan de classement
    - Attribut *type* :
        - **DT** = **Dossier de Travail**
        - **SD** = **Sous dossier**
        - **SSD** = **Sous sous dossier**
        - **F** = **Fichier**
    - Attribut *name* : nom dans le plan de classement
    - Attribut *text* : concaténation du nom dans le plan de classement et éventuellement de méta-données paramétrées en affichage et d'information complémentaire (ex : *Entré en GED le dd/mm/aaaa*

Balises communes aux nœuds de type DT, SD, SSD, F :

- Balise <idd> : id de plan de classement
- Balise <iddt> : id de dossier de travail auquel est rattaché l'élément (valeur 0 si rattaché à aucun dossier de travail)
- Balise <is_readable> : si 1 alors le document est consultable
- Balise <is_alterable> : si 1 alors le document est modifiable
- Balise <is_printable> : si 1 alors le document est imprimable
- Balise <is_addable> : si 1 alors un document peut être ajouté sur ce plan de classement
- Balise <is_removable> : si 1 alors le document est supprimable

Balises communes aux nœuds de type SD, SSD, F :

- Balise <sd> : libellé de sous-dossier auquel est rattaché l'élément
- Balise <ssd> : libellé de sous sous-dossier auquel est rattaché l'élément

Balises communes aux nœuds de type DT et F :

- Balise <sid> : id de la fiche nature rattachée au plan de classement du dossier de travail ou du document
- Balise <metadatas> : balise englobante regroupant les balises *metadata*
- Balise <metadata> : balise correspondant à un champ de fiche nature
    - Attribut *name* : nom physique du champ de fiche nature

Balises exclusives aux nœuds de type F :

- Balise <iddoc> : id de document
- Balise <uid> : identifiant unique du document dans toute la GED
- Balise <file> : nom de fihcier (avec l'extension)
- Balise <ext> : extension de fichier seule
- Balise <size> : taille en octets du document
- Balise <usercrea> : id utilisateur créateur du document
- Balise <datecrea> : date de création en GED du document
- Balise <userrev> : id du dernier utilisateur ayant modifié le document
- Balise <daterev> : date de dernière modification du document
- Balise <typedoc> : champ disponible pour des applications externes
- Balise <suffixe> : champ suffixe, meta-donnée GED du document
- Balise <refdoc> : champ référence, meta-donnée GED du document
- Code erreur : -15 – Explications voir Annexe – Liste des codes retour

Exemple (php/SOAP)

```
$ApiGed = new
SoapClient('http://srvGED/Interconnexion/SoapService.php?wsdl');
                          $result = $ApiGed->GedDossierTreeContent ('CLIENTS',
'admin', 56326, 1);
echo $result;
```

GedCreateDirectory

GedCreateDirectory permet pour les armoires en « mode dossier », de créer des sous-dossiers ou des sous-sous-dossiers de classement dans les dossiers GED.

```
int GedCreateDirectory(
  char *Armoire,
  char *User,
  char *Sd,
  char *Ssd
  );
```

Paramètres

Armoire : Nom physique de l'armoire

User : Login de l'utilisateur

Sd : Nom du sous-dossier de classement (obligatoire)

Ssd : Nom du sous-sous-dossier de classement (facultatif)

Codes retour

0 ou code erreur

Exemple (C++)

```
// Création d'un sous-dossier de classement
// – Armoire MDPH
// – DOSSIERGED = 111545
// – Sous-dossier = Médical
GedInitConnection([SERVEURGED],[PORT],0);
GedAddChampRecherche(« DOSSIERGED », »111545″);
NoSession = GedCreateDirectory( « MDPH »,
                                « JACQUES »,
                                « Médical »,
                                «  »);
```

GedExistDossierTravail

GedExistDossierTravail permet pour les armoires en « mode dossier », de vérifier l'existence d'un dossier de travail (fonctionnalité standard de Multigest : voir documentation Multigest).

Rappel : un dossier de travail Multigest correspond à un sous-dossier habilité, associé à des métadonnées. Chaque document présent dans un dossier de travail hérite de toutes les options de ce dossier de travail.

Un dossier de travail s'identifie par une clé unique passée en métadonnée.

Ces métadonnées associées peuvent être standard (une référence ou un libellé) ou personnalisées par une fiche de métadonnées. Dans ce dernier cas, un champ de la fiche métadonnées est utilisée comme clé unique d'identification du dossier de travail.

Multigest identifie chaque dossier de travail par une clé unique « IdDT ».

```
long __stdcall GedExistDossierTravail(
  char *Armoire,
  char *User,
  long IdMgest,
  char *RefDT,
  char *ModelMetadata
  );
```

Paramètres

Armoire : Nom physique de l'armoire

User : Login de l'utilisateur

IdMgest : Identifiant unique du dossier GED (retourné par la fonction GedDossierExist)

RefDT : Référence unique ou libellé unique affectée par l'application métier (facultatif)

ModelMetadata : Modèle de la fiche métadonnée affectée (facultatif)

Codes retour

Numéro unique Multigest du dossier de travail (IdDT) ou code erreur

Exemple (C++)

```
// Existence dossier de travail (pas de fiche métadonnées)
// – Armoire MDPH
// – Identifiant Multigest du dossier GED = 4444
// – Référence ou libelle dossier de travail = 123456
```

```
GedInitConnection([SERVEURGED],[PORT],0);
IdDT = GedExistDossierTravail( « MDPH »,
                               « JACQUES »,
                               « 4444 »,
                               « 123456 »);

// Existence dossier de travail (avec fiche métadonnées)
// – Armoire MDPH
// – Identifiant Multigest du dossier GED = 4444
// – Référence ou libelle dossier de travail = 123456
// – Clé unique métadonnées (NUMDOSS = 556677)
// – libellé métadonnées (LIBDOSS = ESSAI)
GedInitConnection([SERVEURGED],[PORT],0);
GedAddChampRecherche(« NUMDOSS », »556677″);
IdDT = GedExistDossierTravail( « MDPH »,
                               « JACQUES »,
                               « 4444 »,
                               « 123456 »,
                               « Fiche1 »);
```

GedCreateDossierTravail

GedCreateDossierTravail permet (pour les armoires en « mode dossier »), de créer un dossier de travail (fonctionnalité standard de Multigest : voir documentation Multigest).

3 modes de création avec ou sans fiche de métadonnées.

- Aucune référence au dossier de travail : l'application métier doit stocker le numéro Multigest « IdDT ».
- Une référence unique est passée à Multigest.
- Une liste de métadonnées est passée à Multigest. Ces métadonnées sont utilisées pour identifier le dossier de travail (définissent une clé unique).

Les métadonnées passées ont au préalable été déclarée dans une fiche de métadonnées (Fiche1) dans l'administration de Multigest.

```
long __stdcall GedCreateDossierTravail(
  char *Armoire,
  char *User,
  long IdMgest,
  char *RefDT,
  char *ModelMetadata
  );
```

Paramètres

Armoire : Nom physique de l'armoire

User : Login de l'utilisateur

IdMgest : Identifiant unique du dossier GED (retourné par la fonction GedDossierExist)

RefDT : Référence unique ou libellé unique affectée par l'application métier

ModelMetadata : Libellé du modèle de dossier de travail

Codes retour

Numéro unique Multigest du dossier de travail (IdDT) ou code erreur

Exemple (C++)

```
// Création dossier de travail (pas métadonnées)
// – Armoire MDPH
// – Utilisateur JACQUES
// – Identifiant Multigest du dossier GED = 4444
// – Libelle du dossier de travail à créer = 123456
// – Libellé du modèle de dossier de travail = Demande
GedInitConnection([SERVEURGED],[PORT],0);
IdDT = GedCreateDossierTravail( « MDPH »,
                                « JACQUES »,
                                « 4444 »,
                                « 123456 »,
                                « Demande »);

// Création dossier de travail (avec de fiche métadonnées Fiche1)
// – Armoire MDPH
// – Utilisateur JACQUES
// – Identifiant Multigest du dossier GED = 4444
// – Libelle du dossier de travail à créer = 123456
// – Libellé du modèle de dossier de travail = Demande
// – Champs de métadonnées 1 (NUMDOSS = 556677)
// – Champs de métadonnées 2 (LIBDOSS = ESSAI)
GedInitConnection([SERVEURGED],[PORT],0);
GedAddChampRecherche(« NUMDOSS », « 556677 »);
GedAddChampRecherche(« LIBDOSS », « ESSAI »);
IdDT = GedCreateDossierTravail( « MDPH »,
                                « JACQUES »,
                                « 4444 »,
                                « 123456 »,
                                « Demande avec méta »);
```

GedCreateDocumentToDossierTravailFromModeles

GedCreateDocumentToDossierTravailFromModeles permet (pour les armoires en « mode dossier »), de créer un fichier dans un dossier de travail à partir d'un modèle de document.

```
long __stdcall GedCreateDossierTravail(
  char *Armoire,
  char *User,
  long IdMgest,
  char *IDDT,
  char *classementModel,
  char *RefDT
  );
```

Paramètres

Armoire : Nom physique de l’armoire

User : Login de l’utilisateur

IdMgest : Identifiant unique du dossier GED (retourné par la fonction GedDossierExist)

IDDT : Identifiant du dossier de travail concerné

classementModel : Modèle du document (si vide, génération automatique de tous les documents configurés en création automatique dans le dossier de travail via modèle)

RefDT : Référence unique ou libellé unique affectée par l’application métier

Codes retour

0 ou code erreur

Exemple (php)

```
$ApiGed = new
SoapClient("http://localhost/Interconnexion/SoapService.php?className=APIMult
igest&style=rpc&styletype=encoded");

$ret = $ApiGed->GedCreateDocumentToDossierTravailFromModeles("MDHH", "admin",
"1234","5678", " ", "Demande");
```

GedModifyDossierTravail

GedModifyDossierTravail permet de modifier (mettre à jour) les métadonnées d’un dossier de travail :

- Métadonnées de la référence unique passées par l’application métier.
- Métadonnées de la fiche de métadonnées associée au dossier de travail.

Cette modification des métadonnées ne modifie pas la référence unique passée par Multigest.

```
long __stdcall GedModifyDossierTravail(
  char *Armoire,
  char *User,
  long IdMgest,
  long IdDT,
  char *RefDT = ""
  );
```

Paramètres

Armoire : Nom physique de l’armoire

User : Login de l’utilisateur

IdMgest : Identifiant unique du dossier GED (retourné par la fonction GedDossierExist)

IdDT : Identifiant unique Multigest

RefDT : Libellé du dossier de travail

Codes retour

0 ou code erreur

Exemple (SOAP)

```
$ApiGed = new SoapClient(« http://localhost/Interconnexion/SoapService.php?className=APIMultigest&style=rpc&styletype=encoded »);
$ApiGed->GedAddChampRecherche(« NUMDOSS », « 556677 »);
$ApiGed->GedAddChampRecherche(« LIBDOSS », « ESSAI »);
$ret = $ApiGed->GedModifyDossierTravail(« MDHH », « admin », « 1234 », « 5678 », « Demande »);
```

GedClotureDossierTravail

GedClotureDossierTravail permet de cloturer le dossier de travail. Les documents présents dans le dossier sont reclassés dans le plan de classement standard.

```
long __stdcall GedClotureDossierTravail(
  char *Armoire,
  char *User,
  long IdMgest,
  long *IdDT
  );
```

Paramètres

Armoire : Nom physique de l’armoire

User : Login de l’utilisateur

IdMgest : Identifiant unique du dossier GED (retourné par la fonction GedDossierExist)

IdDT : Identifiant unique Multigest

Codes retour

0 ou code erreur

Exemple (C++)

```
// Création dossier de travail (avec de fiche métadonnées Fiche1)
// – Armoire MDPH
// – Identifiant Multigest du dossier GED = 4444
// – Identifiant Multigest du dossier de travail = 111222
GedInitConnection([SERVEURGED],[PORT],0);
IdDT = GedClotureDossierTravail( « MDPH »,
                                 « JACQUES »,
                                 4444,
                                 111222);
```

GedDeleteDossierTravail

GedDeleteDossierTravail permet de supprimer le dossier de travail. Cette opération est possible uniquement si le dossier de travail est vide.

```
long __stdcall GedDeleteDossierTravail(
  char *Armoire,
  char *User,
  long IdMgest,
  long *IdDT
  );
```

Paramètres

Armoire : Nom physique de l’armoire

User : Login de l’utilisateur

IdMgest : Identifiant unique du dossier GED (retourné par la fonction GedDossierExist)

IdDT : Identifiant unique Multigest

Codes retour

0 ou code erreur

Exemple (C++)

```
// Création dossier de travail (avec de fiche métadonnées Fiche1)
// – Armoire MDPH
// – Identifiant Multigest du dossier GED = 4444
// – Identifiant Multigest du dossier de travail = 111222
GedInitConnection([SERVEURGED],[PORT],0);
IdDT = GedClotureDossierTravail( « MDPH »,
                                 « JACQUES »,
                                 4444,
                                 111222);
```

GedGetDossierTravail

GedGetDossierTravail permet de récupere au format xml la liste des dossiers de travail présents dans un dossier GED. Cette opération est possible uniquement si le dossier GED utilisé existe.

```
int GedGetDossierTravail(
  char *Armoire,
  char * User,
  char * IdMgest
  );
```

Paramètres

Armoire : Nom physique de l’armoire

User : Login de l’utilisateur

IdMgest : Identifiant unique du dossier GED (retourné par la fonction GedDossierExist)

Codes retour

code erreur ou arborescence XML:

```
<dossiers>
  <dossier id= "ID_DOSSIERTRAVAIL">
    <index name="IDDT">1</index>
    <index name="LIBELLE">Dossier demande 7328</index>
    <index name="ETAT">1</index>
    <index name="DATECREA">2013-09-01</index>
    <index name="DATEFIN"></index>
    <index name="USERCREA">admin</index>
    <index name="USERCREA">admin</index>
    <mdatas>
      <data name="CHAMP_1">2013-09-01</data>
      <data name="CHAMP_2">En cours</data>
```

```
      <data name="CHAMP_3">123.69</data>
      …
    </mdatas>
    …
  </dossier>
  …
</dossiers>
```

Exemple (php/SOAP)

```
$ApiGed = new
SoapClient('http://srvGED/Interconnexion/SoapService.php?wsdl');
$result = $ApiGed->GedGetDossierTravail('MPHH','ADMIN','83545´);
```

ou

```
$ApiGed = new
SoapClient('http://srvGED/Interconnexion/SoapService.php?wsdl');
$result = $ApiGed->__soapCall('GedGetDossierTravail',
                              array('Armoire'=>'MPHH', 'User'=>'ADMIN',
'IdMgest'=>'83545')
                              );
```

GedDeleteDirectory

GedDeleteDirectory pour les armoires en « mode dossier », permet de supprimer des sous-dossiers et/ou sous-sous-dossiers de classement dans les dossiers GED. Les fichiers pouvant être contenus dans cette arborescence seront également supprimés.

```
int GedDeleteDirectory(
  char *Armoire,
  char *User,
  char *Sd,
  char *Ssd
  );
```

Paramètres

Armoire : Nom physique de l’armoire

User : Login de l’utilisateur

Sd : Nom du sous-dossier de classement (obligatoire)

Ssd : Nom du sous-sous-dossier de classement (facultatif)

Codes retour

0 ou code erreur

Exemple (C++)

```
// Suppression d'un sous-sous-dossier de classement
// – Armoire MDPH
// – DOSSIERGED = 111545
// – Sous-sous-dossier = Médical/Ordonnance
GedInitConnection([SERVEURGED],[PORT],0);
GedAddChampRecherche(« DOSSIERGED », »111545″);
NoSession = GedDeleteDirectory( « MDPH »,
                                « JACQUES »,
                                « Médical »,
                                « Ordonnance »);
```

GedFusionDossiers

GedFusionDossiers permet de fusionner le contenu de 2 dossiers GED.

```
int GedFusionDossiers(
  char *Armoire,
  char *User
  );
```

Paramètres

Armoire : Nom physique de l’armoire

User : Login de l’utilisateur

Codes retour

0 ou code erreur

Exemple (C++)

```
// Suppression d'un sous-sous-dossier de classement
// – Armoire MDPH
// – DOSSIERGED = 111545
// – Sous-sous-dossier = Médical/Ordonnance
GedInitConnection([SERVEURGED],[PORT],0);
GedAddChampRecherche(« DOSSIERGED », »111111″);
GedAddChampRecherche(« DOSSIERGED », »222222″);
NoSession = GedFusionDossiers( « MDPH »,
                               « JACQUES);
```

Dans cet exemple, le contenu du dossier « 111111 » est déplacé dans le dossier « 222222 ». Les dates des documents déplacés ne sont pas modifiées.

GedApurementDossier

GedApurementDossier permet d'apurer un dossier GED (le vider de son contenu). Seul le contenu du dossier GED est concerné.

```
int GedApurementDossier(
  char *Armoire,
  char *User
  );
```

Paramètres

Armoire : Nom physique de l'armoire

User : Login de l'utilisateur

Codes retour

0 ou code erreur

Exemple (C++)

```
// Suppression d'un sous-sous-dossier de classement
// – Armoire MDPH
// – DOSSIERGED = 111545
// – Sous-sous-dossier = Médical/Ordonnance
GedInitConnection([SERVEURGED],[PORT],0);
GedAddChampRecherche(« DOSSIERGED », »111111″);
NoSession = GedApurementDossier ( « MDPH »,
                                  « JACQUES);
```

GedDossierListFromDate

GedDossierListFromDate permet de lister l'ensemble des dossiers GED créés depuis une date.

```
int GedApurementDossier(
  char *Armoire,
  char *Date
  );
```

Paramètres

Armoire : Nom physique de l'armoire

Date : Date de création des dossiers (format anglais aaaa/mm/jj[hh:mm:ss])

Codes retour

code erreur ou arborescence XML:

```
<dossiers>
  <dossier id= »ID_DOSSIERGED »>
    <index name= »NOM_PHYSIQUE »>VALEUR</index>
    <index name= »NOM_PHYSIQUE »>VALEUR</index>
    …
  </dossier>
  <dossier id= »ID_DOSSIERGED »>
    <index name= »NOM_PHYSIQUE »>VALEUR</index>
    <index name= »NOM_PHYSIQUE »>VALEUR</index>
    …
    </dossier>
  …
</dossiers>
```

Exemple (php/SOAP)

```
$ApiGed = new
SoapClient('http://srvGED/Interconnexion/SoapService.php?wsdl');
$result = $ApiGed->GedDossierListFromDate('MPHH','2013-05-30´);
echo $result;
```

ou

```
$ApiGed = new
SoapClient('http://srvGED/Interconnexion/SoapService.php?wsdl');
$result = $ApiGed->__soapCall('GedDossierListFromDate',
                              array('Armoire'=>'MPHH', 'Date'=>'2013-05-30')
                              );
echo $result;
```