

Connecteurs Multigest V2

Préambule

Le but de ce document est de détailler les différents connecteurs et agents Multigest contenu dans Efallia Process. La configuration ne s'axe plus sur l'utilisation d'un fichier de configuration JSON, chaque connecteur est donc autonome et a ses propres paramètres.

Dans les paramètres en entrée des connecteurs, si le paramètre est entre crochet [paramètre], cela signifie qu'il est optionnel.

A lire absolument !

Les payload JSON

Dans les JSON dynamiques fournis aux connecteurs si, en regard des propriétés, le texte correspond à un nom interne de champs trouvé dans le document Efallia Process, alors la valeur du champ est utilisé. Par contre si vous ne souhaitez pas utiliser de champ technique pour passer une valeur, il est alors possible de spécifier une valeur en la précédant du caractère #.

□ **En d'autres termes**, vous avez le choix de spécifier soit un nom interne de champ pour récupérer sa valeur, soit de spécifier directement une valeur préfixée par #.

L'utilisateur utilisé

Dans les connecteurs, lors de leurs appels, vous devez spécifier l'utilisateur Multigest utilisé. Si vous n'en spécifiez pas un (passage d'un champ vide au connecteur) alors le connecteur utilisera celui paramétré dans l'administration embarquée. Ce sera soit l'utilisateur connecté ou le compte de service déclaré dans l'administration. Rien ne vous empêche d'utiliser à un moment un utilisateur spécifique en valorisant le champ passé au connecteur avec l'utilisateur adéquat.



Connecteurs de dossier

GedDossierCreate

Ce connecteur permet de créer un dossier dans une armoire.

Classe Java

```
com.clog.workey.mg.connectors.v2.GedDossierCreate
```

Copy

Paramètres en entrée

- **armoire** : Nom physique de l'armoire
- **user** : Login de l'utilisateur
- **properties** : Objet JSON dynamique des propriétés du dossier. A minima, la propriété qui identifie le dossier est obligatoire.

```
{  
  "UUID": "UUID_FIELD"  
}
```

Copy

UUID en nom de propriété de l'objet, est le nom de la métadonnée du dossier Multigest.

UUID_FIELD en regard de la propriété ci-dessus, est le nom du champ d'où est issue la valeur.

- **[force]** : Force la création du dossier (true ou false, false par défaut)

Paramètres en sortie

- **folderId** : l'id du dossier créé.

GedDossierUpdate

Ce connecteur permet de modifier les métadonnées d'un dossier contenu dans une armoire.

Classe Java

```
com.clog.workey.mg.connectors.v2.GedDossierUpdate
```

Copy

Paramètres en entrée

- **armoire** : Nom physique de l'armoire
- **user** : Login de l'utilisateur
- **query** : Objet JSON dynamique permettant de cibler le dossier désiré, généralement il contient la propriété qui identifie le dossier.

```
{  
  "UUID": "UUID_FIELD"  
}
```

Copy

UUID en nom de propriété de l'objet, est le nom de la métadonnée qui identifie le dossier Multigest.

UUID_FIELD en regard de la propriété ci-dessus, est le nom du champ d'où est issue la valeur.

- **properties** : Objet JSON dynamique des métadonnées à modifier. Chaque couple propriété-valeur permet de spécifier une valeur qui valorisera la métadonnée identifiée par le nom de la propriété.

```
{  
  "NOM": "NOM_PERSONNE",  
  "PRENOM": "PRENOM_PERSONNE"
```

```
}
```

Copy

NOM en nom de propriété de l'objet, est le nom de la métadonnée du dossier Multigest. *NOM_PERSONNE* en regard de la propriété ci-dessus, est le nom du champ d'où est issue la valeur.

PRENOM en nom de propriété de l'objet, est le nom de la métadonnée du dossier Multigest. *PRENOM_PERSONNE* en regard de la propriété ci-dessus, est le nom du champ d'où est issue la valeur.

Paramètres en sortie

- **retCode** : code retour du traitement

GedDossierExist

Ce connecteur permet de tester l'existence d'un dossier Multigest.

Classe Java

```
com.clog.workey.mg.connectors.v2.GedDossierExist
```

Copy

Paramètres en entrée

- **armoire** : Nom physique de l'armoire
- **user** : Login de l'utilisateur
- **query** : Objet JSON dynamique permettant de cibler le dossier désiré, généralement il contient la propriété qui identifie le dossier.

```
{  
  "UUID": "UUID_FIELD"  
}
```

Copy

UUID en nom de propriété de l'objet, est le nom de la métadonnée qui identifie le dossier Multigest.

UUID_FIELD en regard de la propriété ci-dessus, est le nom du champ d'où est issue la valeur.

Paramètres en sortie

- **retCode** : si le dossier existe, la fonction renvoie l'id du dossier ou sinon un code d'erreur.

Connecteurs de document

GedImporterDocument

Ce connecteur permet de créer un contenu dans un dossier Multigest.

Classe Java

```
com.clog.workey.mg.connectors.v2.GedImporterDocument
```

Copy

Paramètres en entrée

- **armoire** : Nom physique de l'armoire
- **user** : Login de l'utilisateur
- **query** : Objet JSON dynamique permettant de cibler le dossier désiré, généralement il contient la propriété qui identifie le dossier.

```
{  
  "UUID": "UUID_FIELD"  
}
```

Copy

UUID en nom de propriété de l'objet, est le nom de la métadonnée qui identifie le dossier Multigest.

UUID_FIELD en regard de la propriété ci-dessus, est le nom du champ d'où est issue la valeur.

- **attachment** : pièce jointe à créer qui correspond à un composant PJ
- **sd** : le dossier désiré dans le plan de classement
- **ssd** : le sous-dossier désiré dans le plan de classement
- **nomFile** : le nom de document désiré dans le plan de classement

Paramètres en sortie

- **idDoc** : Id du document fraîchement créé

GedDocumentMdatasUpdate

Ce connecteur permet de modifier les métadonnées d'un document Multigest.

Classe Java

```
com.clog.workey.mg.connectors.v2.GedDocumentMdatasUpdate
```

Copy

Paramètres en entrée

- **armoire** : Nom physique de l'armoire
- **user** : Login de l'utilisateur
- **idDoc** : Id du document Multigest
- **mapping** : Objet JSON dynamique des métadonnées à modifier. Chaque couple propriété-valeur permet de spécifier une valeur qui valorisera la métadonnée identifiée par le nom de la propriété.

```
{
```

```
"NOM": "NOM_PERSONNE",  
"PRENOM": "PRENOM_PERSONNE"  
}
```

Copy

NOM en nom de propriété de l'objet, est le nom de la métadonnée du dossier Multigest. *NOM_PERSONNE* en regard de la propriété ci-dessus, est le nom du champ d'où est issue la valeur.

PRENOM en nom de propriété de l'objet, est le nom de la métadonnée du dossier Multigest. *PRENOM_PERSONNE* en regard de la propriété ci-dessus, est le nom du champ d'où est issue la valeur.

Paramètres en sortie

- **retCode** : 1 ou code erreur

Agents

CreateFolderFromInboxes

Cette agent permet de récupérer un document depuis une bannette, de créer un dossier Multigest contenant le document récupéré et de créer un processus Workey associé.

Classe Java

```
com.clog.workey.mg.agents.v2.CreateFolderFromInboxes
```

Copy

Paramètres en entrée

- **inboxIds** : Id des bannettes à scanner. Les différents Id sont séparés par le caractère | (par ex: 1|47|7)
- **folderKeyName** : Nom de la propriété Multigest utilisée pour identifier un dossier (par ex: UUID)
- **armoire** : Nom physique de l'armoire cible

- **classifying** : Chemin cible dans le plan de classement (par ex: Annexes/Archives/Document)
- **workkeyApiUrl** : URL de l'API workey à utiliser
- **workkeyProcessDNames** : Noms internes du processus cible (par ex: Process_A/Document/Role_1)
- **workkeyDocumentState** : Etat initial du document créé dans Workey
- **workkeyFieldNameTarget** : Nom interne du composant MG Folder cible
- **workkeyApiLogin** : Login de l'utilisateur utilisé pour créer le document Workey via l'API
- **workkeyApiPassword** : Mot de passe de l'utilisateur utilisé pour créer le document Workey via l'API
- **mapping** : Objet JSON des métadonnées du dossier à modifier. Chaque couple propriété-valeur permet de spécifier une valeur qui valorisera la métadonnée identifiée par le nom de la propriété.

```
{
  "PRENOM": "Léonard",
  "NOM": "de Vinci"
}
```

Copy

PRENOM en nom de propriété de l'objet, est le nom de la métadonnée du dossier Multigest. *NOM* en nom de propriété de l'objet, est le nom de la métadonnée du dossier Multigest.

- **user** : Utilisateur Multigest utilisé

Composants

Il existe deux composants utilisables avec Multigest.
Un composant Dossier MG et un composant Documents MG



Dossier MG

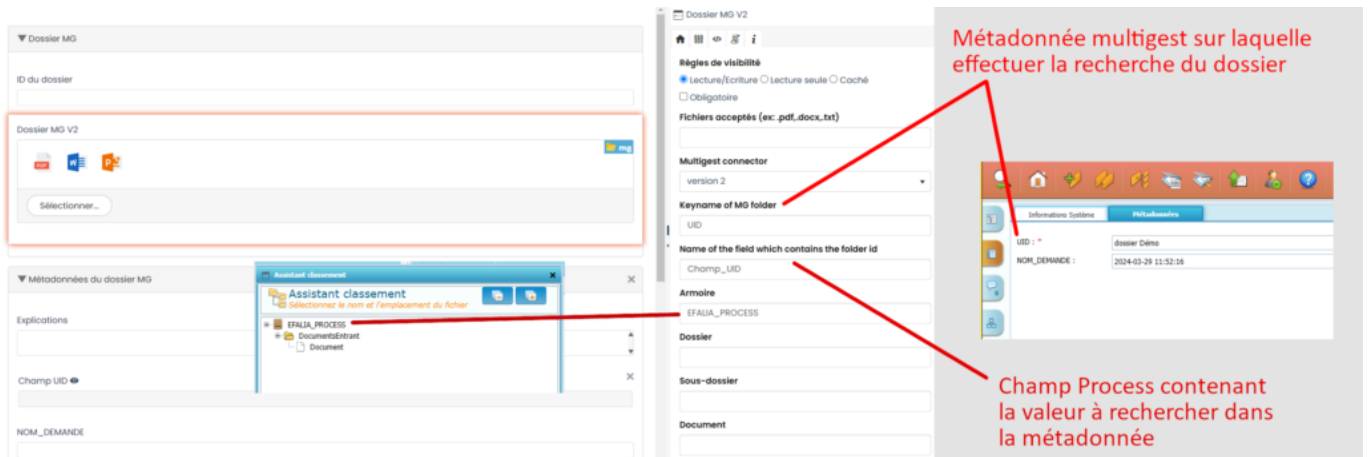


Documents
MG

Dossier MG

Le dossier MG sert à avoir une visibilité sur un dossier donné de Multigest. Il permet d'uploader, downloader les documents s'y trouvant ainsi que de les modifier par PDFtron/OnlyOffice.

Pour désigner le dossier à visualiser, il faut définir une de ses métadonnées à partir de laquelle effectuer la recherche du dossier, ainsi qu'un champ Process contenant la valeur recherchée.



Il faut aussi renseigner (à minima) le nom de l'armoire visée.

Le composant se comporte ensuite comme un composant Pièce-Jointe Process mais en pointant vers un dossier de la GED.

Document MG

Ce composant se comporte de façon semblable au composant réalisé pour Efallia Doc.

Il peut servir pour un lancement de workflow depuis la GED et il faut lui fournir des paramètres sous forme d'un json contenant :

```
{
  id: <id du document MG>,
  name: <id du document MG>,
  mimetype: <type MIME du document>,
  updatedOn: <date de mise à jour>,
  folder: <facultatif pour les fichiers sans dossier : dossier où
se trouve le document>,
  subfolder: <facultatif : sous-dossier où se trouve le document> ,
  filingCabinet: <armoire où se trouve le document>
}
```

Voir

<https://docs.efalia.com/docs/configurer-mg-9/administration-du-workflow/workflow-vers-process/>