* 1. **DétecterBrokenLinks.fmw**

Danielle Poirier Version originale 2021-04-21

FME 2020 – 20200309 - Build 20200 - WIN64))

## Description du programme

Ce programme détecte tous les liens brisés des métadonnées publiées de la PGF. Chaque lien brisé sera inscrit dans une base de données Postgres, appelée broken\_links. Une métadonnée peut avoir de 0 à n liens brisés donc plusieurs enregistrements dans la base de données. De plus, à chaque lien brisé un statut est assigné. Un changement de statut amène la création d’un nouvel enregistrement dans la base de données. Un enregistrement n’est jamais supprimé; on pourra alors extraire un historique par métadonnée et par métadonnée-lien brisé.

Une métadonnée en erreur aura autant d’entrées dans la table broken\_links\_issues qu’elle a de liens brisés. À chacune de ces combinaisons « Metadata\_UUID—URL », il y a 3 valeurs de statut possible et chaque valeur différente aura son propre enregistrement. Ainsi, chaque fois qu’une erreur change de statut, un nouvel enregistrement dans la table broken\_links\_issues est créé. De plus, une erreur corrigée peut être à nouveau en erreur, la date de détection de l’erreur est importante et sera utilisée pour certains traitements.

### Schéma de la base de données

**Table broken\_links\_issues**metadata\_uuid (PK) Char(36) Identifiant unique de la métadonnée en erreur  
url VarChar(),Null=No url du lien brisé  
resource\_format (FK) VarChar(), Null=Yes Nom du format de la ressource en erreur  
broken\_link\_problem VarChar(), Null=Yes Type d’erreur  
broken\_link\_status (FK) VarChar(), Null=No Statut de l’erreur  
broken\_link\_status\_date Date, Null=No Date de la détection de l’erreur (YYYY-MM-DD)  
juridiction VarChar(),Null=Yes Provenance de la métadonnée dans le catalogue date\_email Date, Null=Yes Statut du courriel

La clé unique est composée des 3 attributs suivants : metadata\_uuid + url + broken\_link\_status\_date

**Table cl\_broken\_link\_problem**problem (PK) VarChar(),Null=No Type d’erreur  
description\_en VarChar(),Null=Yes Description en anglais du type d’erreur  
description\_fr VarChar(),Null=Yes Description en français du type d’erreur

**Table cl\_resource\_format**format (PK) VarChar(),Null=No Nom du format de la ressource en erreur  
description\_en VarChar(),Null=Yes Description en anglais de la ressource  
description\_fr VarChar(),Null=Yes Description en français de la ressource

**Table cl\_broken\_link\_status**status (PK) VarChar(),Null=No Statut de l’erreur  
description\_en VarChar(),Null=yes Description en anglais du statut de l’erreur  
description\_fr VarChar(),Null=Yes Description en français du statut de l’erreur

### Exemple de contenu de la table broken\_links\_issues

Exemple du contenu de la table BROKEN\_LINKS   
METADATA\_UUID URL STATUS DATE\_DETECTION

MD1 url1 observé 2020-12-01  
MD2 url2 observé 2020-12-01  
MD3 url4 observé 2020-12-02  
MD1 url3 observé 2020-12-02  
MD2 url2 confirmé 2020-12-02  
MD3 url4 confirmé 2020-12-02  
MD1 url1 réglé 2020-12-02  
MD1 url1 observé 2020-12-03  
MD2 url2 réglé 2020-12-03  
MD1 url3 réglé 2020-12-03

Les changements de statut permis sont les suivants :  
 De Observé (OBSERVED) à Confirmé (CONFIRMED)  
 De Confirmé (CONFIRMED) à Réglé (FIXED)  
 De Observé (OBSERVED) à Réglé (FIXED)

## Traitements

### Lecture du catalogue

### Le catalogue de la PGF est lu au complet en utilisant le custom transformer Catalogue\_Reader\_6. Les paramètres à préciser sont les suivants : Provider : Laisser vide Environnement : Prod – externe Identifiant : Laisser vide CUSTOM\_URL : Laisser vide

### Détection des liens brisés

Tous les liens de chaque métadonnée lue sont vérifiés à l’aide du custom transformer BrokenLinks\_Detector. Ce transformeur contient une seule sortie, soit Broken\_links. Le résultat contient les valeurs qui seront utilisées pour ajouter un enregistrement dans la table broken\_links\_issues s’il y a lieu. Les valeurs sont les suivantes :  
 IN\_METADATA\_UUID Identifiant unique de la métadonnée en erreur  
 IN\_RESOURCE\_FORMAT Nom du format de la ressource en erreur  
 IN\_URL\_ERRONE Lien url de la ressource en erreur  
 IN\_TYPE\_ERREUR Type d’erreur  
 IN\_JURIDICTION Provenance de la métadonnée qui possède au moins un lien brisé

### Peupler la table broken\_links\_issues

Une métadonnée en erreur aura autant d’entrée dans la table broken\_links\_issues qu’elle a de liens brisés. À chacune de ces combinaisons « Metadata\_UUID—URL », il y a 3 valeurs de statut possible et chaque valeur différente aura son propre enregistrement. Ainsi, chaque fois qu’une erreur change de statut, un nouvel enregistrement dans la table broken\_links\_issues est créé. De plus, une erreur corrigée peut être à nouveau en erreur, la date de détection de l’erreur est importante et sera utilisée pour certains traitements.

Pour peupler la table broken\_links\_issues, le custom transformer PeuplerBrokenLinks.fmx est utilisé. Ce transformeur a 4 paramètres d’entrée et 4 paramètres de sortie  
 IN\_ERREUR Information fournie par le transformeur BrokenLinks\_Detector.fmx  
 IN\_RESOURCE\_FORMAT Table cl\_resource\_format  
 IN\_TYPE\_ERREUR Table cl\_broken\_link\_problem  
 IN\_BROKEN\_LINKS\_ISSUES Table broken\_links\_issues  
 OUT\_BROKEN\_LINKS\_ISSUES Table broken\_links\_issues  
 Unknown\_Format Contenu de IN\_ERREUR  
 Unknown\_Type\_Erreur Contenu de IN\_ERREUR  
 Unknown\_Statut Contenu de IN\_ERREUR + broken\_links\_issues.broken\_link\_status

Seuls les liens brisés dont la juridiction (i.e. IN\_JURIDICTION fournie par BrokenLinks\_Detector) égale la valeur du paramètre IN\_CHOICE\_JURIDICTION.

Avant de créer les enregistrements dans la broken\_links\_issues, le transformeur s’assure que   
 IN\_RESSOURCE\_FORMAT contient une valeur de la table cl\_resource\_format et   
 IN\_TYPE\_ERREUR contient une valeur de la table cl\_broken\_link\_problem.

Lorsque IN\_TYPE\_ERREUR n’est pas valide, l’information qui a été transmise par BrokenLinks\_Detector.fmx est retournée par le port Unknown\_Type\_Erreur et le message d’erreur « *ERREUR DE TYPE D'ERREUR : @Value(IN\_METADATA\_UUID), ERREUR : @Value(IN\_TYPE\_ERREUR) »* est écrit dans le fichier .log. Aucun enregistrement dans la table broken\_links\_issues ne peut être créé.

Lorsque IN\_RESOURCE\_FORMAT n’est pas valide, l’information qui a été transmise par BrokenLinks\_Detector.fmx est retournée dans le port Unknown\_Format et le message d’erreur « *ERREUR DE FORMAT : @Value(IN\_METADATA\_UUID), FORMAT : @Value(IN\_RESOURCE\_FORMAT)* » est écrit dans le fichier .log. De plus, comme le fichier .log est énorme, un fichier Excel est créé afin d’identifier plus facilement les métadonnées en erreur. À noter, qu’un enregistrement dans la table broken\_links\_issues pourra quand même être créé.

Lorsque l’attribut broken\_links\_issues.broken\_link\_status contient une valeur autre que OBSERVED, CONFIRMED ou FIXED, l’information fournie par Broken\_Links et la valeur de broken\_link\_status sont retournées par le port Unknown\_Statut. Le message ERREUR DE STATUT : @Value(IN\_METADATA\_UUID), STATUT : @Value(\_listMerge{0}.broken\_link\_status) est alors écrit dans le fichier .log.

Le transformeur PeuplerBrokenLinks détermine la valeur de l’attribut broken\_link\_status et assigne la date du jour à l’attribut broken\_link\_status\_date. Les valeurs possibles qu’il peut attribuer sont OBSERVED, CONFIRMED.

### Assigner la valeur FIXED à l’attribut broken\_link\_status\_date

### Les enregistrements de la table broken\_links\_issues peuvent avoir 3 valeurs de statuts, OBSERVED, CONFIRMED, FIXED. Les valeurs OBSEVED et CONFIRMED sont déterminées par le custom PeuplerBrkenLinks tandis que le custom PeuplerBrokenLinks\_FIXED assigne la valeur FIXED à l’attribut broke\_link\_status.

Lira la table broken\_links\_issues et ne conserver que les enregistrements dont broken\_links\_issues.juridiction = IN\_CHOICE\_JURIDICTION.

Tous les enregistrements à l’état OBSERVED et CONFIRMED pour lesquels le custom BrokenLinks\_Detector n’a détecté aucune erreur sont considérés réglés. Un enregistrement est alors créé avec l’attribut broken\_link\_status=FIXED et broken\_link\_status\_date = Date du jour.

## Liste des paramètres :

* IN\_CHOICE\_JURIDICTION Choix de valeurs permises : Fed, ProvTer,Tout  
  IN\_OUT\_POSTGRES\_CONNECTION Nom de la connexion à la BD, préalablement définie
* OUT\_XLS\_FORMAT Nom complet du fichier Excel Table pour inscrire les formats invalides
* LOG\_FILE Nom complet du fichier .log

## Intrants/extrants, pré/post-conditions, cas limites

Intrants/Extrants :  
 Base de données Postgres avec la table broken\_links\_issues

Intrants :   
 Base de données Postgres avec les tables   
 cl\_broken\_link\_problem  
 cl\_broken\_link\_status

Extrants :   
 OUT\_BROKEN\_LINKS\_ISSUES Enregistrement de lien brisé à écrire dans la table broken\_links\_issues  
 Unknown\_Format Valeurs des attributs de IN\_ERREUR  
 Unknown\_Type\_ERREUR Valeurs des attributs de IN\_ERREUR  
 Unknown\_Statut Valeurs des attributs de IN\_ERREUR

## Composantes du logiciel

Catalogue\_Reader Lecture du catalogue de la pGF  
BrokenLinks\_Detector Détection des liens brisés dans les enregistrements du catalogue de la PGF  
PeuplerBrokenLinks Enregistrement des méta données qui ont des liens brisés  
PeuplerBrokenLinks\_FIXED Enregistrement des liens brisés réglés

## Messages

ERREUR DE FORMAT : @Value(IN\_METADATA\_UUID), FORMAT : @Value(IN\_RESOURCE\_FORMAT)

ERREUR DE TYPE D'ERREUR : @Value(IN\_METADATA\_UUID), ERREUR : @Value(IN\_TYPE\_ERREUR)

ERREUR DE STATUT : @Value(IN\_METADATA\_UUID), STATUT : @Value(\_listMerge{0}.broken\_link\_status)