

# GETTREES V.0.2

© Eddy BRIERE – 2022



# Table des matières

<b>1</b>	<b>INTRODUCTION .....</b>	<b>6</b>
1.1	ORGANISATION DES FICHIERS ET DOSSIERS.....	6
1.1.1	<i>Exemple de l'arborescence .....</i>	<i>6</i>
1.2	L'ORGANISATION DES DOSSIERS « PROJET » .....	7
1.2.1	<i>Exemple de projet « LOL » .....</i>	<i>7</i>
1.3	AVANTAGE ET INCONVENIENT DE CETTE ARBORESCENCE.....	8
1.4	POURQUOI GETTREES ?.....	8
<b>2</b>	<b>GETTREES .....</b>	<b>9</b>
2.1	LE SCRIPT GETTREES.SH .....	9
2.1.1	<i>Authentification par clé.....</i>	<i>9</i>
2.1.2	<i>Le fichier de configuration.....</i>	<i>10</i>
2.1.3	<i>Le fichier LOG .....</i>	<i>10</i>
2.1.4	<i>Le PID et fichiers temporaires.....</i>	<i>11</i>
2.1.5	<i>Étapes réalisées par le script getTrees.sh.....</i>	<i>11</i>
2.2	LE SCRIPT TREES2XML (AWK) .....	11
2.2.1	<i>Le fichier xml .....</i>	<i>13</i>
2.2.2	<i>Lecture des TAGs XML .....</i>	<i>13</i>
2.2.3	<i>Note sur trees2xml .....</i>	<i>14</i>
2.3	LE SCRIPT INITDATABASE.SH .....	15
<b>3</b>	<b>LA BASE DE DONNEES.....</b>	<b>17</b>
3.1	LA TABLE CATALOG .....	17
3.2	LA TABLE KEYWORDS .....	17
3.3	LA TABLE LINK_KEYWORDS.....	17
3.4	LE DIAGRAMME DE LA BASE DE DONNEES PHOTOCATALOG .....	18





# 1 Introduction

GETTREES est un ensemble de scripts permettant de collecter les fichiers et dossiers qui regroupent des travaux photos de l'entreprise EDDY BRIERE pour en extraire les informations pertinentes et les sauvegarder dans une base de données.

Cette collecte permet de faciliter la recherche par mot clé et de produire facilement différents types de rapports.

## 1.1 Organisation des fichiers et dossiers

Les différents travaux photographiques du photographe E.B. sont organisés sur un disque de stockage localisé sur un serveur NAS.

La base du dossier se nomme « PHOTOS » et contient deux dossiers :

- STOCK (dossier contenant l'ensemble des photos brutes « les RAW »)
- WORK (dossier contenant l'ensemble des photos retravaillées)

Dans chacun de ces dossiers nous avons une arborescence similaire :

[ANNEE]/[CATEGORIE]/[CLIENT]/[PROJET]/...

### 1.1.1 Exemple de l'arborescence

Ici nous voyons l'organisation des fichiers pour le dossier STOCK/2012/



Les catégories disponibles pour cette année sont : Comédiens, Commandes, People et Perso.

Si nous prenons la catégorie « Commandes » nous pouvons voir mes clients suivants : Bad Réputations, Canal+, Fauchon, Fidelio, etc.

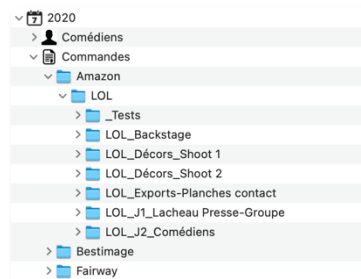
Enfin pour le client Fidélio, nous voyons deux projets : « L'Écume des jours » et « La cage Dorée »

## 1.2 L'organisation des dossiers « Projet »

Les dossiers « Projet » n'ont pas d'organisation particulière, néanmoins le nom des sous-dossiers et des fichiers contiennent des informations importantes.

### 1.2.1 Exemple de projet « LOL »

En 2020 nous avons réalisé le projet Photographique LOL pour Amazon :



Ci-dessus nous voyons bien les dossiers respectifs « Amazon » et « LOL ». Nous voyons aussi des sous-dossiers contenant le découpage photographique :

- LOL\_Backstage,
- LOL\_Décor\_Shoot 1,
- etc.

Chacun des mots de ces dossiers sont pour nous des mots clés permettant de faciliter la recherche de photo.

Un sous dossier peut aussi contenir d'autres sous-dossiers afin d'affiner l'organisation des images. Comme par exemple le sous-dossier LO\_J2\_Comédiens :



Cette organisation nous offre suffisamment d'information pour retrouver une ou des photos.

### 1.3 Avantage et inconvénient de cette arborescence

Cette organisation de dossiers a pour avantage de reposer sur une normalisation de fichiers compatible avec tous les systèmes d'exploitation et des systèmes de fichiers anciens, actuels et futurs.

Cette organisation ne dépend pas d'un logiciel propriétaire et donc garantie une pérennité.

En revanche, le volume des travaux devenant de plus en plus important il implique de posséder une excellente mémoire quand il s'agit de retrouver des photos. Dans le cas contraire il faudra parcourir à la main un ensemble de dossiers avant de retrouver une ou des photos.

### 1.4 Pourquoi GETTREES ?

Pour remédier aux inconvénients de cette arborescence est venue l'idée de concevoir un outil permettant d'inscrire les informations du disque « Photos » vers une base de données.

De cette manière nous pourrons facilement concevoir des outils de recherches et de production de rapports.



## 2 GETTREES

GETTREES est composé d'un ensemble de scripts :

```
.  
├── ./awk_script  
│   ├── ./awk_script/format_work_tree  
│   ├── ./getTrees.sh  
│   ├── ./log  
│   └── ./log/gettrees.log  
├── ./mysql_script  
│   └── ./mysql_script/initDatabase.sh
```

Dans les chapitres suivants nous allons décrire chacun d'entre eux.

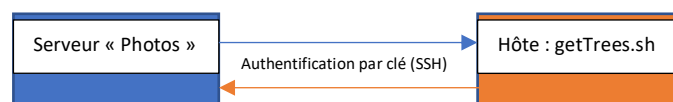
### 2.1 Le script getTrees.sh

Ce script est écrit en langage Shell et est donc prévu pour fonctionner sur des systèmes de type Unix (dans notre cas Linux Ubuntu Distribution).

La base de données utilisée est MySQL.

#### 2.1.1 Authentification par clé

Point important, l'accès à l'arborescence du serveur se fait par un mécanisme d'authentification par clé (SSH) entre le serveur et l'hôte qui exécute le script getTrees.sh.



Il est important de le mettre en place en amont pour que getTree.sh fonctionne. Je vous invite à vous documenter sur le sujet via l'adresse suivantes :

<https://doc.fedora-fr.org/wiki/SSH : Authentification par clé>

Cette authentification par clé nous permettra d'automatiser l'exécution du script getTrees.sh sans avoir besoin de fournir de mot de passe lors de la connexion au serveur « PHOTOS ».

### 2.1.2 Le fichier de configuration

Le script « gettrees.sh » est le script initial et interroge un fichier de configuration :

```
/etc/gettrees/gettrees_config
```

Le fichier de configuration « gettrees\_config » doit contenir les informations suivantes :

- GT\_SERVER -> serveur où se trouve le dossier « PHOTOS »
- GT\_SSH\_PORT -> port ssh de connexion au serveur
- GT\_PHOTOS\_DIR -> Chemin où se trouve le dossier « PHOTOS » sur le serveur
- GT\_USER -> Compte utilisation sur le serveur <sup>1</sup>
- DB\_USER -> Compte utilisateur sur la base de données
- DB\_SERVER -> nom ou adresse IP du serveur de base de données
- DB\_PASSWORD -> mot de passe de compte base de données
- DB\_NAME -> nom de la base de données

Voici un exemple de fichier de configuration pour le script gettrees.sh :

```
GT_SERVER=serenity.frogs.local
GT_SSH_PORT=
GT_PHOTOS_DIR=/volume2/Photos
GT_USER=eddy
DB_USER=eddy
DB_SERVER=cyberlab.frogs.local
DB_PASSWORD=
DB_NAME=PhotoCatalog
```

### 2.1.3 Le fichier LOG

Le fichier log/gettrees.log permet de contrôler le bon déroulement du script. Il contient les différents warning, étapes du script et au pire les erreurs d'exécution.

Exemple :

```
ven. 24 juin 2022 14:06:50 UTC -> Start Import from server serenity.frogs.local files from
/volume2/Photos/Work
ven. 24 juin 2022 14:07:06 UTC -> End Import from server serenity.frogs.local files from
/volume2/Photos/Work
ven. 24 juin 2022 14:07:06 UTC -> Start Import from server serenity.frogs.local files from
/volume2/Photos/Stock
ven. 24 juin 2022 14:10:55 UTC -> End Import from server serenity.frogs.local files from
/volume2/Photos/Stock
ven. 24 juin 2022 14:10:55 UTC -> Init database PhotoCatalog
mysql: [Warning] Using a password on the command line interface can be insecure.
mysql: [Warning] Using a password on the command line interface can be insecure.
mysql: [Warning] Using a password on the command line interface can be insecure.
ven. 24 juin 2022 14:10:55 UTC -> Init database PhotoCatalog finished
ven. 24 juin 2022 14:10:55 UTC -> End getTrees
```

---

<sup>1</sup> La connexion au serveur est réalisée par une authentification par clé. Si ce mécanisme n'est pas mis en place au préalable alors le script getTrees.sh ne pourra en aucun cas atteindre l'arborescence du serveur.

### 2.1.4 Le PID et fichiers temporaires

Le PID (Process identification) est un numéro fourni par le système lors de l'exécution du script `getTree.sh`. Il permet au script de marquer les fichiers temporaires qu'il crée.

Normalement ces fichiers sont supprimés une fois le script exécuté, mais en cas d'erreur vous pouvez consulter ces fichiers dans le dossier `/tmp` afin d'identifier le moment où le problème c'est produit.

Exemple de fichiers temporaire produit pour le pid 451382 :

```
-rw-rw-r-- 1 eddy eddy 3924491 juin 24 10:23 /tmp/gt_import_stock451382_formated.xml
-rw-rw-r-- 1 eddy eddy 46266033 juin 24 10:23 /tmp/gt_import_stock451382.txt
-rw-rw-r-- 1 eddy eddy 643562 juin 24 10:20 /tmp/gt_import_work451382_formated.xml
-rw-rw-r-- 1 eddy eddy 2354482 juin 24 10:20 /tmp/gt_import_work451382.txt
```

### 2.1.5 Étapes réalisées par le script `getTrees.sh`

- `getTrees.sh`
  - Connexion au serveur via ssh avec exécution de la commande 'ls'
    - Création d'un fichier texte temporaire contenant l'arborescence complète du dossier STOCK
    - Conversion du fichier (STOCK) temporaire en fichier XML ([trees2xml script](#))
    - Création d'un fichier texte temporaire contenant l'arborescence complète du dossier WORK
    - Conversion du fichier (WORK) temporaire en fichier XML ([trees2xml script](#))
  - Initialisation de la base de données ([initDatabase.sh script](#))
  - Écriture des données XML en Base de données
  - Suppression des fichiers temporaires
  - Fin

## 2.2 Le script `trees2xml` (AWK)

Lorsque la remontée des arborescences des dossiers STOCK et WORK via la commande 'ls' est terminée, nous obtenons 2 fichiers textes dans le dossier `/tmp` :

```
gt_import_stock[PID].txt
gt_import_work[PID].txt
```

**Note** : `[PID]` correspond à une valeur numérique unique

Contenu des fichiers temporaire se présente comme suit (extrait):

```
/volume2/Photos/Work:
total 0
drwxr-xr-x 1 gwen users 50 Oct 25 2019 2004
drwxr-xr-x 1 root root 94 Jul 18 2019 2005
drwxr-xr-x 1 root root 92 Nov 11 2019 2006
drwxr-xr-x 1 root root 118 Nov 23 2019 2007
drwxr-xr-x 1 root root 138 Nov 11 2019 2008
drwxr-xr-x 1 root root 138 Jul 18 2019 2009
drwxr-xr-x 1 root root 138 Jul 18 2019 2010
drwxr-xr-x 1 root root 124 Jul 18 2019 2011
drwxr-xr-x 1 root root 124 Nov 20 2019 2012
drwxr-xr-x 1 root root 124 Jul 18 2019 2013
drwxr-xr-x 1 root root 124 Jul 18 2019 2014
drwxr-xr-x 1 root root 124 Nov 21 2019 2015
drwxr-xr-x 1 root root 122 Nov 21 2019 2016
drwxr-xr-x 1 root root 78 Oct 27 2019 2017
drwxr-xr-x 1 root root 88 Jul 18 2019 2018
drwxr-xr-x 1 root root 80 Jul 18 2019 2019
drwxr-xr-x 1 eddy users 88 Jun 5 2020 2020
drwxr-xr-x 1 eddy users 100 Jul 13 2021 2021
drwxr-sr-x 1 eddy users 100 May 30 08:23 2022
-rwxr-xr-x 1 root root 0 Jul 18 2019 Icon

/volume2/Photos/Work/2004:
total 0
-rwxr-xr-x 1 gwen users 0 Oct 25 2019 Icon
drwxr-xr-x 1 gwen users 198 Nov 23 2019 Perso

/volume2/Photos/Work/2004/Perso:
total 1928
drwxr-xr-x 1 gwen users 174 Oct 31 2019 Deborah
drwxr-xr-x 1 gwen users 362 Oct 31 2019 Deborah Emily & Caroline
drwxr-xr-x 1 gwen users 136 May 28 2020 Malika (Paris)
drwxr-xr-x 1 gwen users 238 Oct 31 2019 Stéphanie Jacquet (gouttes)
-rwxr-xr-x 1 eddy users 1970591 Feb 15 2006 affiche.jpg
...
```

Le script trees2xml permet de convertir le contenu ci-dessus au format xml

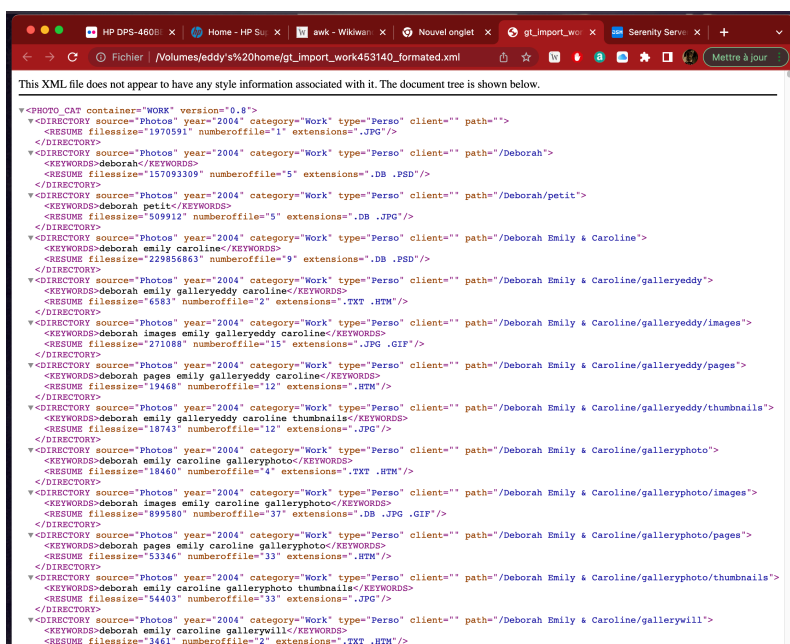
Extrait de sortie du script trees2xml :

```
<?xml version="1.0"?>
<PHOTO CAT container="WORK" version="0.8" >
  <DIRECTORY source="Photos" year="2004" category="Work" type="Perso" client="" path="" >
    <RESUME filesize="1970591" numberoffile="1" extensions=".JPG" ></RESUME>
  </DIRECTORY>
  <DIRECTORY source="Photos" year="2004" category="Work" type="Perso" client=""
path="/Deborah" >
    <KEYWORDS>deborah</KEYWORDS>
    <RESUME filesize="157093309" numberoffile="5" extensions=".DB .PSD" ></RESUME>
  </DIRECTORY>
  <DIRECTORY source="Photos" year="2004" category="Work" type="Perso" client=""
path="/Deborah/petit" >
    <KEYWORDS>deborah petit</KEYWORDS>
    <RESUME filesize="509912" numberoffile="5" extensions=".DB .JPG" ></RESUME>
  </DIRECTORY>
  <DIRECTORY source="Photos" year="2004" category="Work" type="Perso" client=""
path="/Deborah Emily & Caroline" >
    <KEYWORDS>deborah emily caroline</KEYWORDS>
    <RESUME filesize="229856863" numberoffile="9" extensions=".DB .PSD" ></RESUME>
  </DIRECTORY>
  <DIRECTORY source="Photos" year="2004" category="Work" type="Perso" client=""
path="/Deborah Emily & Caroline/galleryeddy" >
    <KEYWORDS>deborah emily galleryeddy caroline</KEYWORDS>
    <RESUME filesize="6583" numberoffile="2" extensions=".TXT .HTM" ></RESUME>
  </DIRECTORY>
  ...
```

### 2.2.1 Le fichier xml

Nous aurions pu inscrire directement l'arborescence des dossiers STOCK et WORK dans une base de données. Cependant il nous semblait plus judicieux de passer par un format standard.

Dans un premier temps la standardisation du XML, offre de multiples outils permettant de les éditer et donc de visualiser facilement nos données afin d'en vérifier la cohérence. Les éditeurs comme 'Sublime Text' ou encore un simple navigateur permet de colorer les TAG/MARQUEUR du langage XML.



Visualisation du fichier XML dans le navigateur Chrome

Si la cible finale est aujourd'hui une base de données, il est possible que demain nos données soient exploitées par d'autres outils. Nous pensons que le format XML reste un format judicieux pour une éventuelle migration vers un autre outil.

### 2.2.2 Lecture des TAGs XML

Voyons dans cette partie la description des différents TAGs (communément appelé **élément** en XML) dans notre fichier XML et leurs attributs respectifs.

Il est évidemment important de respecter l'agencement des éléments dans le fichier XML. :

- PHOTO\_CAT
  - DIRECTORY
    - KEYWORDS
    - RESUME

#### 2.2.2.1 PHOTO\_CAT

PHOTO_CAT	
Container	L'arborescence contenu dans le fichier XML exemple : WORK
Version	Version de getTrees.sh qui l'a produite

#### 2.2.2.2 DIRECTORY

DIRECTORY	
Source	Dossier de base de l'arborescence (toujours Photos dans notre cas)
Year	L'année de production des photos
Category	Source du dossier (WORK ou STOCK)
Type	Type de projet : Perso, Comédiens, People, Commandes ...
Client	Nom du client
Path	Chemin dans l'arborescence du serveur

#### 2.2.2.3 KEYWORDS

Ce TAG liste l'ensemble des mots clé à inscrire dans la base de données pour les recherches ultérieur via l'interface Web.

Chaque mot clé est séparé par un caractère espace :

```
<KEYWORDS>deborah images emily caroline galleryphoto</KEYWORDS>
```

#### 2.2.2.4 RESUME

RESUME	
Filesize	Somme de la taille des fichiers dans le dossier (en octets)
Numberoffile	Nombre de fichiers dans le dossier
Extensions	Extension trouvées dans le dossier

#### 2.2.3 Note sur trees2xml

Trees2xml est écrit en Awk, ce langage facilite le formatage de texte. Nous vous invitons à découvrir Awk via le lien suivant afin de mieux appréhender ce script :

<https://www.wikiwand.com/fr/Awk>

### 2.3 Le script initDatabase.sh

Ce script appelé par getTrees.sh permet de créer la base de données (**PhotoCatalog**) ainsi que les tables (**keywords**, **catalog** et **link\_keywrods**). Ce script est écrit en Shell et utilise la commande 'mysql'. Si la base

de données et les tables sont déjà créées alors le script rend la main au script getTrees.sh.

**Notes :** *L'utilisateur de la base de données MySql et les droits doivent être créer en amont. Dans le cas contraire, le script sera dans l'impossibilité de créer la base de données.*

### 2.4 Le Script updateDB

Le script updateDB met la base de données à jour.

Ce script n'existe pas encore...






### 3 La base de données


La base de données **PhotoCatalog** est constituée de 3 tables :

- Catalog
- Keywords
- Link\_keywords

#### 3.1 La table catalog

TABLE CATALOG	
Id 	Clé primaire
Year	Année de production des photos
Category	Catégorie : WORK ou STOCK
Type	Typé de projet : Perso, People, Comédiens, Commandes, etc.
Client	Nom du client
Path	Chemin du dossier
Totalsize	Poids des fichiers dans le dossier (en octets)
Totalfiles	Nombre total de fichiers dans le dossier
Warning	Exemple : l'enregistrement précédant différent (ex : taille de fichiers)
Extensions	Extensions de fichiers présent dans le dossier (ex : TIFF, PSD, etc.)
Date	Date de l'enregistrement de la ligne

#### 3.2 La table keywords

TABLE KEYWORDS	
Id 	Clé primaire
Keyword	Mot clé : exemple « Martin »

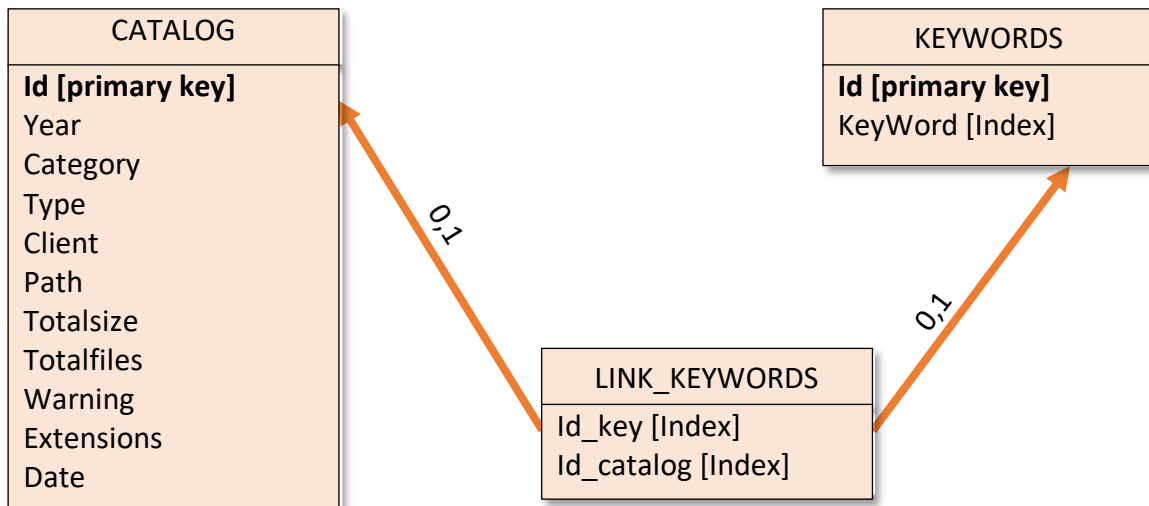
#### 3.3 La table link\_keywords

TABLE LINK_KEYWORDS	
Id_key	L'id de la table KEYWORDS
Id_catalog	L'id de la table CATALOG

### 3.4 Le diagramme de la base de données PhotoCatalog

L'ensemble des attributs du TAG (élément) DIRECTORY et RESUME sont inscrit dans la table CATALOG. En revanche les mots clé sont inscrit dans la table sont uniques.

C'est donc la table LINK\_KEYWORDS qui va créer les liens entre les mots clés et les dossiers.



Exemple avec une ligne **n** de la table CATALOG contenant les mots clés suivants :

- DEMAIN
- EST
- A
- VOUS
- JEAN
- RICHARD

Exemple d'extrait des tables LINK\_KEYWORDS & KEYWORDS

LINK_KEYWORDS	
3	<b>n</b>
7	<b>n</b>
43	<b>n</b>
23	<b>n</b>
45	<b>n</b>
17	<b>n</b>
7	m
18	m
...	...

KEYWORD	
3	A
7	EST
17	DEMAIN
43	VOUS
23	JEAN
45	RICHARD
46	LA
50	CHEMIN
...	...