Documentation technique bulletin Pasicrisie

Ce document a pour but d’expliciter le fonctionnement du logiciel de recherche dans les bulletins de Pasicrisie, disponible à l’adresse bulletin.pasicrisie.lu

# Côté client

L’interface est une application angular2 qui peut être servie sous forme de fichiers statiques une fois assemblée. Elle utilise le framework ionic afin de s’afficher d’une manière conventiionnelle sur des agents mobiles.

Afin de modifier cette application, il sera nécessaire d’en télécharger le code source depuis git (jusqu’à présent, sur un git externe), d’installer les dépendances (npm i), puis de procéder aux ajustements nécessaires.

Afin de déployer, la commande (ionic build) sera utilisée, ce qui génèrera les fichiers à servir dans le répertoire www. Ce contenu peut être placé à disposition d’un serveur virtuel apache ou nginx comme c’est le cas actuellement.

Le serveur actuel est bulletin.pasicrisie.lu (user : pasicrisie\_adm\_bulletin\_ftp, pass : iHftPM@92) en FTP, la racine du site virtuel est web.

# Côté serveur

Le serveur est entièrement virtualisé dans le cloud d’AWS. Il est composé de trois fonctions lambda, qui représentent respectivement :

* La transformation des RTF en RTF liés et PDF en détectant un upload de fichier dans S3 ;
* La validation manuelle d’un fichier pour le rendre trouvable ou non par la recherche, et modifier certaines de ses propriétés qui auraient été mal indexées ;
* L’API que le côté client utilise, afin de lister les tomes disponibles puis d’effectuer une recherche.

Ces fonctions lambdas sont cachées devant une API Gateway fournie par AWS afin d’avoir une URL externe. Cependant, le routing est fait par la fonction car très simple.

Ses données sont stockées dans une table DynamoDB, et les fichiers sont stockés dans S3. L’entièreté des exécutions peut être surveillée via les logs récoltés dans Cloudwatch. La gestion des utilisateurs se fait via Cognito, et les groupes permettent de créer des rôles en fonction des abonnements des utilisateurs.

Le code source des fonctions lambda est également versionné dans git. Afin d’en déployer une nouvelle version, s’assurer que le package uploadé dans AWS comprend les librairies associées collectées dans node\_modules par npm.

Les fonctions lambda devront également avoir accès à S3/DynamoDB, ce qui est réalisé par un rôle d’exécution.

Le compte AWS actuellement utilisé est (user : [sys@ic.lu](mailto:sys@ic.lu), pass : S6DN3TpE-uo- )

# Opérations techniques

## Import de bulletins (nouveaux PDFs)

Il suffit de se connecter à la console AWS, et de importer les PDFs ici <https://s3.console.aws.amazon.com/s3/buckets/pasicrisie-pdf?region=eu-central-1&prefix=bulletin/&showversions=false> (en mode lecture privée)

## Import de nouveaux chapitres (RTFs)

Il faut se connecter à la console AWS et importer les fichiers ici <https://s3.console.aws.amazon.com/s3/buckets/pasicrisie-raw-rtf?region=eu-central-1&prefix=bulletin/&showversions=false> (en mode lecture privée), cela lance le parsing automatisé et l’index puis renvoie les fichiers dans <https://s3.console.aws.amazon.com/s3/buckets/pasicrisie-linked-rtf?region=eu-central-1&prefix=bulletin/&showversions=false>

Il faut ensuite les télécharger, et les transformer en PDFs avec la macro word suivante (trouvable parmi les codes sources, également), pour le lancer : https://www.datanumen.com/blogs/how-to-run-vba-code-in-your-word/

|  |
| --- |
| Sub BatchConvertDocxToPDF()  Dim objDoc As Document  Dim strFile As String, strFolder As String    'Initialization  strFolder = "C:\Users\gma09\Desktop\linked\"  strFile = Dir(strFolder & "\*.rtf", vbNormal)    'Precess each file in the file folder and convert them to pdf.  While strFile <> ""  Set objDoc = Documents.Open(FileName:=strFolder & strFile)    objDoc.ExportAsFixedFormat \_  OutputFileName:=Replace(objDoc.FullName, ".rtf", ".pdf"), \_  ExportFormat:=wdExportFormatPDF, OpenAfterExport:=False, OptimizeFor:=wdExportOptimizeForPrint, \_  Range:=wdExportAllDocument, Item:=wdExportDocumentContent    objDoc.Close  strFile = Dir()  Wend  End Sub |

Il faut ensuite télécharger les fichiers PDFs générés dans, comme pour les bulletins, <https://s3.console.aws.amazon.com/s3/buckets/pasicrisie-pdf?region=eu-central-1&prefix=bulletin/&showversions=false> (en mode lecture privée)

Parfois, le parsing automatisé génère un RTF non lisible par word, c’est souvent le cas pour « étrangers » et « transport ». Il faut alors lancer le parsing en mode local avec le fichier rtf-linker.js et sauter manuellement les points problématiques.

La commande est

|  |
| --- |
| * node rtf-linker.js C:\Path\to\rtf.rtf |

Le but étant de supprimer le parsing haut niveau en commentant les lignes 52 à 92. Un fichier rtf-linked.rtf est généré et peut être transformé en PDF pour continuer l’opération.