

Cahier de recette

Rédigé par :

Nicolas Broute

Killian Buhr

Noémie Deloeuvre

Projet tableau de bord
Année universitaire 2017 - 2018

Versions du cahier de recette

Date	Version	Parties modifiées	Cause
26 / 03 / 2018	V 1.0	Toutes	Début du cahier de recette

Sommaire

Historique des révisions

Sommaire

1. Introduction

1.1. Objectif

1.2. Portée

2. Fonctions

2.1. Extraction

2.1.1. Utils

2.1.2. Crawler

2.2. Nettoyage

2.3. Insertion

2.4. Visualisation

2.4.1. Dashboard

2.4.2. Maps

1. Introduction

1.1. Objectif

Ce document présente la procédure à utiliser pour valider et recetter l'application OWEP. Il décrit, de manière détaillée, pas à pas, un ensemble de scénarios de tests regroupés par exigences. Ce document sert de support aux superviseurs pour évaluer si le produit répond à leurs attentes.

1.2. Portée

Le plan de recette est destiné aux superviseurs et aux clients du projet ainsi qu'aux membres de l'équipe.

2. Fonctions

2.1. Extraction

2.1.1. Utils

Nom fonction	utils_v40.py
Objectif	Généraliser des processus pour tous les crawler
librairie requise	csv datetime json os bs4 requests hashlib unicodecode re
Initialisation	Chemin où l'on veut créer le fichier json Liste des articles Nom du journal Abréviation du journal Titre de l'article Auteurs de l'article Date de publication de l'article Thème de l'article Lien url
Sortie	Fichier index (pour vérifier que l'article n'a pas déjà été récupéré) Fichiers json Page url étudiable Article en format json

2.1.2. Crawler

Nom fonction	crawler_NOMJOURNAL.py
Objectif	Récupérer tous les articles qui nous intéressent d'un journal
librairie requise	utils_v40.py re datetime
Initialisation	Lien url où l'on va trouver les articles Chemin où l'on veut créer les fichiers json
Sortie	Un fichier json pour chaque article

2.2. Nettoyage

Nom fonction	clean.py
Objectif	Nettoyer les articles pour ne garder que les informations importantes et en extraire des métadonnées. Il en ressort des fichiers csv correspondant aux tables de notre base de données
Librairies requises	json pickle unidecode csv spacy re datetime numpy sklearn
Initialisation	lien fichier "stopwords.p" lien des fichiers des journaux liste des journaux liste du nombre d'articles par journal liste de la syntaxe avant le numéro de l'article (généré dans le crawler) liste de la syntaxe après le numéro de l'article (généré dans le crawler)
Sortie	5 fichiers csv relatifs aux tables de la BD : <ul style="list-style-type: none"> - articles.csv - contenus_mots_cles.csv - titres.csv - nombres.csv - mots.csv

2.3. Insertion

Nom fonction	insertion_bd_NOMTABLE.py
Objectif	Insérer les données dans la table qui correspond
Librairies requises	csv pyodbc
Initialisation	Lien fichier csv Connexion à la base de données sur SQL Server
Sortie	Les données sont insérées dans la table

2.4. Visualisation

2.4.1. Dashboard

Nom fonction	Cloud.py
Objectif	Créer des nuages de mot, de pays
librairie requise	matplotlib wordcloud
Initialisation	un fichier dont la première colonne contient des pays ou bien des mots et dans la deuxième colonne leurs poids
Sortie	un graphique avec les 10 pays ou mots les plus importants

2.4.2. Maps

Nom fonction	maps.py
Objectif	Créer des cartes pour représenter les pays qui sont les plus cités dans les journaux de notre corpus et voir ceux qui sont les plus meurtriers
librairie requise	os folium csv
Initialisation	liste des différentes cartes que l'on souhaite (toujours avec le format 'carte_XXXX') liste des coefficients de multiplications lien où écrire les cartes
Sortie	7 fichiers html représentant chacun une carte avec une catastrophe bien précise ou bien avec les catastrophes naturelles en générales.