Ce document est un guide concernant l'indexation des personnages fictifs et des personnes réelles de la Comédie Humaine. Il explique comment l'indexation est effectuée et comment la poursuivre, la personnaliser ou en étendre les possibilités sans pré-requis en informatique, à partir de logiciels spécifiquement codés pour cela.

Ce rapport est disponible en ligne et en français :
https://github.com/Armellei/IndexBalzac

Rapport sur l'indexation
des personnages fictifs & des personnes réelles
de la Comédie humaine

Préambule

1.1 Présentation des choix d'indexation
1.2. Semi-automatisation du travail à l'aide de scripts
1.3 Logiciels requis pour démarrer l'indexation

Indexation

1.1 Workflow général
    1.1.1 Principes du workflow
    1.1.2 Procédure d'indexation par défaut
    1.1.3 Procédures d'indexation personnalisées et possibilités d'ouverture
1.2 Règles d'indexation
    1.2.1 Parti-pris de la Pléiade
    1.2.2 Attribution d'un identifiant unique
    1.2.3 Vedette et alias
    1.2.4 Divisions & subdivisions de l'index
    1.2.5 Descriptif des personnages et établissement de critères discriminants
    1.2.6 Nomenclature et syntaxe des bases de données
    1.2.7 Sensibilité à la casse
        1.2.7.1 Termes répétés pouvant comporter une majuscule
        1.2.7.2 Intégration d'exceptions à la sensibilité à la casse
        1.2.7.3 Interdictions
    1.2.8 Règles orthotypographiques
    1.2.9 Noms indexés et non indexés
1.3 Présentation des scripts
    1.3.1 Dutocq
    1.3.2 Rabourdin
    1.3.3 Colleville
    1.3.4 Bixiou
    1.3.6 txt2json
1.3 Repérage des doublons
    1.3.1 Repérages sur l'index en cours de construction
    1.3.2 Lancement du script de détection des doublons
1.4 Suppression des doublons
    1.4.1 Principe général
    1.4.2 Procédure
1.5 Balisage automatique en XML-TEI
    1.5.1 Balisage par défaut
    1.5.2 Balisage personnalisé
        1.5.2.1 Possibilités d'affinage
        1.5.1.2 Modification personnalisée des scripts python pas à pas
    1.5.3 Affichage HTML des noms balisés en XML-TEI

Génération de l'index pas à pas

1.1 Préparation des fichiers pour l'indexation
1.2 Écriture des bases de données
1.3 Lancement des scripts générant l'index
1.4 Vérifications

Iconographie

1.1 Traitement des gravures, icônes et blasons
1.2 Recommandations de retouche

Recommandations

1.1 Recommandations générales

Erreurs fréquentes lors de l'indexation & débuggage des scripts

1.1 Lecture des indications de débuggage données par les scripts
    1. Fonctionnement du débuggage et typologie d'erreurs
    2. Exemples
1.2 Erreurs sur les personnages
    1.1.1 Suppression des doublons
    1.1.2 Suppression de données différentes sur un même personnage
    1.2.3 Erreurs de remplissage des champs en txt
1.3 Erreurs de syntaxe
    1.2 .1 Erreurs de syntaxe des fichiers txt et JSON
    1.2.2 Erreurs d'encodage en UTF-8 et gestion des glyphes spéciaux
1.4 Erreurs multiples

Mise en ligne des fichiers

1.1 Mise à jour des bases de données depuis /IndexBalzac sur GitHub
1.2 Préparation des fichiers pour intégration web

Mise à disposition du code

Préambule

1.1 Présentation des choix d'indexation

L'indexation consiste en la création de bases de données référençant les noms des personnages fictifs et ceux des personnes réelles déployés par Balzac dans chacun des livres de la Comédie humaine. À chaque ouvrage sont donc systématiquement associées deux bases, distinguant les deux types de personnes qu'on y rencontre (réelles dans une base, fictives dans l'autre). Ces bases de données sont de simples fichiers texte faciles à manipuler, que l'on convertira automatiquement à la fin du travail en fichiers JSON.

Chaque livre de la Comédie humaine existe à la fois dans une version XML-TEI, disponible sur GitHub, et dans une version HTML, disponible en ligne sur le site eBalzac, version exportée à partir du fichier XML. Les bases de données sont liées à ces deux formats de fichier, avec lesquels elles communiquent.

Le travail d'indexation des personnages au sein des bases s'effectue « manuellement » par un opérateur qui lit chaque livre un à un en s'aidant de scripts pour accélérer le travail. Les opérations de contrôle (erreurs, cas particuliers) sont font semi-automatiquement. Pour chaque ouvrage, on créé deux bases de données que l'on complète. Ensuite, on peut contrôler automatiquement les erreurs éventuelles dans ces dernières (doublons, etc.), puis on génère un index qui prend la forme d'un fichier HTML permettant un deuxième contrôle rapide. Cet index est un fichier de travail destiné à la personne qui indexe, et qui reflète les avancées de son indexation : à chaque nouveau livre traité, l'index se met à jour. Ensuite, une fois l'index de travail mis à jour, on lance un balisage automatique du fichier XML-TEI de l'ouvrage pour y repérer les noms des personnages préalablement inscrits dans les deux bases concernées. Ce balisage permettra, entre autres, d'effectuer des recherches à partir de l'index numérique dans les textes, de mettre à disposition des outils d'exploitation des données balisées et de proposer un affichage dynamique d'informations extraites de l'index ou des livres dans leur version HTML . Enfin, on convertit automatiquement les bases de données texte en JSON avant de déposer ces dernières sur le site internet qui héberge l'index.

L'indexation de noms propres à l'aide de logiciels spécifiquement conçus pour cela (TXM, etc.) présentant un taux d'erreur important, la procédure d'indexation mise en place ici doit donc être semi-automatisée et a été conçue sur mesures pour s'adapter à l'écriture de Balzac.

En moyenne, un à deux livres par jour peuvent être indexés, selon la vitesse de lecture de l'opérateur.

1.2. Semi-automatisation du travail à l'aide de scripts

Cinq scripts, détaillés dans la section 1.3, servent à automatiser une partie du travail : ce sont de petits logiciels, dont quatre sont codés en python, et un en awk. Il est possible de travailler à l'indexation sans faire usage de tous ces scripts ou sans avoir besoin de les comprendre : il suffit

simplement de cliquer dessus et ils font leur travail tout seul.

Les scripts se lancent un par un à chaque étape de l'indexation : ils n'ont volontairement pas été fusionnés en un seul logiciel afin que l'opérateur soit libre d'intervenir script par script s'il a besoin de les modifier ou de chercher une erreur précisément dans l'un des logiciels. Toutefois, le code étant librement publié en ligne, on pourra toujours générer si nécessaire un unique fichier exécutable regroupant les 4 scripts codés en python ainsi que le script awk.

1.3 Logiciels requis pour démarrer l'indexation

Deux logiciels, dont la prise en main est simple, et deux langages permettant d'exécuter les scripts ou de les modifier devront être installés sur le système.

- un **éditeur de texte** de type Notepad++ ;
L'éditeur doit gérer l'affichage des caractères en UTF-8.

- un **terminal** ;
Un terminal est un logiciel d'interprétation de commandes en mode texte.

Sous Mac : Terminal
Sous Windows : Ne pas utiliser l'invite de commande Windows (cmd), télécharger plutôt Git Bash > https://gitforwindows.org/
Sous Linux : Terminal, gnome-terminal, konsole, xterm...

- **Python** ;
Python est un langage permettant de lire et d'exécuter des scripts qui automatiseront ici le travail d'indexation.

Installer Python > https://www.python.org/download/

- **Awk**
Awk est un langage permettant de traiter des données textuelles qui automatiseront ici le travail d'indexation. NB : Awk est directement disponible après avoir installé GitBash.

Sous Mac : Installer Homebrew, ouvrir in terminal, puis taper :

```
brew install gawk
```

Sous Windows : installer GitBash
Sous Linux : sous distrib compatible avec Debian (Ubuntu...) ouvrir un Terminal, puis taper :

```
sudo apt-get update
sudo apt-get install gawk
```

Sinon, installer gawk depuis le gestionnaire de paquets.

Indexation

1.1 Workflow général

1.1.1 Principes du workflow

Le workflow adopté pour indexer est le suivant :

- Constitution « manuelle » des bases de donées sous la forme de fichiers texte
- Débalisage manuel éventuel du fichier XML-TEI à l'aide d'un fichier de débalisage
- Débuggage éventuel des bases de données à l'aide d'un script python : Dutocq
- Génération automatique d'un index en HTML à l'aide de Dutocq
- Vérification des doublons à l'aide d'un script python : Bixiou
- Balisage automatique des ouvrages en XML-TEI à l'aide d'un script python : Rabourdin
- Transformation des bases de données texte en fichiers JSON à l'aide d'un script awk : txt2json

- Dépôt des bases de données JSON sur le site eBalzac

1.1.2 Procédure d'indexation par défaut

La procédure d'indexation consiste à référencer les personnages ainsi qu'un ensemble de critères les caractérisant et les discriminant les uns par rapport aux autres. Ces informations sont contenues au sein des bases de données, qui sont liées aux fichiers XML-TEI par un balisage volontairement réduit au strict minimum : des balises <persName> augmentées d'un identifiant unique permettant de préciser qui est référencé suffisent. Ces identifiants permettent de repérer chacun des personnages au sein de l'index, fichier XML-TEI, des bases de données et des bases exportées en JSON. Cet ID unique permet d'effectuer des renvois entre ces différents fichiers.

Ce balisage évite :
- d'alourdir inutilement les fichiers XML-TEI, déjà très conséquents, en interrogeant à l'affichage de l'index uniquement les bases de données légères en JSON qui, elles, peuvent contenir un grand nombre d'informations relatives aux personnages : les lieux qu'ils fréquentent, leur réseau relationnel, des informations personnelles, etc. ;
- des erreurs de balisage concernant tout autre information (lieux...) que le nom des personnages et que l'on voudrait baliser aussi, ces erreurs pouvant être pénibles à déceler au sein d'un lourd fichier XML-TEI. Leur correction dans une base de données textuelle est en effet bien plus aisée.

Enfin, le balisage en XML-TEI se fait automatiquement, en lançant un script qui interroge les bases de données, en relève les erreurs de syntaxe ou les doublons, et balise ensuite tout seul le fichier si tous les signaux sont au vert.

1.1.3 Procédures d'indexation personnalisées et possibilités d'ouverture

Il est possible de baliser automatiquement :

- plus finement les <persName> déjà existants à l'aide d'attributs ;
- toute autre chaîne de caractères à l'aide de n'importe quelle balise TEI.

Pour modifier la balise <persName> ou en ajouter d'autres, il faut modifier le script Rabourdin.py comme suit :

- Ouvrir Rabourdin.py dans un éditeur de texte de type Notepad++
- Modifier la ligne 7 en changeant le nom de la balise ou en en ajoutant d'autres au besoin
- Modifier en conséquence les lignes 42, 43, 68, 69 en se référant aux commentaires du script

Si d'autres informations sont balisées (par exemple, des lieux), on peut créer de nouvelles bases de données référençant tous les lieux avant d'automatiser le balisage des fichiers, à l'instar du modèle proposé par les personnages.

1.2 Règles d'indexation

1.2.1 Parti-pris de la Pléiade

En règle générale, les règles d'indexation suivent celles de l'index des personnages fictifs et des personnes réelles de la Pléiade, tome XII. En cas de doute, s'y référer. Elle renseignera par exemple sur l'orthographe d'un nom qui doit être privilégiée si Balzac propose des graphies différentes, sur l'état civil parfois trouble des personnages, ou encore sur un certain nombre d'erreurs de l'auteur (homonymes, personnes réelles et fictives à la fois, enfants posthumes, etc.).

1.2.2 Attribution d'un identifiant unique

Chaque personne possède un identifiant unique (ID) permettant de la discriminer. Cet identifiant est, pour les personnages fictifs, un numéro unique suivi de la lettre P, et pour les personnages réels, un numéro unique suivi de la lettre R.

Exemple :

Modeste Mignon est identifiée comme : 181P
Aspasie est identifiée comme : 200R

1.2.3 Vedette et alias

Vedette

- Chaque personnage est connu sous un nom principal - la vedette - et d'autres formes du même nom, nommées alias.

- Les personnes possédant plusieurs identités apparaissent dans l'index sous leur identité la plus répandue (vedette), les autres identités étant ajoutées en alias.

Alias

Toutes les autres formes du nom d'un personnage n'étant pas la forme principale (vedette) doivent impérativement être indexés telles qu'elles apparaissent dans les écrits de Balzac. Sont donc considérés comme des alias devant être indexés sous la forme exacte relevée dans le texte :

- Le ou les prénoms ;
- Le nom de jeune fille des femmes mariées ;
- Les patronymes successifs des personnages remariés ;
- Les sobriquets ;
- Les surnoms ;
- Les noms de plume ;
- Les noms de guerre ;
- Les personnes désignées par un titre ;
- Toute autre forme de nom désignant explicitement un personnage.

→ Si un personnage n'apparaît jamais sous son prénom, le prénom ne sera donc pas indexé dans les bases de données.

1.2.4 Divisions & subdivisions de l'index

L'index est divisé en deux parties principales, qui pourront aussi être affichées simultanément :

personnages fictifs / personnes réelles

L'index est ensuite subdivisé par :

catégorie sociale / métier / lieu de naissance ou de vie / sexe / réseau relationnel (familial, amical, amoureux, politique, professionnel, etc.) / caractère reparaissant ou non / tout autre critère discriminant présent dans la description qu'on fera au sein des bases de données

1.2.5 Descriptif des personnages et établissement de critères discriminants

Chaque personnage peut faire l'objet d'une description dont la taille n'est pas limitée. Par défaut, les personnes réelles sont définies succinctement, mais les personnages fictifs peuvent être présentés de manière plus libre, en ajoutant des informations qui ne sont pas présentes dans la Pléiade et qui peuvent permettre d'établir une typologie de personnages plutôt qu'un déroulé des grandes étapes de leur existence, comme le propose l'ouvrage de référence.

1.2.6 Nomenclature et syntaxe des bases de données

Chaque ouvrage comporte deux bases de données textuelles ainsi qu'un fichier tiers nommé « fichier de débalisage » dont la structuration, la syntaxe et la nomenclatures, extrêmement simples mais immuables, sont régies par les règles suivantes :

Nomenclature

- La base de données dédiée aux personnages fictifs est un fichier texte portant le nom de l'ouvrage suivi de l'extension « _P.txt » ;

- La base de données dédiée aux personnes réelles est un fichier texte portant le nom de l'ouvrage suivi de l'extension « _R.txt » ;

- Le fichier de débalisage est utilisé librement par l'opérateur qui indexe ; il n'est régi par aucune syntaxe mais porte impérativement le nom de l'ouvrage suivi de l'exension « _DEBUG.txt ».

Syntaxe

- Chaque personnage fictif ou personne réelle est contenu dans un paragraphe au sein du fichier texte constituant la base de données à laquelle il appartient. Chaque groupe d'informations propre à un même personnage fictif ou à une même personne réelle occupe une ligne de texte.

- Chaque personnage fictif ou personne réelle est contenu au minimum sur une première ligne, obligatoire, constituant un premier groupe d'informations, mais il peut avoir ensuite autant de lignes facultatives que l'on voudra si d'autres informations sont ajoutées.

- Chaque ligne de texte débute et se termine par un caractère alphanumérique. Aucune espace ne doit être laissée en fin ou en début de ligne, en fin ou en début de fichier.

- Le premier groupe d'informations, obligatoire, occupant la première ligne du paragraphe dédié à un personnage, est constitué des 6 informations suivantes dont 2 seulement peuvent ne pas être renseignées. Elles sont séparées par 5 chevrons :

ID unique>Vedette>Descriptif>Lieu>Genre(0/1/2/3)>Lettre pour indexation

- Le deuxième groupe d'informations, facultatif, ocuppant une deuxième ligne dans le fichier texte, est constitué d'un @ décrivant les relations du personnage, suivi du signe : puis de l'ID du ou des personnages partageant la relation. Chaque type de relation identique occupe une ligne et peut comporter plusieurs ID séparés au besoin par des virgules.

- Le troisième groupe d'informations, facultatif, est constitué du ou des alias du personnage. Chaque alias occupe une ligne.

- Les 5 chevrons sont obligatoires.

- À l'exception des troisième et quatrième champs d'informations qui peuvent être vides, toutes les informations doivent être renseignées.

- La structure générale de chaque paragraphe est donc la suivante :

ID unique>Vedette>Descriptif>Lieu>Genre(0/1/2/3)>Lettre pour indexation
@relations
Alias

→ Il peut y avoir plusieurs relations, et plusieurs alias, soit une structure comme suit :

ID unique>Vedette>Descriptif>Lieu>Genre(0/1/2/3)>Lettre pour indexation
@relations : ID1
@relations : ID2, ID3
@relations : ID4
Alias 1
Alias 2
Alias 3

1.2.7 Règles orthotypographiques

Lettre d'entrée dans l'index

Chaque nom possède une lettre d'entrée dans l'index, celle que l'on cherche pour le retrouver. Cette lettre est choisie par l'opérateur qui indexe, elle n'est pas générée automatiquement à cause de la complexité de certains patronymes.

Si la personne possède un patronyme avec une particule, le choix de la lettre d'entrée dans l'index suit les exemples suivants :

Nom	Inscription du nom dans l'index	Lettre d'entrée dans l'index
Machin Chouette	CHOUETTE (Machin)	C
Machin de Chouette	CHOUETTE (Machin de)	C
Machin des Chouettes	CHOUETTES (Machin des)	C
Machin Du Truc	DU TRUC (Machin)	D
Machin de La Chouette	LA CHOUETTE (Machin de)	L
Machin Le Truc	LE TRUC (Machin)	L
Machin La Chouette	LA CHOUETTE (Machin)	L
Machin de L'Houette	L'HOUETTE (Machin de)	L
Machin d'Houette	HOUETTE (Machin d')	H
Machin Chouette de Saint-Truc	CHOUETTE DE SAINT-TRUC (Machin)	C

Exemple :

Jeanne de La Peyrade des Canquoëlles sera indexée:
LA PEYRADE DES CANQUOËLLES (Jeanne de).

Usage des petites capitales

Les patronymes sont composés en petites capitales, suivis du ou des prénoms et/ou titres en bas de casse. Les particules *de* et *d'* et *des* ne prennent pas de petite capitale et seront composées en bas de casse.

Les particules étrangères suivent les règles suivantes :

Particule	Exemple de nom	Composition
di	PIOMBO (Bartholoméo di)	bas de casse
van	GOBSECK (Esther van)	bas de casse
von	WITTGENSTEIN (Peter-Ludwig von)	bas de casse

Patronymes et métiers

L'entrée dans l'index se fait par le patronyme, ou à défaut par le métier quand Balzac ne donne pas d'indication patronymique, comme c'est souvent le cas pour les domestiques, les personnages important peu ou ceux les moins fortunés.

Cas particuliers : les diplomates (indexés par métier et non par origine), les domestiques ou valets désignés par leur métier uniquement (indexés par métier suivi d'une caractéristique mise entre parenthèse, si possible, afin de différencier les innombrables valets ou domestiques présents dans les ouvrages). La caractéristique des domestiques ou valets est appliquée entre parenthèses uniquement si le métier est très représenté (femme de chambre, portière, cuisinière, etc.).

Exemples :

Vieil ambassadeur de Sardaigne sera indexé :
Ambassadeur de Sardaigne

Valet de chambre de XXX sera indexé :
Valet de chambre

Vieille femme de chambre de XXX sera indexée :
Femme de chambre (vieille)

Titres de noblesse
Ils sont inclus dans la description de la personne et dans le nom sous lequel ils sont indexés. Il peut s'agir des titres suivants :
comte, comtesse, vicomte, vicomtesse, baron, baronne, marquis, marquise duc, duchesse, prince, princesse.

Exemple :
baron MACHIN DE LA CHOUETTE
LA CHOUETTE (baron Machin de)

Madame, mademoiselle, monsieur
Si une personne n'est connue que sous l'un de ces termes suivi du patronyme, on l'indexera comme suit :

MACHIN (madame)
MACHIN (mademoiselle)

MACHIN (monsieur)

Les personnes connues comme «le père Machin» ou «la mère Machin» seront indexées de la même manière :

MACHIN (le père)
MACHIN (la mère)

Nom identique sur plusieurs générations
Les personnes qui ont le même nom que des aïeux ou descendants sont distinguées autant que possible par le rang familial dans l'index (mère, fille, père, fils, etc.) :

MACHIN (mère)
MACHIN (fille)

Casse-pieds notoires :
- Les personnes se remariant à plusieurs reprises, reprenant un nom de jeune fille, changeant d'identité, possédant plusieurs états civils (espagnol, français, etc.), ayant un nom de guerre, un pseudonyme, un sobriquet, un surnom, etc. sont indexées suivant le choix effectué dans la Pléiade, qui tranchera.

1.2.7 Sensibilité à la casse

La sensibilité à la casse est primordiale pour le balisage correct des fichiers XML-TEI : elle permet de distinguer les noms propres des noms communs homonymes, les différentes graphies d'un même nom que nous propose Balzac, et permet d'éviter de confondre le Tasse avec une tasse, Racine avec une racine, etc.

Termes répétés pouvant comporter une majuscule

Cependant, il peut être pénible lors de l'indexation de noter systématiquement deux fois des mots apparaissant tantôt avec une lettre capitale, tantôt sans, alors que l'on doit impérativement indexer dans les bases de données toutes les formes d'un même nom tel que l'écrit Balzac, majuscule ou non. Il s'agit par exemple du cas suivant, où ces deux graphies doivent impérativement être renseignées :

madame de Latournelle
Madame de Latournelle

Intégration d'exceptions à la sensibilité à la casse

Le script Rabourdin intègre des exceptions pour certains termes qui ne doivent pas être sensibles à la casse, comme « madame », « monsieur », « mademoiselle », « lord », « lady », etc. Ce sont des termes que l'on retrouve en début de phrase, d'où l'emploi de la majuscule.

D'autres termes insensibles à la casse peuvent être indiqués à Rabourdin. Pour ce faire :

- Ouvrir Rabourdin dans un éditeur de texte de type Notepad++
- Indiquer ligne 13 les termes que l'on souhaite rendre insensibles à la casse.

Interdictions

Attention, il est interdit d'inscrire « le », « la », « les », « de », « du », ou toute autre particule comprise dans un nom propre ! Seuls des termes comme « madame », « monsieur », « mademoiselle » ou certains titres doivent être renseignés.

1.2.8 Noms indexés et non indexés

Sont indexés :

- Tous les noms à l'exception des noms non indexés (voir ci-dessous) ;
- Les noms des personnages religieux (apôtres, prophètes...) à l'exception des saints et de Dieu : sont donc indexés Jésus-Christ, la Vierge Marie, Adam et Eve, Noé, Mahomet, etc. ;
- Les tribus peu connues permettant de comprendre le caractère ou l'origine d'un personnages (les Abencerrages).

Ne sont pas indexés :

- Les noms compris dans les dédicaces ;
- Les noms de personnages constituant des titres de romans cités (« vous lirez dans LOUIS LAMBERT que… ») ;
- Les saints (Saint Michel, Sainte Geneviève, la croix de Saint-Louis, le pont Sainte-Anne, la ville de Saint-Denis...) ;
- Dieu ;
- Les épouses des rois, filles ou fils de rois, etc., non cité(e)s précisément (Madame la Dauphine, …) ;
- Le nom de Balzac lui-même tel qu'il apparaît en fin d'ouvrage ou dans les éventuelles parties liminaires ;
- Les peuples, tribus, ethnies (les Arabes, les Perses...), à l'exception des tribus peu connues dont certains personnages sont originaires et qui aident à les comprendre ;
- Rose et Colas, dans *Une Double Famille*.

1.3 Présentation des scripts

1.3.1 Dutocq

Dutocq sert à générer un index en .html à partir de toutes les bases de données pré-constituées, qu'il interroge pour en extraire des informations. Dutocq sait aussi débugger : détection des doublons, des erreurs de structuration des bases de données en .txt et d'identifiants uniques improprement attribués à un personnage.

1.31.2 Rabourdin

Rabourdin sert à lire une base de données en .txt pour en extraire les noms des personnages, à partir desquels il balise automatiquement un fichier XML-TEI. Il ajoute simplement des balises <persName> à chaque personnage fictif ou personne réelle rencontrée au fil de sa lecture du fichier TEI ainsi qu'un attribut @ref lui assignant un identifiant unique. Sensible à la casse, il intègre des exceptions qui simplifient la construction des bases .txt en évitant d'y consigner des termes pour lesquels la casse ne doit exceptionnellement pas avoir d'importance.

1.3.3 Colleville

Colleville sert à la fois à vérifier si le balisage automatique des noms dans le fichier XML-TEI s'est

réalisé correctement, sans avoir besoin d'ouvrir ce fichier TEI pour le relire entièrement, et à vérifier si les noms entrés dans les bases de données de chaque ouvrage ont bien été pris en compte : il surligne les noms balisés dans la version HTML de l'ouvrage, permettant de voir en un instant les erreurs ou oublis.

1.3.4 Bixiou

Bixiou sert à ranger automatiquement les noms de tous des personnages déjà indexés par ordre alphabétique, puisque l'index généré en HTML ne permet pas de le faire. Il faut lancer ce script pour pouvoir repérer qui se situe où dans l'ordre alphabétique afin de repérer plus facilement d'éventuels doublons.

1.3.6 txt2json

txt2json est un outil de conversion. Il sert à transformer les bases de données en .txt, aisément manipulables sans compétences en informatique, en fichiers .json, requis par les développeurs.

1.3 Repérage des doublons

1.3.1 Repérages sur l'index en cours de construction

Les doublons sont évités en cherchant à chaque fois qu'un nom apparaît lorsqu'on lit l'ouvrage en cours d'indexation si ce nom ne se trouve pas déjà dans l'index. Malgré cette précaution, les doublons restent parfois inévitables, si par exemple un personnage possède plusieurs noms, en a changé, ou si la personne qui indexe a besoin d'aller dormir.

1.3.2 Lancement du script de détection des doublons

Le script bixiou.py a été écrit pour résoudre ce problème : il aide à détecter les doublons en rangeant simplement l'index par ordre alphabétique une fois que ce dernier est généré en HTML. On peut ensuite aisément voir les répétitions. cf. section 1.3.4 ci-dessus.

Pour détecter les doublons à l'aide de Bixiou :

Créér un fichier nommé tri.txt.
Copier-coller dans ce fichier l'ensemble des noms générés par l'index
Ouvrir le terminal et lancer le script Bixiou en tapant :

python bixiou.py

La liste rangée dans l'ordre alphabétique va apparaître aussitôt. Il suffit de la parcourir pour relever immédiatement les éventuels doublons.

1.4 Suppression des doublons

1.4.1 Principe général

La suppression des doublons se fait en vérifiant toutes les apparitions d'un nom dans les différentes bas où on le retrouve. Une fois toutes les bases connues, on choisit la description à conserver et on la copie-colle à l'identique dans toutes les autres bases où ce même nom apparaît.

1.4.2 Procédure

1. Ouvrir tous les fichiers _P ou _R où le doublon apparaît.
2. Choisir le nom qui nous convient le plus (description plus complète de la personne dans l'un des deux fichiers, etc.) et copier-coller l'ensemble du paragraphe (vedette, relations et alias) dans les autres fichiers où ce doublon apparaît, en supprimant l'ancienne entrée du nom en double. En faisant cela, on relève l'ID que l'on supprime définitivement (le numéro d'identification).

3. Ouvrir le fichier P ou R encore vide du prochain ouvrage à baliser (P ou R selon la personne que l'on a rencontrée) et ajouter l'ID à ce fichier, suivi des 5 chevrons réglementaires. Ce nouvel ID est désormais disponible.

1.5 Balisage automatique en XML-TEI

1.5.1 Balisage par défaut

Le balisage par défaut est effectué par Rabourdin, qui ajoute une balise <persName> encadrant chaque nom trouvé. Cette balise contient un attribut @ref donnant l'identifiant unique de la personne.

Exemple : <persName ref="148P">Marie Willemsens</persName>

1.5.2 Balisage personnalisé

cf. 1.1.3 Procédures d'indexation personnalisées et possibilités d'ouverture

1.5.2.1 Possibilités d'affinage

cf. 1.1.3 Procédures d'indexation personnalisées et possibilités d'ouverture

1.5.1.2 Modification personnalisée des scripts python pas à pas

L'ensemble des scripts python est commenté pour faciliter la modification des fichiers. S'y référer.

1.5.3 Affichage HTML des noms balisés en XML-TEI

L'affichage des noms balisés se fait à l'aide sur script Colleville.

Génération de l'index pas à pas

1.1 Préparation des fichiers pour l'indexation

Avant de commencer à indexer, il faut créer sur son ordinateur un dossier – peu importe son nom et son emplacement – dans lequel on travaillera. C'est dans ce dossier qu'on lancera le terminal pour exécuter les scripts. Ce dossier contiendra tous les logiciels et fichiers relatifs à l'indexation, que l'on téléchargera depuis GitHub, c'est-à-dire :

- Toutes les bases de données P & R déjà constituées et celles en cours de création ;
- Tous les fichiers de débalisage déjà constitués et ceux en cours de création ;
- Tous les scripts python et awk : Dutocq, Rabourdin, Colleville, Bixiou, txt2json ;
- Une copie de tous les fichiers XML des ouvrages que l'on veut baliser et sur lesquels on a déjà travaillé, ou sur lesquels on travaille actuellement ;
- Une copie de tous les fichiers HTML correspondant aux ouvrages sur lesquels on a travaillé, ou sur lesquels on travaille actuellement ;
- Un dossier nommé img, contenant les blasons, gravures et icônes de l'index ;
- Un dossier nommé ttf, contenant les typographies de l'index ;
- Un fichier nommé index.html ;
- Un fichier nommé index-template.html ;
- Un fichier nommé style.css ;
- Un fichier nommé highlighting.css.

Les cinq premiers éléments de la liste ci-dessus permettent de construire l'index grâce aux scripts et aux bases de données, les six derniers éléments permettent de le visualiser grâce à une page HTML mise à jour à chaque fois que le travail d'indexation se poursuit, nommée index.html.

On pourra regrouper tous ces éléments dans des sous-dossiers comme on voudra si besoin, mais il faudra veiller à laisser impérativement dans le même dossier ou sous-dossier les bases de données + les 4 scripts python + les copies des fichiers XML et HTML. Les autres fichiers peuvent être rangés dans des sous-dossiers, mais il faudra alors veiller à bien lier les deux feuilles CSS (style.css et highligting.css) aux deux documents HTML index.html et index-template.html.

Pour télécharger l'ensemble de ces fichiers et scripts :

https://github.com/Armellei/IndexBalzac

1.2 Écriture des bases de données

1. Faire une copie du fichier HTML et du fichier XML de l'ouvrage que l'on veut baliser et déposer ces deux copies dans le dossier de travail.
2. Créer dans le dossier un fichier P.txt, un fichier R.txt, un fichier de débalisage DEBUG.txt. Ouvrir ces trois fichiers dans Notepad++.
3. Ouvrir le texte que l'on veut baliser dans sa version HTML ou l'afficher depuis le site ebalzac.com.
4. Inscrire dans le fichier P.txt le dernier identifiant unique P non encore attribué, que l'on trouvera en ouvrant le dernier fichier P constitué. Inscrire une cinquantaine de nouveaux identifiants non

encore attribués.
5. Répéter la procédure pour le fichier R.
6. Lire le texte que l'on veut baliser ligne par ligne. À chaque nom relevé, vérifier s'il existe dans l'index.html : si c'est le cas, copier-coller sa description depuis la dernière base dans laquelle il a été inscrit (l'index en HTML l'indique), sinon, l'inscrire et lui attribuer l'un des identifiants disponibles.
7. Répéter la procédure jusqu'à la fin du texte.
8. Si des termes nécessitent un balisage manuel (homographes, etc.), utiliser le fichier de débalisage pour indiquer ces cas particuliers.

1.3 Lancement des scripts générant l'index

9. À la fin de l'indexation, déposer la copie du fichier HTML du livre sur lequel on travaille sur l'icône de Colleville. Débugger si nécessaire.
10. Après constitution définive des bases de données et du fichier de débalisage, double-cliquer sur Dutocq pour générer l'index.
11. Débugger si nécessaire à l'aide des messages d'erreurs.
12. Ouvrir Index.html et vérifier rapidement que l'indexation a fonctionné (les nouveaux noms fraîchement balisés doivent être visibles).
13. Déposer le fichier XML correspondant à l'ouvrage qu'on veut baliser sur l'icône de Rabourdin.py qui balise automatiquement en TEI.
14. Débugger si nécessaire. Cf Erreurs fréquentes lors de l'indexation & débuggage des scripts.

→ Les scripts une fois lancés doivent se fermer tout seuls. Si ce n'est pas le cas, c'est qu'ils affichent une erreur dans la console qu'il faut corriger.

1.4 Vérifications

15. Utiliser le script Bixiou pour vérifier la présence de doublons en rangeant les noms par ordre alphabétique : créer un fichier nommé « tri.txt », copier-coller dans ce fichier l'ensemble des noms de l'index HTML et lancer Bixiou qui affichera les doublons. Corriger les erreurs éventuelles.
16. Ouvrir l'index HTML et vérifier que les corrections ont été faites.
17. Répéter la procédure à partir du point 1 pour l'ouvrage suivant.

Iconographie

1.1 Traitement des gravures, icônes et blasons
1.2 Recommandations de retouche

Gravures – icônes – blasons

Gravures :

Taille : 1350 x 2000px
Fond : blanc
Netteté : au min à 100%
Format : .jpg

Icônes de gravures :

Taille : 400 x 400px
Diamètre du rond : 400px
Fond : transparent
Format : .png

Blasons :

Taille : 1670 x 2237px
Fond : transparent
Format : .png

Recommandations de retouche pour ajouter des gravures à l'index (sous Photoshop) :

- Mode colorimétrique : niveaux de gris
- Équilibrer les blancs, les gris et les noirs (Niveaux + Courbes)
- Obtenir un fond blanc uniforme (rgb 255,255,255)
- Obtenir une netteté minimale à 100% (régler le filtre Accentuation : Filtres > Renforcement > Accentuation)
- Suppression des artefacts au maximum
- Augmentation des noirs sur les noms des graveurs + illustrateurs + légende en bas de page si besoin pour qu'ils soient bien lisibles
- Suppression des tampons de bibliothèque (outil Remplir utile s'il ne déforme pas la gravure : Édition > Remplir > Remplir avec Contenu pris en compte > Fusion normale, Opacité 100%)
- Retoucher en .psd ou .eps uniquement avant l'export final en .jpg ou .png
- Taille, format : voir ci-dessus (« Gravures »)

Recommandations

- Ne pas inscrire les noms comportant le terme « madame », « mademoiselle » ou « monsieur » (termes insensibles à la casse par défaut) avec une majuscule à « madame », « mademoiselle » ou « monsieur ».

- Penser à effectuer systématiquement une recherche dans chaque fichier P ou R sur les noms « louches », à problèmes, suspects.Vérifier systématiquement dans l'index de la Pléiade tous les noms « louches » et inscrire dans le fichier de débalisage tout doute.

Exemple : Le Tasse est simplement cité par Balzac sous la forme « Tasse » au sein d'une phrase et peut, de fait, donner lieu à un certain nombre de balisages incongrus de tasses (l'objet) en persName.

- Inscrire systématiquement dans le fichier de débalisage les termes à dé-baliser car deux personnes portent le même nom, ou partagent un prénom, un nom, un surnom, un titre de noblesse ou honorifique en commun.

Exemple : *Bettina* désigne dans Modeste Mignon à la fois la mère et la fille selon le contexte. Faire le choix de baliser la personne qui apparaît le plus souvent si possible, sinon celle qui est la première à apparaître dans le texte, et penser à inscrire dans le fichier de débalisage le nom de l'autre pour changer son ID manuellement et l'attribuer au bon homonyme.

- Conserver impérativement deux bases de données par livre (ne jamais les fusionner), sous peine d'obtenir un nombre incalculable d'homonymes faussant le travail. L'indexation doit toujours se poursuivre livre par livre, dans l'ordre défini sur eBalzac.com en version Furne corrigé.

Erreurs fréquentes lors de l'indexation & débuggage des scripts

1.1 Lecture des indications de débuggage données par les scripts

1. Fonctionnement du débuggage et typologie d'erreurs

Deux scripts servent à débugger : Dutocq et Bixiou.
Dutocq s'occupe des bases de données : il affiche nos erreurs dans les bases. Bixiou s'occupe de l'index : il trie les noms par ordre alphabétique pour vérifier les doublons.

Deux types d'erreurs peuvent apparaître : des erreurs sur les personnages (doublons, homonymes, oublis...), ou des erreurs sur la syntaxe des bases de données, qui font planter les scripts et empêchent l'index de se générer tant que l'erreur n'est pas résolue. Les scripts sont conçus pour empêcher le travail de se poursuivre tant que subsistent des problèmes.

Les erreurs humaines liées à l'indexation d'un grand nombre de personnages sont souvent les suivantes :
- inscription d'un personnage existant déjà dans une base de données sous un nouvel identifiant ;
- ajout de nouvelles informations dans la fiche d'un personnage existant dans d'autres bases non mises à jour elles aussi ;
- erreurs de syntaxe dans la constitution des bases de données : espaces, @ ou > manquants, champs obligatoires non remplis dans un enregistrement, etc.

Ne pas oublier de sauvegarder impérativement chaque base de donnée constituée en la déposant sur GitHub une fois le travail effectué et corrigé.

2. Exemples

Les messages renvoyés par les scripts s'affichent dans le terminal, comme suit :

1.2 Erreurs sur les personnages

1.1.1 Suppression des doublons

cf. 1.3 Repérage des doublons

1.1.2 Suppression de données différentes sur un même personnage

cf. 1.3 Repérage des doublons

1.2.3 Erreurs de remplissage des champs en txt

cf. .2.6 Nomenclature et syntaxe des bases de données

1.3 Erreurs de syntaxe

1.2 .1 Erreurs de syntaxe des fichiers txt et JSON

cf. 1.2.6 Nomenclature et syntaxe des bases de données

1.2.2 Erreurs d'encodage en UTF-8 et gestion des glyphes spéciaux

Encodage en UTF-8
L'encodage des caractères dans les fichiers P et R doit toujours se faire en UTF-8. Si l'encodage est différent, des erreurs peuvent apparaître : convertir le fichier qui pose problème et relancer les scripts.

Glyphes pénibles

Espaces fines insécables

Certains noms ne vont pas s'afficher correctement dans le fichier HTML de sortie du texte : les noms coupés par une espace fine insécable en font partie. Il faut copier-coller directement dans le HTML ou dans une autre database déjà constituée ces noms, espace fine insécable incluse, et ne pas retaper le nom depuis son clavier en utilisant la barre espace lorsqu'on veut les inscrire dans une base de données!

Noms en petites capitales dans le fichier XML

Les noms en petites capitales constituent un problème non résolu encore : ils sont reconnus par le fichier XML via Rabourdin une fois sur deux. Il faut donc les baliser manuellement dans le fichier XML pour les prendre en compte.

Noms avec exposant

Les noms qui comportent un <sup> dans le fichier XML ne sont pas balisés entièrement : on s'arrête avant les caractères alphanumériques en exposant et on ne balise que ce qui précède. Par exemple : Charles I^{er} → on entre dans la base de données « Charles I », c'est tout.

Caractères spéciaux

Certains caractères auront beau être inscrits dans les bases de données, il ne seront pas reconnus par le fichier XML qui ne les balisera pas. Il peut s'agir par exemple des caractères suivants, qu'il conviendra alors de remplacer dans la base de données txt par leur code HTML :

&#00C1; = Á
#U+00D0; = Đ
etc.

1.4 Erreurs multiples

Plusieurs erreurs peuvent apparaître sur un même ouvrage : les scripts en informent dans la console qui s'ouvre automatiquement après avoir cliqué sur Entrée pour passer à l'erreur suivante. Lorsqu'on a cliqué plusieurs fois sur entrée et qu'il n'y a plus d'erreur, la console se ferme toute seule.

Grève des employés

Rabourdin refuse de se lancer :

Cela peut être à cause d'un caractère mal géré en UTF-8 : le caractère *Á* par exemple ! Supprimer ce caractère, et inscrire le nom qui le contient dans le fichier de débalisage pour faire un balisage manuel, ou remplacer ce caractère par son code HTML (&#XXXX;).

Colleville ne balise rien :

Si Colleville ne balise rien et affiche un message d'erreur qui ferme la fenêtre automatiquement : il y a une erreur dans les fichiers P ou R. Lancer alors Dutocq qui va afficher l'erreur en question. La corriger. Relancer Colleville.

Rabourdin ne balise rien :

Les bases de données ne sont pas dans le même dossier que Rabourdin.py. Les placer dans le même dossier, et recommencer à lancer le script.

→ Les messages affichés par Rabourdin, Dutocq et Colleville sous la forme « X was not found ! » sont normaux : il ne faut pas en tenir compte, la console va se fermer toute seule et il ne s'agit pas d'une erreur, mais d'une information sur une graphie d'un nom qui n’apparaît pas dans le texte sur lequel on travaille alors qu'elle a déjà pu être trouvée dans un autre texte. Ce cas de figure se rencontre uniquement chez les personnages reparaissants possédant des alias, déjà inscrits dans une ou plusieurs bases de données constituées avant celle sur laquelle on est en train de travailler.

Mise en ligne des fichiers

1.1 Mise à jour des bases de données depuis /IndexBalzac sur GitHub

Les bases de données déjà constituées, complètes et débuggées sont librement disponibles à l'adresse suivante : https://github.com/Armellei/IndexBalzac/tree/master/Database

1.2 Préparation des fichiers pour intégration web

Les fichiers à livrer pour l'intégration de l'index sont :

- les bases de données P et R converties en JSON par txt2json
- les fichiers XML-TEI balisés par Rabourdin
- les images retouchées (gravures, icônes, blasons)

Il est inutile de livrer les fichiers de débalisage, l'index dans sa version HTML ou les scripts.

Mise à disposition du code

Les codes de chaque script sont disponibles librement sur GitHub à l'adresse suivante : https://github.com/Armellei/IndexBalzac/tree/master/Indexing%20scripts