Documents structurés – TALA540a

Projet fin de semestre

Informations générales

Date de rendu : **10 janvier 2021** jusqu'à 23 h 59. La date de votre commit fera foi.
Dépôt : sur votre Github, dans un nouveau projet.

Présentation

L'objectif de ce projet est de réaliser une chaîne de traitement pour la publication de documents en ligne. À partir de documents qui vous seront fournis, vous allez devoir proposer un site web pour présenter le contenu de ces documents et interagir avec.

Deux grandes étapes seront mises en œuvre.

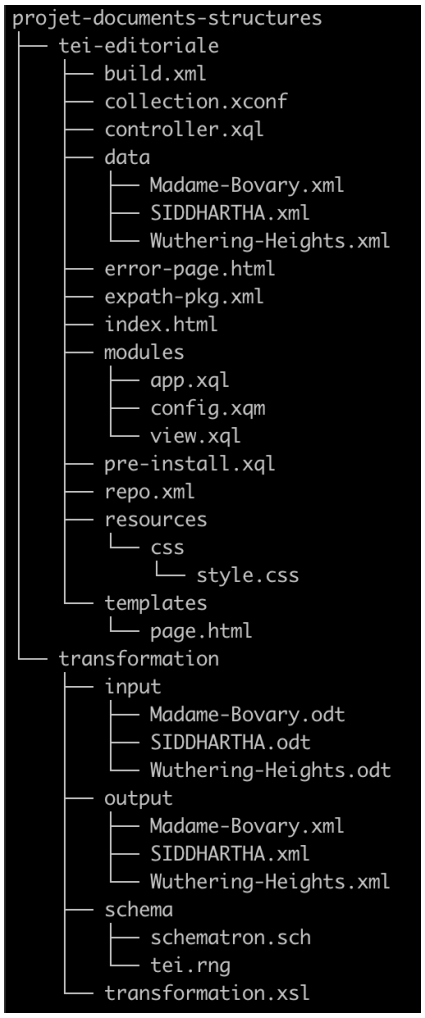
La première étape consistera à transformer les documents au format OpenDocument en document XML selon le standard de la Text Encoding Initiative (TEI). Pour cela, son schéma vous sera communiqué afin de vous permettre de valider la structure. De plus, un Schematron vous permettra de respecter l'encodage attendu de certaines métadonnées.

Une fois cette étape terminée, les documents seront stockés dans une base de données XML (eXist-db). À partir de celle-ci, vous allez devoir réaliser un site internet à l'aide de xquery et xslt. Sur ce site seront présentées trois interfaces : une pour la recherche d'informations dans les documents, une autre pour afficher vos documents mis en page et finalement une page pour présenter le projet.

Explications

Architecture

Il est impératif de respecter la structure suivante pour organiser votre projet. Pour la partie avec eXist-db, il faudra procéder à un export de l'application (ici, tei-editoriale) selon la documentation et dézipper celle-ci avec la commande « unzip ».



Étape 1 – 10 points

Trois documents « Madame-Bovary.odt », « SIDDHARTHA.odt » et « Wuthering-Heights.odt » vous sont fournis. Nous avons vu lors du premier cours que le format OpenDocument est une archive de plusieurs fichiers XML. Vous allez devoir explorer cette archive pour extraire le contenu du document et ses métadonnées. Pour se faire, il va falloir employer XSLT pour réaliser cette extraction.

Le schéma de la TEI va vous permettre de générer le document XML en vous guidant sur la structure à adopter. De plus, certaines métadonnées nécessiteront un formatage supplémentaire qui vous sera indiqué par un Schematron. Ils sont tous les deux dans le dossier « transformation/schema ».

Les documents odt sont *stylés* avec des « styles personnalisés », grâce à eux vous pourrez effectuer l'équivalence avec les balises de la TEI. Il est également possible d'attacher des métadonnées à un document (File > Properties), celles-ci figureront dans le teiHeader de votre fichier XML. L'équivalence des styles et métadonnées vous est donnée en dessous.

Nom de la métadonnée / du style	équivalent XML TEI
Titre	title
Auteur	author
Licence	availability
Date de publication	publicationStmt/date
Source	bibliography
Description	projectDesc
Date de la source	creation/date
Title	head (qui englobe celles du Heading_n)
Heading_n	head (où *n* indique un chiffre)
Text Body	p
gras	hi/@rend
italique	hi/@rend
citation	quote

Pour cette étape, il est nécessaire d'utiliser XSLT 3. Vous pouvez vous servir d'Oxygen XML Editor et de toutes les ressources en ligne. N'hésitez pas à fouiller la documentation de la TEI ou bien chercher des exemples.

Bonus : réalisation d'un script pour appliquer vos transformations à vos documents.

Étape 2 – 10 points

Les documents XML générés vont être manuellement ajoutés dans eXist-db. Exist-db permet de créer des sites à l'aide de xquery. L'objectif de cette étape est de réaliser un site pour présenter vos documents. Il y aura trois interfaces à réaliser.

La première interface sera la page d'accueil de votre site. Votre contrainte pour cette page est de la construire à partir d'un document XML qui sera transformé par XSLT pour un affichage HTML. Vous devrez y joindre une grammaire RelaxNG pour structurer votre page qui utilise des patrons nommés. Vous présenterez sur cette page :

- le projet et les étapes de mise en œuvre
- la navigation sur le site et ce qu'on peut y trouver
- les difficultés rencontrées

La seconde interface sera une page où l'on pourra effectuer de la recherche plein texte sur vos documents. Ainsi que la possibilité de sélectionner les documents selon les métadonnées grâce à des formulaires de recherche. Par exemple, on pourra choisir les documents créés en 2000 pour afficher

leur nom puis effectuer une recherche sur le mot « esprit » à l'intérieur. Cette partie sera réalisée en xquery.

Finalement, la troisième interface sera pour vous l'occasion de proposer un rendu graphique de vos documents XML. C'est-à-dire présenter le contenu des documents éditorialisé sur une page HTML de façon enrichie avec de la css. L'affichage pourra respecter celui du document original ou bien celui que vous avez choisi. Ici, vous pouvez utiliser à la fois xslt et/ou xquery.