

# ENC/ XSLT/ TD XML vers XML (séance 5)

Jean-Damien Généro

2023

## Environnement de travail

- Document XML : `class5_st_julien_hospitalier.xml` (il s'agit du même document que celui utilisé pour la séance 4).
- Document XSL : `class5_TD.xsl` (il s'agit des règles écrites lors de la séance 4, cf. `class-4/exercices_class4.pdf`).

## Consignes

### 1. La bibliographie

- Dans `<listBibl>`, remplacer chaque référence contenue dans `<biblStruct>` par un `<bibl>` reproduisant le contenu des commentaires.
  - Utiliser des boucles et conditions ;
  - Trier les références par ordre alphabétique ;
  - Les guillemets français correspondent aux entités `&#171;` et `&#187;`.
- Résultat attendu :

```
<listBibl>
  <bibl>DORD-CROUSLÉ Stéphanie, « Établir le texte de Trois contes »,
    <hi rend="i">Flaubert. Revue critique et génétique</hi>
    [<ref target="https://journals.openedition.org/flaubert/4143">
    en ligne</ref>], n°24, 2020.</bibl>
  <bibl>NARR Sabine, « Le Saint Julien de Flaubert - une poétique
    du vitrail narrativisé », <hi rend="i">Les Dossiers du Grihl</hi>
    [<ref target="http://journals.openedition.org/dossiersgrihl/6447">
    en ligne</ref>, DOI : <ref target="https://doi.org/10.4000/
    dossiersgrihl.6447">10.4000/dossiersgrihl.6447</ref>], n°9-1,
    2015.</bibl>
  <bibl>SAMSON Véronique, « Le roman après la fin », dans
    <hi rend="i">Après la fin. Gustave Flaubert et le temps du
```

```

        roman</hi>, chap. 2, 2021, p. 105-190.</bibl>
</listBibl>

```

## 2. Le <text>

- Reproduire les balises <head> et <div>.
- Numéroté en continu les <p>.
- Transformer les <hi rend="b"> en <persName ana=""> et les <hi rend="i"> en <placeName ana="">, le @ana devant à chaque fois avoir pour valeur le texte contenu dans la balise, mais sans les apostrophes et avec les espaces remplacés par des tirets bas (\_).
  - Exemple : <hi rend="b">empereur d'Occitanie</hi> est transformé en <persName ana="empereur\_doccitanie">empereur d'Occitanie</persName>.
  - Pour éviter une erreur de validation (“Unmatched quote in expression”), utiliser la variable apos (en début de fichier).
- Reprendre l’index de la séance 4 et :
  - Écrire en majuscule les entrées d’index ;
  - Ajouter aux <p> un @corresp avec pour valeur : # suivi du contenu des @ana créés plus haut.
- Résultat attendu (dans le <body>, après les <div> des chapitres) :

```

<div>
  <head>Index</head>
  <div type="name">
    <p corresp="#père">père : 1.</p>
    <p corresp="#mère">mère : 1.</p>
    <p corresp="#julien">Julien : 1, 2.</p>
    <p corresp="#dauphin_de_france">dauphin de France : 2.</p>
    (...)
  </div>
  <div type="place">
    <p corresp="#milan">Milan : 2.</p>
    <p corresp="#oberburbach">Oberburbach : 2.</p>
  </div>
</div>

```

## 3. Un document par chapitre et index

### 3.1 Avant de commencer : <xsl:result-document/>

- <xsl:result-document/> permet de générer plusieurs documents de sortie. C’est une “nouveauité” de XSLT 2.0.
- Cette instruction s’utilise comme enfant d’un <xsl:template/>.
- L’attribut @href permet d’indiquer l’URI du document à générer. Elle peut être absolue (depuis la racine) ou relative (dans ce cas, le processus part de l’emplacement du fichier d’entrée).

- L'attribut `method` permet de spécifier le format du code de sortie.
- L'attribut `indent` permet d'indiquer si le résultat sera indenté.

### 3.2. Consignes

- Utiliser l'instruction `<xsl:result-document/>` pour générer trois documents de sortie : `st_julien_chap1.xml`, `st_julien_chap2.xml` et `st_julien_index.xml`.
- Utiliser une variable pour stocker le `<teiHeader/>` **qui sera le même pour chacun des trois documents**.
- Ajouter les `<div>` de chaque chapitre *avec les modifications définies précédemment* et celle de l'index dans le fichier correspondant. Les `<head>` des fichiers des chapitres doivent contenir le mot **Chapitre** suivi du numéro.