

Introduction

Rappel 1

Où puis-je trouver les styles CSL dans Zotero ? Localiser le répertoire de données `zotero`, puis répertoire `styles`.

Rappel 2

Tous les problèmes de mise en forme ne proviennent pas des styles bibliographiques, vérifiez la qualité et la complétude des données de votre bibliothèque Zotero en premier lieu !

Rappel 3

Le plus difficile n'est pas forcément d'encoder le style, mais de disposer de consignes claires et précises, traduisibles dans un langage informatique.

==> un exemple à suivre pour les revues, les indications de infoclio pour les règles de citation : <https://www.infoclio.ch/fr/node/139278>. Cette page précise en outre de façon détaillée les champs Zotero utilisés pour chaque type de document.

Principes généraux de CSL

Principes du langage XML

Prologue XML

C'est la première ligne du fichier CSL. Il contient la déclaration XML et spécifie le codage des caractères. Il se présentera ainsi dans la plupart des cas : `<?xml version="1.0" encoding="utf-8"?>`.

Éléments et hiérarchie

Les éléments sont les blocs de base avec lesquels un fichier XML est construit. Ils peuvent être imbriqués hiérarchiquement : on parle d'éléments **parent** et d'éléments **enfant**. Le premier élément est l'élément **racine** (`style` dans le langage CSL), duquel tous les éléments dépendent. On indente généralement les éléments enfant par des espaces ou des tabulations pour faciliter la compréhension. Chaque élément est introduit par une balise `<ouvrante>` et clos par une balise `</fermante>`, ou une barre oblique s'il n'a pas de contenu. Toute balise ouverte doit impérativement être fermée. Ex:

```
<element parent>
  <element enfant1 attribut="valeur"/>
  <element enfant2> text </element enfant2>
</element parent>
```

==> ici il faut comprendre que l'élément enfant 1 est qualifié par un attribut et que le contenu de l'élément enfant 2 est du texte, entre deux balises (cf. ci-dessous)

Attributs et contenu de l'élément

Un élément peut être qualifié et contenir des informations de deux manières :

- Soit par du contenu textuel inséré entre la balise ouvrante et la balise fermante, ex:

```
<author>
  <name>Anton Perdoncin</name>
  <email>anton.perdoncin@gmail.com</email>
</author>
```

- Soit par des attributs. Si l'ordre des attributs importe peu, une valeur doit obligatoirement être renseignée, entre guillemets. Ex : `<link href="http://traces.revues.org" rel="documentation"/>`

Echappement

Pour éviter toute ambiguïté dans l'écriture XML, certains caractères significatifs pour la syntaxe XML doivent être substitués par d'autres lorsqu'ils sont utilisés dans un attribut ou dans le contenu textuel d'un élément. Les séquences d'échappement sont les suivantes :

- `<` pour <
- `>` pour >
- `&` pour &
- `'` pour '
- `"` pour "

Commentaires

Des commentaires pour expliciter des choix d'écriture ou clarifier des points peuvent être introduits sous la forme suivante : `<!-- commentaire libre à rédiger -->`. Ils seront reconnus par les applications utilisant le fichier comme du commentaire et non du code.

Fichier bien formé et valide

Contrairement à HTML, XML ne pardonne aucune erreur de syntaxe. Toute erreur (oubli d'une balise fermante, échappement incorrect, etc.) empêchera le fichier XML de fonctionner. Il convient donc de s'assurer que le fichier CSL fonctionne correctement en vérifiant qu'il est :

- **bien formé**, *i. e.* qu'il respecte les règles de XML et ne contient pas d'erreur d'encodage,
- **valide**, *i. e.* qu'il est conforme aux règles du schéma CSL, qui décrit tous les éléments CSL, leurs attributs et leurs règles d'utilisation.

Adapté de : Rintze M. Zelle, *Primer : An introduction to CSL - XML basics* - CC-BY-SA - Disponible sur : <http://docs.citationstyles.org/en/stable/primer.html#xml-basics>

Le « kit CSL »

Spécification CSL

<http://docs.citationstyles.org/en/stable/specification.html>

==> *Le Dictionnaire et la grammaire de CSL*

Correspondance types de documents/champs Zotero/CSL

<http://aurimasv.github.io/z2cs/typeMap.xml>

==> *Spécificités de Zotero : exemple la date originale est une variable CSL mais ne correspond pas à un champ Zotero. Il faut donc utiliser de façon spécifique le champ `Extra` pour renseigner une date originale (de même qu'un éditeur et un lieu de publication originaux).*

- Date originale d'un document = 1 variable de type date = `original-date`
- Lieu de publication original d'un document = `original-publisher-place`
- Éditeur original d'un document = `original-publisher`.

Ex encodage CSL : `<date variable="original-date" form="text" date-part="year"/>` Ces variables doivent être saisies dans le champ `Extra` de la notice Zotero, sous la forme :

```
{ -- 1891}
{ -- Librairie Ch. Delagrave}
{ -- Paris}
```

On peut aussi détourner des champs, mais cela doit être fait avec prudence :

- tous les champs ne sont pas disponibles pour tous les types de documents ;
- on risque de se priver d'un champ utile pour certains types de document ;
- il faut bien indiquer à tous les utilisateurs du style le détournement effectué.

Encodage HTML de la ponctuation (espaces insécables, tirets demi-cadratin, etc.)

<http://character-code.com/punctuation-html-codes.php> et http://www.mus.ulaval.ca/roberge/gdrm/08-codes.htm#espaces_tirets

==> *Spécificités françaises à encoder*

Fichier locale-fr

<https://github.com/citation-style-language/locales/blob/master/locales-fr-FR.xml>

==> *Éléments de traduction en français : traduction par défaut disponible*

Structure générale d'un style CSL

- `style` : élément racine - précise notamment la version de CSL, le type de style et permet des paramétrages globaux : gestion des noms à particules avec l'attribut `demote-non-dropping`, de l'indication du nombre de pages, abréviation des prénoms composés - spécification CSL : <http://docs.citationstyles.org/en/1.0.1/specification.html#the-root-element-cs-style>

==> Certains styles sont agnostiques du point de vue de la langue, ils ne comportent pas d'attribut `default-locale`. Ils peuvent être utilisés dans toutes les langues ; ce sont alors les paramètres par défaut définis dans les fichiers `locale` qui sont appliqués (voir lien vers le fichier `locale-fr-FR` dans le kit CSL). Certains styles sont quant à eux utilisables pour certaines langues seulement, ce qui permet de définir des spécificités propres à la langue différentes de celles prévues par défaut (en termes de ponctuation, de termes prédéfinis, etc.). Dans l'entrepôt des styles Zotero on peut retrouver les styles localisés dans une langue en indiquant le nom de la langue dans le formulaire de recherche : "french" affichera ainsi tous les styles ayant un attribut `default-locale="fr-XX"`.

- `info` : métadonnées décrivant le style (nom, auteur, etc.) - il faudra notamment modifier les éléments `title` et `id` du style que vous aurez créé à partir d'un autre

==> il faut bien changer l'`id` et le `title` sinon le style que vous avez créé sera écrasé lors des mises à jour du style duquel il est dérivé.

- `citation` : décrit la façon dont les appels de citation ou les notes (pour les styles **note**) sont formatés - spécification CSL : <http://docs.citationstyles.org/en/stable/specification.html#citation>
- `bibliography` : décrit la façon dont la bibliographie est formatée - spécification CSL : <http://docs.citationstyles.org/en/stable/specification.html#bibliography>
- `macro` : permet la réutilisation des règles de formatage dans `citation` et `bibliography` et des styles plus compacts - spécification CSL : <http://docs.citationstyles.org/en/stable/specification.html#macro>

==> Une macro est un programme qui permet d'être appelé (fonction), il est plus facile de faire des macro pour que ça soit plus propre. C'est aussi ce qu'on utilise pour faire des mises en forme conditionnelles. C'est l'une des bonnes pratiques d'écriture recommandées par adamsmith :

Bonne pratique 2 : Utilisez abondamment les macros. Gardez les éléments `citation` et `bibliography` brefs et ne leur intégrez qu'au minimum des éléments choose .

Source : Adamsmith. Writing CSL - Features and Best Practices. The Zoteroist, 2013. Disponible sur : <https://zoteromusings.wordpress.com/2013/10/28/writing-csl-features-and-best-practices>

- `locale` : permet de spécifier des termes, formats de dates et options de formatage différents de ceux prévus par défaut pour la langue - spécification CSL : <http://docs.citationstyles.org/en/stable/specification.html#locale>
- `terms` : permet la modification de chaînes de caractères spécifiques (ex. remplacer « edited by » par « ed. by ») - spécification CSL : <http://docs.citationstyles.org/en/stable/specification.html#terms> - voir aussi : <https://zoteromusings.wordpress.com/2013/10/28/writing-csl-features-and-best-practices>

Anatomie des éléments `style` et `info`

- Type de style : l'attribut `class`
- Style agnostique du point de vue de la langue vs. style localisé
- Style dépendant (ex : Loisir et Société / Society and Leisure) vs. style indépendant.

==> Pour économiser du code, certains styles sont dépendants d'un autre style. Dans ce cas là, le fichier CSL se réduit à un élément `info`. La référence au style principal est indiquée par l'attribut

`rel="independent-parent"` .

```
<?xml version="1.0" encoding="utf-8"?>
<style xmlns="http://purl.org/net/xbiblio/csl" version="1.0" default-locale="en-US"
  <!-- Generated with https://github.com/citation-style-language/utilities/tree/master/generate_dependent_styles/data/taylor-and-francis -->
  <info>
    <title>Loisir et Société / Society and Leisure</title>
    <id>http://www.zotero.org/styles/loisir-et-societe-society-and-leisure</id>
    <link href="http://www.zotero.org/styles/loisir-et-societe-society-and-leisure" rel="self"/>
    <link href="http://www.zotero.org/styles/apa" rel="independent-parent"/>
    <link href="http://www.tandfonline.com/action/authorSubmission?journalCode=RLES20&page=instructions" rel="documentation"/>
    <category citation-format="author-date"/>
    <category field="political_science"/>
    <issn>0705-3436</issn>
    <eissn>1705-0154</eissn>
    <updated>2014-05-17T16:46+00:00</updated>
    <rights license="http://creativecommons.org/licenses/by-sa/3.0/">This work is licensed under a Creative Commons Attribution-ShareAlike 3.0 License</rights>
  </info>
</style>
```

Les principaux types d'éléments

On manipule dans les appels de citation, les notes et la bibliographie des données de nature différente, auxquelles des paramétrages spécifiques peuvent être appliqués. A chaque type de donnée correspond ainsi un type d'éléments CSL. D'autres éléments, `group` et `choose`, par exemple, sont davantage des éléments de

rédaction que de contenu.

- `text` : le texte à afficher peut être celui d'une `variable`, d'une `macro`, d'un `term` ou d'une `value`
- `date` : date de publication, date de consultation, etc.
- `names` : noms des auteurs, des éditeurs scientifiques, des éditeurs commerciaux, etc.
- `number` : volume, numéro, numéro dans la série, etc.
- `label` : étiquette de certaines données, par exemple page ou p. pour page, vol. ou volume pour volume, sous la dir. de pour l'éditeur scientifique, etc.
- `group` : simplifie l'écriture d'un style en :
 - définissant la ponctuation au niveau d'un groupe d'éléments plutôt qu'élément par élément,
 - conditionnant l'affichage de certaines données à la présence d'autres (voir exemple : <https://zoteromusings.wordpress.com/2013/10/28/writing-csl-features-and-best-practices/>)
- `choose` : créer des conditions - voir *infra* les différents types de condition

Outils d'édition des styles

Quand j'ai commencé à faire des photographies, j'ai acheté un appareil argentique entièrement manuel.

Quand j'ai commencé à faire de la couture, j'ai acheté une machine entièrement manuelle.

Quand j'ai commencé à éditer des styles CSL, j'ai utilisé l'éditeur intégré à Zotero.

Editeur de style Zotero

==> Aller dans Zotero > Préférences > Avancées > Générales > Ouvrir Editeur de style

On sélectionne le style, des documents de sa bibliothèque, puis on actualise pour afficher la fenêtre de code et la prévisualisation des citations et de la bibliographie.

Avantages

- Toute erreur de syntaxe XML donne lieu à un message d'erreur immédiat : on voit tout de suite que son fichier n'est pas bien formé.
- On prévisualise en direct les modifications de paramétrage du style, et ce à partir des exemples de document de sa bibliothèque.

Limites

- Pas de coloration syntaxique
- Pas de contrôle de la validité CSL
- L'écran est découpé en 3 bandeaux horizontaux, ce qui peut occasionner une lisibilité très réduite en fonction de la taille de son écran

Editeur de style en ligne

<http://editor.citationstyles.org>

Rechercher un style à partir d'un exemple

Editer un style

2 modes d'édition :

- éditeur visuel, sous la forme d'un formulaire
- éditeur de code, avec coloration syntaxique

Avantages

- Coloration syntaxique
- Toute erreur de syntaxe XML donne lieu à un message d'erreur immédiat : on voit tout de suite que son fichier n'est pas bien formé
- Pas de contrôle de la validité CSL
- Prévisualisation en direct des modifications de paramétrage du style

Limites

- La prévisualisation n'est pas effectuée à partir des documents de sa bibliothèque, elle porte uniquement sur des exemples non personnalisables de livre et d'article

L'éditeur de texte

Quelques exemples d'éditeur :

- NotePad++ (pas Mac)
- TextWrangler (Mac)
- Atom (toutes plateformes)

Avantages

- Coloration syntaxique
- Contrôle de la validité CSL si on utilise un éditeur XML
- En fonction de l'éditeur, aides à la saisie et contrôle du caractère bien formé du fichier

Limites

- Pas de prévisualisation des modifications de paramétrage du style

Valider son style

Cet outil permet de vérifier que son fichier CSL respecte bien le langage CSL et est valide. NB un fichier peut être bien formé, respecter la syntaxe XML (balises ouvertes bien fermées, etc.) et ne pas être valide par rapport au schéma CSL. Or cette validité est indispensable pour installer le fichier CSL dans Zotero et pouvoir l'utiliser. <http://validator.citationstyles.org/> permet de valider son fichier et affiche le cas échéant de façon détaillée les erreurs d'encodage.

Exemples de modification mineures d'un style existant

Macro simple

Exercice de style *Nature*.

Exemple de la macro auteur Exemple commenté avec capture d'écran ici : https://paris-sorbonne.libguides.com/zotero_styles/macros

Modifiez le formatage des noms d'auteur pour :

- avoir le prénom en entier;
- relier le nom des deux derniers auteurs par and ;
- afficher la mention et al en gras.

==> aller dans la macro `author` . Faire bien attention à la ponctuation, il faut indiquer explicitement les espaces après les marques de ponctuation

Avant

```
<macro name="author">
  <names variable="author">
    <name sort-separator=", " delimiter=", " and="symbol" initialize-with=". " delimiter-precodes-last="never" name-as-sort-order="all"/>
    <label form="short" prefix=", "/>
    <et-al font-style="italic"/>
  </names>
</macro>
```

Après

```
<macro name="author">
  <names variable="author">
    <name sort-separator=", " delimiter=", " and="text" delimiter-precodes-last="never" name-as-sort-order="all"/>
    <label form="short" prefix=", "/>
    <et-al font-weight="bold"/>
  </names>
</macro>
...

```

==> On peut aussi changer le

```
* `delimiter` par exemple une ,
* `name-as-sort-order` : all on a toujours Nom-Prénom, Nom-Prénom etc.,
* `et-al` si on le veut en italique ou en gras `font-style="italic"` ou `font-weight="bold"`. On peut aller voir la documentation csl en recherchant l'

**Autre exemple :**

* pour le nom d'auteur en majuscule
```

...

```
## Macro conditionnelle

_==> ce sont des macros plus complexes où on peut utiliser `<choose>` pour définir des conditions._

### Les différents types de condition

Attributs possibles pour les éléments `if` et `else-if`. NB on peut indiquer plusieurs valeurs, sauf pour l'attribut `desambiguate`.

* `desambiguate="true"` (seule valeur possible de l'attribut) : la condition n'est réalisée que si elle permet de désambiguer
* `is-numeric="variable"` valeur de l'attribut = variable qui doit avoir un contenu numérique pour que la condition se réalise
* `is-uncertain-date="date"` valeur de l'attribut = date qui doit être incertaine pour que la condition se réalise [NB Zotero ne gère pas les dates ince
* `locator="XXX"` valeur de l'attribut = locator auquel la condition s'applique
* `position="XXX"` cf. infra, utilisées pour les notes de bas de page ou de fin
* `type="XXX"` valeur de l'attribut = type de document auquel la condition s'applique
* `variable="variable"` valeur de l'attribut = variable dont l'absence ou la présence conditionne la réalisation de la condition
  * `match="any"` = OU = la condition se réalise si au moins un des critères est rempli ex : `<if type="book thesis" match="any">` la condition se réa
  * `match="all"` = ET = la condition se réalise si TOUS les critères sont remplis
  * `match="none"` = NON = la condition se réalise si AUCUN des critères n'est rempli ex : `<if variable="volume" match="none">` la condition se réalis

**Exercice de style _Nature_.**

Exemple commenté avec capture d'écran ici : [https://paris-sorbonne.libguides.com/zotero_styles/macros](https://paris-sorbonne.libguides.com/zotero_styles/macros)
- Modifiez le formatage des titres d'actes de conférence, pour qu'ils soient affichés entre guillemets.

_Avant_
```

Après

_==> NB le `title` correspond au titre du document. Si on souhaite définir le formatage du titre de revue (pour les articles), du titre de livre (pour les

Élément citation

Exemples commentés et détails ici : http://paris-sorbonne.libguides.com/zotero_styles/citation
Exemple style numérique

_==> `sort` indique l'ordre (numérique ou bien chronologique ou par nom d'auteur), `layout` nous montre le rendu. Il y a des paramètres particuliers pour
Exemple style auteur-date

Exemple style note

Élément bibliography

Exemples commentés et détails ici : [http://paris-sorbonne.libguides.com/zotero_styles/bibliography](http://paris-sorbonne.libguides.com/zotero_styles/bib)
Un exemple :

`` ==> Ce sont ainsi des attributs de l'élément `bibliography` qui permettent de paramétrer notamment :

- le nombre d'auteur avant l'indication d'un `et-al-min="6"`,
- `second-field-align="flush"` pour le retrait de seconde ligne.

Enregistrer son style

==> Pour enregistrer son style : dans l'éditeur Zotero, pensez à bien changer le nom du titre et l'identifiant et faire 'enregistrer sous' pour générer le fichier CSL. Pour l'installer : soit on le fait glisser dans une fenêtre de Firefox, soit on va dans *Zotero* > *Préférences* > *Citer* > *Gestionnaire de style* et on peut rajouter ou supprimer des styles Pour Standalone, voir aussi : https://www.zotero.org/support/styles_Pour éviter les mauvaises surprises à cette ultime étape, n'oubliez pas de valider au préalable votre style grâce au validateur en ligne : <http://validator.citationstyles.org/>.

Organisation d'un travail collaboratif en France pour la création de styles de citation en SHS sur Trello

Allez sur : <https://trello.com/b/ACMPVFQf/csl-france>

Organisation du trello :

- style demandés
- style en cours de création
- styles prêts à l'usage

Questions fin de séance

- utiliser bien le forum Zotero pour poser ces questions (le forum est très réactif et souvent on a des réponses aux questions que d'autres se sont déjà posées)
- pour d'autres langues : Juris-M permet de gérer le multilinguisme
- utilisation de CSL : par Mendeley, Papers,
- lien avec LaTeX : l'une des meilleures solutions jusqu'à présent était Better Bib(La)TeX, mais pour l'instant sa compatibilité avec Zotero v5 n'est pas assurée.