

Arbre Notice papier

①

N° de la notice . TITRE . DATE

REGEX: $\^((([0-9]|I|E)+(-[0-9]+)?)(\.\.)\.*([A-Z]\{2,3\})+)$
 = Pas de pertes
 $\hookrightarrow \text{substitut: } <\!\!\text{TEI}\!\!> \n$
 $<\!\!\text{TEI}\!\!> \n = \$1 \n$

REGEX: $[0-9]I[E] + \.\.*[A-ZÉÈÀÙ]\{2,3\}$
 $\^([0-9]I[E]) + \.\.(.*[A-ZÉÈÀÙ])$
 $+ \.\. = \text{Pas de pertes}$
 $\hookrightarrow \text{substitut: } <\!\!\text{summary}\!\!> \$1 \n$
 $<\!\!\text{summary}\!\!> \n$

REGEX: $\^([0-9]I|E) + \.\.*[A-ZÉÈÀÙ]\{2,3\}$
 $\.\.* \hookrightarrow (\.\.*[0-9]IX) + \.\.) \n = \text{Pas de pertes}$
 $\hookrightarrow \text{substitut: } <\!\!\text{origDate}\!\!> \$1 \n$
 $<\!\!\text{origDate}\!\!> \n$

<summary>

Bibliothèque nationale , cote .

REGEX: $\^((Bibliothèque nationale),)\{1\} = \text{Pas de pertes}$
 $\hookrightarrow \text{substitut: } <\!\!\text{repository}\!\!> \2
 $<\!\!\text{repository}\!\!> \n$

REGEX: $\^((Bibliothèque nationale), \{1\})(\.*[0-9+] [A-Z]? \.\.)\$ \n = \text{Pas de pertes}$
 $\hookrightarrow \text{substitut: } <\!\!\text{idno}\!\!> \3
 $<\!\!\text{idno}\!\!> \n$

<repository>

Si 1^{er} § ne débute pas par un folio ou la mention
 d'un feuillet de garde

Condit Python ou XSLT: Si il y a un § apr. la cote et avant la description du
 feuillet de garde et/ou de l'ancienne cote (à tester)

<summary>

§ avec folios, citat et tirets longs :

\hookrightarrow Majuscule ou chiffre \longrightarrow tiret long

↑ Regress trop dangereuses

Condit en Python ou XSLT: jouer sur
 les §, les tirets et les n° de
 folios (à tester)

↑ sauf si le tiret long est
 suivi d'une (y calendrier)

<msItem>

\hookrightarrow si dans un même §, les n° de folios
 qui suivent sont compris dans la tranche de folios qui
 précède ($n \leq n'$) \longrightarrow clôturé un <msItem> jusqu'au prochain

triet long suivi d'un n° de folio > au dernier de la tran-
che définie, ou bien = avec v°. Condition XSLT ou Python avec
fonct range pour tranches de folios avec format (regEx) → à tester

↳ [F/fol.] n° (à n°)(v°); ou ; (A les n°)

de folios sont parfois encadrés par des ()

↓
1 faire ap. avoir enco-
di les autres éléments
du <mstItem>
étape ⑥

↳ n° de folio(s) + [titre de la partie . -

REGEX: \-?((à|lat|[<.*]) [0-9] A-21 v°]{1,3}(.)([A-ZÉÄÜ]+ [a-zéèàù]+ (.*)?)[;|\n|\n])
=> pas de parties MAIS cela ne matche
pas les rubriques qui sont officielles de titre

étape ①

↳ [n° de folio(s) + << italique >>]

Comment jouer sur
italique
(seule différence
avec incipit)?

étape ②

<rubric>

↳ [n° de folio(s) + (<<) texte + {::??} -]

étape ③

<incipit> ?

(n° de folios)(...) - texte ... -

étape ④

<quote> ?

↳ [n°... + texte + {::??}]

Excepté qui font que il n'y a pas de différences
mais bien entre les 3 types de italique.
REGEX: ((-?) ((0-9) A-21 v°){1,3}(.)([A-ZÉÄÜ]+ [a-zéèàù]+ (.*)?)[;|\n|\n])
? << ((.*?){3}!|(\.){3})?>>
= >> italique de
matchées x 35 italiques produites

à faire ap. avoir délimité
les <mstItem> pour éviter les erreurs

étape 5

③

explicat ?

étape ①

Ancienne côte
au · ou au

→ d'un / ^{étape ③} Reht du
calendrier de · — à »
avec jours entre ()

Credit XSLT ou Python (parcourt du calendrier)
car regex trop dangereuses (peuvent prendre qu'il
ne faut) : si un titre a la ment "date"
l'heure suivie d'un · à au plus espaces et saut
d'un saut de ligne ; prendre ce qui suit le —
jusqu'au dernier » suivie de fol/chiffre/naj.

REGEX (jeu feuillet de garde + anc côte)
((.-+?) (Feuillet(s)? | Feuillet(s)? + (de garde)?
| anciennels)? | Anciennes(s)? côte)+(.+?)
(\ - [0-9]{1,2}A-Z) MAIS trouver
moyen d'exclure la phrase qui contient
"Parch/Parch"
↳ substitut : \n<note>\$1</note>\n

*

§ suivant ceux sur le contenu : ment de l'usage,
souvent de l'office, et du calendrier

étape ①

origin

date

phrases avec la ment de
→ origDate

étape ②

REGEX : \. \s+ ([A-ZÉÉÀÀÙÙ]) (.+?) date (.+?) \.
=> risque de ne pas tout matcher
↳ substitut : \n<origDate>\$1</origDate>\n
Credit Python ou XSLT : matcher phrases contenant
vb "date".

Saut de ligne (double) puis Parch./Parch.

perches REGEX : \n^ (Parch\.\. | Parch\.\.), => pas de
↳ substitut : \n<material>\$1</material>\n

material

nbre de folios

nbre de col.)

jusqu'au · (—)

REGEX : ([0-9]+ col\.-)

=> pas de perches

↳ substitut : ([0-9]+ col\.-)

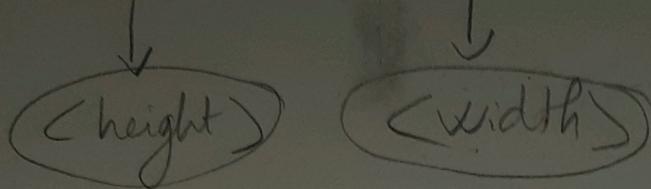
layout

measure

REGEX : \n^ (Parch\.\. | Parch\.\.), \s
([0-9]+ \s) \. (.+?) (col\.-)? (.+?) \.
=> pas de perches
↳ substitut : \n<measure>\$2</measure>\n

chiffre sur chiffre mill.

④



REGEX: $([0-9]+)$ sur $([0-9]+)$ mill
=> pas de fentes
↳ substitut: \n< height > \$1< height >
\n< width > \$2< width >

— phrase) ou § avec ment de "peinture",
"décorat", "initiales", "miniatures" → entre
les dimens° du manuscrit et la reliure.
+ Gedit Python / XSLT (regressions trop dangereuses): à drag notice, matcher ce qui est
entre mill. - et Rel. / Rel. / Demi-reliure
étape ①

< decoNote >

étape ② => si "usage" et/ou "possesseurs"

REGEX: \n(.*)? (usage | possesseur) (.*) \n => pas
de fentes pour ce qui a été
matché pour < decoNote >

< origin >

< provenance >

+ Gedit Python ou XSLT pour
extraire la phrase et les suivantes
contenant "possesseur".

< p >

< CPS >

§ commençant par "Rel. / Rel. / Demi-reliure" jusqu'
au -(—)

↓
< binding >
(p)

REGEX: ((Rel. | Rel. | Demi-
reliure)(.+)?)((-() [A-ZÉ
ÈÀÙ])?) => pas de fentes
↳ substitut: \n< p > \$1< p >\n

Si — NON (jusqu'à chiffre ou MAISOULE. ⑤

étape ① REGEX: (-)?((\D)?[A-ZÉÈÀÙ]{2}){2}\} \((.+\?)\)(-|\n) => pas de pertes pour biblio. à la fin de la notice MAIS si dans la notice il y a une séquence de (et de lettres dans la notice ça matche aussi. Ne pas prendre en compte ce qui a déjà été encodé ?

étape ② REGEX: ((\N)?[A-ZÉÈÀÙ]+(.+\?))\.\) +\-\+|\n) => Sépare les réf à partir des titres MAIS ne matche pas la dernière réf.
Corriger les erreurs d'OCK récurrentes à la fin de l'encodage :

- Panil. → Panch.

- Arren → Amen

- Rcl. → Rel.

- ds @n de <TEI> → remplacer "I" et/ou "l" par 1

(*) <note> au sein d'un <msItem> :

étape ④

- fin d'un § après >> -

? Regex trop dangereuse car \n.*>>\s+ - \s+(.*) font matcher la bonne <note> MAIS aussi d'autres fins de § construits de la même manière (citat ou titre d'un <msItem>).

étape ⑤

- § au milieu de 2 autres sans mercé de folios suivis d'un titre ↗ aux sauts de lignes dans un m§ créés par l'OCR

Condition XSLT ou Python : regex (>>\s+\n?\n.*?\n*\(.*\|\n*\.*\)\n fol1 [0-9A-Z]) trop dangereuse car matche ④ qu'il ne faut. Bien matcher un § en entier dans la condition.

étape ②

- Pour la mercé de "L/lacune" entre - et . - /ou \s+ de ligne

REGEX : ((lacune | lacune) entre(.*)?)\.\{1\})(\s+|-)? => pas de pertes

↳ substitut : \n<note>\$1</note>\n

étape ⑥

- Pour la mercé d'une autre main précédée de - ou de . ou de : et suivie de (ou de :

Regex délicate : (\-|([0-9A-Z]\.\.)|([a-zA-Z]\.))\{1\}+((.+)?) une autre main (.+?)) (|(1:)) => Pas de pertes MAIS cela matche parfois trop. Match juste pour ". D'une autre main : "