

Aanwinsten - templates

reStructuredText

Aanwinsten gebruikt reStructuredText als een soort tussenformaat bij het maken van de lijsten. reStructuredText is een soort Wiki-achtige taal. Deze documentatie is ook in reStructuredText geschreven. reStructuredText kan met diverse programma's worden omgezet in andere formaten, zoals HTML en PDF. Aanwinsten gebruikt daarvoor de programma's rst2html en rst2pdf.

Vijf bestanden

De reStructuredText voor een lijst wordt bepaald door vier template-bestanden:

Document

Beschrijft het document

Root section

Beschrijft de hoofdrubriek

Section

Beschrijft een rubriek

Record

Beschrijft een boek

Thesis

Beschrijft een scriptie

Deze templates en de content en de vlaggen die worden meegegeven aan programma's rst2html en rst2pdf bepalen hoe de lijsten eruit komen zien.

Taal is Nederlands of Engels

Aanwinsten is tweetalig, Nederlands en Engels. De template-bestanden zijn dat ook. De taal komt in de naam van het bestand terug ('-en' of '-nl'). Er wordt altijd gezocht op de meest specifieke variant. Daarna wordt gezocht naar een variant zonder de taal.

Doel is HTML of PDF

In bepaalde gevallen moet het template voor PDF anders zijn dan voor HTML. In aanvulling op de taal-selectie is er ook nog selectie op doel=HTML ('-html') of doel=PDF ('-pdf'). Dat werkt met hetzelfde terugvalmodel.

Mogelijke varianten van taal en doel

- (geen) = taal en doel niet gespecificeerd
- -nl = taal Nederlands, doel niet gespecificeerd
- -nl-html = taal Nederlands, doel HTML
- -nl-pdf = taal Nederlands, doel PDF
- -en = taal Engels, doel niet gespecificeerd
- -en-html = taal Engels, doel HTML
- -en-pdf = taal Engels, doel PDF

- -html = taal niet gespecificeerd, doel HTML
- -pdf = taal niet gespecificeerd, doel PDF

Bij opleveren van Aanwinsten zijn er de volgende templates:

- document-en-pdf.template
- document-en.template
- document-pdf.template
- document.template
- root-section.template
- section-pdf.template
- section.template
- record.template
- thesis.template

Varianten -nl en -html komen niet voor omdat dat de terugvalvarianten Nederlands en HTML zijn. Engels en PDF zijn steeds de uitzonderingen.

Template document.template bevat Nederlandse tekst. Voor een Engelstalige lijst worden dus bestanden 'document-en.template', 'section.template', 'record.template' en 'thesis.template' gebruikt. Voor een Nederlandstalige lijst worden bestanden 'document.template', 'section.template', 'record.template' en 'thesis.template' gebruikt. Zou je voor Nederlandstalige- en Engelstalige scripties verschillende vormgeving willen hebben, kopieer dan thesis.template naar thesis-en.template en pas beide bestanden aan.

Variabelen

De templates (mallen) die Aanwinsten gebruikt, zijn in reStructuredText geschreven, waarbij variabelen worden vervangen door waarden. Die variabelen die eruit zien als `{{bla}}` maar maken geen deel uit van de reStructuredText-specificatie zelf. Die variabelen maken deel uit van een ander templating-mechanisme.

De templates

Ik behandel ze van 'onder' naar 'boven'.

Record

Een record.template-bestand kan er als volgt uit zien:

```
| `{{245-a}} <https://tilburguniversity.on.worldcat.org/oclc/{{001}}?↵
|      databaseList=2294,638>`__ / {{100-a}}. - {{260-a}}. - {{300-a}}. - {{490-a}}
| {{500-a}}
| {{020-a}}
| **{{852-h z}}**
```

Er is een lege regel aan het einde vereist omdat alle records achter elkaar met het template worden ingevuld. De lege regel maakt een scheiding tussen verschillende blokken (records).

De pipe-tekens ('|') zorgen er voor dat een groepje regels als een blok wordt beschouwd waarbij nieuwe regels in het blok gehandhaafd worden. De constructie met een backtick op de eerste regel, die eindigt met twee underscores, beschrijft een link in reStructuredText. Tussen '<' en '>' staat de URL van de link en daarvoor de tekst van de link. Tekst ingesloten door twee dubbele asterisken wordt dubbelvet (strong) afgebeeld.

De variabelen, zoals \${500-a}, geven de velden aan in het MARC21-record.

Er zijn meerdere varianten mogelijk, uitgelegd aan de hand van voorbeelden:

- \${123} = veld zonder subvelden
- \${123-a} = subveld 'a' van veld '123' (in MARC vaak aangeduid met '123\$a'; dat dollarteken gaf problemen in de programmatuur hier, vandaar het minteken), eerste voorkomen
- \${123-a+} = subveld 'a' van veld '123' alle voorkomens, gescheiden door puntkomma's ('waarde; waarde; waarde')
- \${123-a b+} = subvelden 'a' en 'b' van veld '123', gescheiden door een spatie, alle voorkomens, gescheiden door puntkomma's ('waardea waardeb; waardea waardeb; waardea waardeb')
- \${123-a[4=aut]+?sep=" . "} = subveld 'a' van veld '123' alle voorkomens waar subveld '4' van datzelfde veld '123' waarde 'aut' heeft, gescheiden door punten (punt vervangt default puntkomma)
- \${123-a[4=edt]+?sep=" . "&pre="ed. by "} = subveld 'a' van veld '123' alle voorkomens waar subveld '4' van datzelfde veld '123' waarde 'edt' heeft, gescheiden door punten, met 'ed. by ' ervoor als waarde niet leeg
- \${123-a[4=aut|edt]} = subveld 'a' van veld '123' alle voorkomens waar subveld '4' van datzelfde veld '123' waarde 'aut' of 'edt' heeft
- \${123-a[4=aut|(null)]} = subveld 'a' van veld '123' alle voorkomens waar subveld '4' van datzelfde veld '123' waarde 'aut' heeft of waar subveld 4 niet bestaat

Te onderscheiden onderdelen van een variabele zijn:

- fieldid = 3-cijferige veld aanduiding van Marc OF "LEADER" als de Leader getoond moet worden
- subfieldid = een string van minimaal lengte 1; elke letter of cijfer duidt een subfield aan (staat direct achter fieldid, gescheiden door een min-teken)
- condfieldid = 1 letter of cijfer dat een subveld aanduidt waarvoor je een conditie wilt opgeven
- condvalues = waarde(n) waaraan condfieldid moet voldoen opdat fieldid\$subfieldid wordt afgebeeld (condfieldid en condvalues staan met een '=' ertussen tussen vierkante haken, direct achter subfieldid en er mag maar 1 zo'n conditie worden opgegeven; er mogen meerdere waarden worden opgegeven, gescheiden door een pipe-teken '|', waar je voor 'niet-bestaand' '(null)' moet gebruiken, zoals in voorbeeld hierboven)
- multi = '+' of afwezig, maar bij '+' alle waarden tonen en scheiden door '; ' (default alleen de eerste tonen) (multi werkt alleen bij subvelden)

Met ?a=b&c=d&e=f kan je nog extra parameters toevoegen. Dat patroon is als bij request-parameters in URLs. Er kunnen de volgende parameters worden opgegeven:

- sep = gebruik deze separator tussen waarden als er meerdere waarden zijn. Dat vervangt de default '; '. Omgeef de waarde eventueel door dubbele quotes; dat hoeft niet per se. Is alleen relevant als '+' is opgegeven.
- pre = plaats deze string voor een waarde indien die niet leeg is. Hiermee kan bijvoorbeeld 'ed. by ' voor de editors worden geplaatst, maar alleen als er ook editors zijn.
- post = plaats deze string achter een waarde indien die niet leeg is. Hiermee kan bijvoorbeeld ' editor(s)' achter de editors worden geplaatst, maar alleen als er ook editors zijn.

Filtering en opschoning

Er vindt enige filtering en opschoning plaats.

Veld 001 bevat het OCN, maar dat OCN begint zelf weer met de letters 'ocn' of 'ocm'. Die worden er afgehaald.

Alle velden worden afgebroken op een maximale lengte van 300 tekens. Is een veldwaarde langer dan 300 tekens, dan worden de eerste 299 getoond, aangevuld met een beletselteken ('...').

Er vindt na het toepassen van de template opschoning plaats. Bijvoorbeeld dubbele spaties vervangen door enkele, of weghalen van een spatie voor een interpunctieteken of verwijderen blocklines zonder inhoud. De hele reeks vervangingen is configureerbaar. Die reeks vervangingen wordt net zo vaak toegepast tot er niks meer verandert. Dat geeft ook risico's. Een vervanging kan een andere vervanging teniet doen. Dan zal het programma niet meer ophouden. Niets beschermt tegen dat probleem. Zie instelling `aanwinsten.cleanups-file` en bestand `cleanup.conf`.

Thesis

Een `thesis.template`-bestand kan er als volgt uit zien:

```
| ${100-a}  
| ${245-a}  
| ${653-a+}  
| `Link <https://tilburguniversity.on.worldcat.org/search?←  
    queryString=scr.uvt.nl+${001}>` _
```

Hier zie je een voorbeeld van het tonen van alle keywords door het plusteken in `${653-a+}`. Ook hier een lege regel aan het einde nodig.

Filteren en opschonen als bij record.

Section

Een `section.template`-bestand kan er als volgt uit zien:

```
${section-id} ${section-title}  
${adornment}  
  
.. container:: top  
    `▲ <#top>` _  
  
${records}  
  
${subsections}
```

Hierin worden de volgende vervangingen gedaan:

`${section-id}` wordt vervangen door het ID van de rubriek.

`${section-title}` wordt vervangen door de titel van de rubriek, afhankelijk van de taal (Nederlands of Engels).

`${adornment}` zorgt voor de juiste onderstreping van de titel. Door verschillende onderstrepingen, zoals '===', '+++', '---' en '~~~' te doen in een stricte volgorde, weet `reStructuredText` de niveau's van secties in het document te onderscheiden. Programma

rst2html wijst op basis daarvan h1 t/m h6 toe aan titels. Aanwinsten zoekt zelf de juiste onderstreping uit.

`{records}` wordt vervangen door een gesorteerde lijst van records (of scripties) die elk al door het `record.template` of `thesis.template` zijn verwerkt.

`{subsections}` wordt vervangen door alle subsecties die door ditzelfde `section.template` zijn verwerkt. Dit maakt het template recursief.

De constructie met `.. container:: top` zorgt ervoor dat wat daarbinnen staat (de link) in een `<div class="container top">` komt in HTML. Dat biedt de mogelijkheid om met CSS te zorgen dat in HTML het pijltje omhoog naar rechts wordt uitgelijnd.

Het anker '#top' vind je terug in het voorbeeld voor `document.template`.

Root section

Een `root-section.template`-bestand kan er als volgt uit zien:

```
{records}

{subsections}
```

Voor de mogelijke vervangingen, zie Section.

Document

Het Document-template beschrijft een hele aanwinstenlijst. Hierin komen alle voorgaande templates samen.

Een `document.template`-bestand kan er als volgt uit zien:

```
.. _top:

Aanwinstenlijst {collection} {year} {month}
{adornment}

Deze aanwinstenlijst geeft een overzicht van nieuwe titels.
Het is mogelijk om via e-mail geattendeerd te worden telkens wanneer er
een nieuwe aanwinstenlijst verschijnt.
Vul hiervoor een `formulier <https://www.tilburguniversity.edu/nl/over/
universiteitsbibliotheek/collectie/aanwinsten/aanmelden.htm>`_ in.

.. contents::

.. footer::
    ###Page###

{sections}

top_
```

Hierin worden de volgende vervangingen gedaan:

`{collection-id}` wordt het collectie-id zoals dat in de configuratie staat.

`{collection-name}` wordt de collectienaam, NL of EN, afhankelijk van de situatie, of

`{collection-id}` wanneer de naam niet beschikbaar is in de configuratie.

`${year}` wordt het jaar, bijvoorbeeld '2017' als in januari 2018 de lijsten van de vorige maand worden gemaakt.

`${month}` wordt de maand als 2 cijfers, 01-12.

`${month-long}` wordt de maand uitgeschreven in de juiste taal, dus 'januari', 'februari', 'maart', ... in het Nederlands en 'January', 'February', 'March', ... in het Engels.

`${adornment}` is al behandeld bij Section.

`${sections}` de secties, verwerkt door het `section.template`.

`${file-basename}` naam van doelbestand zonder de extensie en de taal van de huidige context, dus bijvoorbeeld "CIF-2016-09-en".

`${file-basename-nl}` naam van doelbestand zonder de extensie en taal=nl, dus bijvoorbeeld "CIF-2016-09-nl".

`${file-basename-en}` naam van doelbestand zonder de extensie en taal=en, dus bijvoorbeeld "CIF-2016-09-en".

In dit voorbeeld zie je nog `.. _top:`. Dat leidt tot een id waar de pijltjes-links in de sections naartoe kunnen wijzen:

```
<span id="top"></span>
```

De `top_` onderaan geeft een referentie-link met tekst 'top'. Zonder dat wordt het `` niet gemaakt.

Met `.. contents::` wordt aangegeven dat daar een inhoudsopgave moet komen.

De constructie met `.. footer::` zorgt voor paginanummering in PDF. In HTML werkt dat natuurlijk niet. Daarom komt dat in templates voor HTML niet terug.