

Workflow for Critter of the Week stuff

Relevant links	1
Adding a new Critter of the Week	1
Updating the Listeria page when a new Critter has been added	2
Updating the Page assessment table	2
Critters with Wikipedia pages	3
Critters without Wikipedia pages	5

Relevant links

Listeria table listing all Critters of the Week (update when moved):

<https://en.wikipedia.org/wiki/User%3ADrThneed%2FCritteroftheWeek>

Example Critters showing Wikidata statements:

<https://www.wikidata.org/wiki/Q129026>

<https://www.wikidata.org/wiki/Q163283>

Table of page assessments for Critter of the Week (ie worklist) (update when moved):

<https://en.wikipedia.org/wiki/User:DrThneed/CritteroftheWeekPageAssessments>

Adding a new Critter of the Week

1. Find the relevant Wikidata item for the species (check Latin name and common name)
2. Create statement
 - a. P1343 (described by source): Q58450498 (Critter of the Week) with qualifiers:
 - P577 (publication date) of broadcast date
 - P407 (language of work), English
 - P1810 (named as) name used in the programme
 - P953 (full work available at URL). It's important to have the full URL here rather than just in the reference so it gets included in the Listeria query.
3. Reference:
 - a. P953 (full work available at URL), link to the RNZ programme
 - b. P813 (retrieved date) date you enter the info.
4. Also check the alias includes the common name (or the Latin name if the label is the common name) and whether a P1843 (common name) statement is needed.

Updating the Listeria page when a new Critter has been added

The Listeria table will update automatically but if you want to check how it looks, click the 'Update the list now' link on the right top of the table. This will take you to another page reporting whether the update worked or not, from where you click back to the updated page. If the update fails or the result is not as intended, the Sparql may need updating, check with DrThneed.

The table is currently set to sort by Class. You can sort the table by any of the columns you like, by clicking on the small arrows next to the column title.

The current Sparql query to produce the table (*you do not need to use or change this to run the update*):

```
SELECT ?item ?taxon_common_name ?parent WHERE {  
  ?item wdt:P1343 wd:Q58450498;  
    wdt:P1843 ?taxon_common_name .  
  ?item wdt:P171* ?parent .  
  ?parent wdt:P105 wd:Q37517 .  
FILTER (langMatches( lang(?taxon_common_name), "en" ) )  
}
```

(if you want to try the query outside of Listeriabot, then use one with labels:

```
SELECT ?item ?taxon_common_name ?parent ?parentLabel WHERE {  
  ?item wdt:P1343 wd:Q58450498;  
    wdt:P1843 ?taxon_common_name .  
  ?item wdt:P171* ?parent .  
  ?parent wdt:P105 wd:Q37517 .  
FILTER (langMatches( lang(?taxon_common_name), "en" ) )  
SERVICE wikibase:label { bd:serviceParam wikibase:language "[AUTO_LANGUAGE],en". }  
}
```

And the columns are P18:Image, label:Label,description, ?taxon_common_name:Common name, P225:Taxon, ?parent:Class, P1343/Q58450498/P577:Broadcast date, P1343/Q58450498/P953:Broadcast Url

Updating the Page assessment table

Individual critters can be added to the tables with direct editing on an as-needed basis (these tables are not created using Listeria). The workflow below describes how to reproduce the tables to update the page assessments, and find any critters that have had new English Wikipedia pages made for them since the tables were last updated. The workflow shows how to use a Sparql query to produce a taxon list that is imported into OpenRefine, extract the Wikipedia API for the page assessments, and import the results into Wikipedia.

Critters with Wikipedia pages

Begin with Sparql query to get Wikipedia article link for all Critters of the Week (as defined by having a described by source = critter of the week). Query is here <https://w.wiki/oAr>

```
SELECT ?item ?itemLabel ?sitelink WHERE {  
  ?item wdt:P1343 wd:Q58450498 . #item has described by source = critter of the week  
  ?sitelink schema:about ?item;  
  schema:isPartOf <https://en.wikipedia.org/>.  
  SERVICE wikibase:label {bd:serviceParam wikibase:language "en"}  
}
```

This gets around the fact that some of the Wikipedia links in the original critter table were redirects, and you cannot get a page assessment on a redirect.

Export query as csv file

Import csv file into OpenRefine (N.B. In March 2021 this file had 235 rows, replacing a table with 229 rows)

Just use the item column from the critter list, and reconcile (reconcile against no particular type).

Then add another column from the reconciled values, typing **Senwiki** and selecting **SPARQL: Senwiki** from the dropdown. This returns a list of page titles without replacing characters for apostrophes, macrons and spaces. Call the new column **page title**.

From the page title column, Add column based on fetching URL using function

```
"https://en.wikipedia.org/w/api.php?action=query&titles=" +  
value.escape('url') +  
"&prop=pageassessments&format=json"
```

This takes a while (up to an hour! Go do something else).

Then split the results column multiple times to get the first assessment out or write GREL that works:

Split at "class": maximum columns 2

Split result at ",", maximum columns 2

Cluster or facet results and replace Stub with {{Icon|Stub}} etc

(Note that the little facet box on the left, where you can edit the Stub to {{Icon|Stub}}, gives you a handy little summary of how many of each type there are e.g. in March 2021

{{Icon|Stub}} 56

{{Icon|B}} 5

{{Icon|C}} 67

{{Icon|GA}} 6

{{Icon|Start}} 101

Check there are no blanks (if the first page assessment template on a page is blank, you will get no result, I had one of these so just added the assessment by hand).

In order for the URLs to work in the final table, we need to use the page titles to create a complete URL

Add column based on Sitelink enwiki, using GREL: `"https://en.wikipedia.org/wiki/" + value`

Page titles with multiple words will still have spaces, so use Cells Transform to find and replace space with underscore, just in the URL column.

Export the OpenRefine project to Excel (via CSV file). At this point you need to create a custom list to sort the data by page quality. Under 'Excel', 'Preferences', 'Custom lists' add a new list of {{Icon|Stub}}, {{Icon|Start}}, {{Icon|C}}, {{Icon|B}}, {{Icon|GA}}, {{Icon|A}}, {{Icon|FA}} and then sort the data, with alphabetical by item as the second sort order. (You only need to add the custom sort order the first time you do this).

It could be useful to create a numbered column on the left to keep track of how many items are in the list. At this point you want columns reflecting Item, Wikipedia page link, Wikidata item link, and page assessment. Import the resulting table into Wikipedia using the <https://excel2wiki.toolforge.org/> tool (or similar), checking the 'sortable table' option.

Import csv file into OpenRefine

Split the sitelink column at separator `wiki/`

Rename column sitelink2 as `enWiki page title`.

Add column based on fetching URL using function

```
"https://en.wikipedia.org/w/api.php?action=query&titles="
+value.escape('url')
+"&prop=pageassessments&format=json"
```

This takes a while (maybe an hour? Just leave it alone and do something else!).

Double check that the column has returned a page assessment for all pages (e.g. great white shark might be a problem, if it is just one, edit by hand after the split step).

Then split the results column multiple times to get the first assessment out or write GREL that works:

Split at `"class":` (you may need to type not copy this), maximum columns 2

Split result at `"`, maximum columns two

Cluster or facet results and replace Stub with {{Icon|Stub}} etc

Check there are no blanks (if the first page assessment template on a page is blank, you will get no result, I had one of these so just added the assessment by hand)

Export results to Excel (via CSV file). The first time you need to create a custom list to sort the data by page quality. Under 'Excel', 'Preferences', 'Custom lists' add a new list of

{{Icon|Stub}}, {{Icon|Start}}, {{Icon|C}}, {{Icon|B}}, {{Icon|GA}}, {{Icon|A}}, {{Icon|FA}} and then sort the data by the page assessment.

Import the resulting table into Wikipedia using the <https://excel2wiki.toolforge.org/> tool (or similar), checking the 'sortable table' option.

Critters without Wikipedia pages

This Sparql query finds items with a Critter of the Week statement but no English Wikipedia page:

<https://w.wiki/oB4> (can add taxon common name but it doesn't take notice of 'distinct' and therefore increases the number of results because some things have several common names)

```
SELECT DISTINCT ?item ?itemLabel WHERE {  
  ?item wdt:P1343 wd:Q58450498.  
  SERVICE wikibase:label { bd:serviceParam wikibase:language "en". }  
  FILTER(NOT EXISTS {  
    ?sitelink schema:about ?item;  
    schema:isPartOf <https://en.wikipedia.org/>.  
  })  
# OPTIONAL { ?item wdt:P1843 ?taxon_common_name. } gives more than one result per item  
}
```

Export results to Excel (via CSV file) and import into Wikipedia using the <https://excel2wiki.toolforge.org/> tool (or similar). (NB was 18 items prior, now 21 results as of March 2021)