EC Crawler
Documentation

Single Digital Gateway
Table of Contents

I. Goal ............................................................................................................................................. 2
   1.1. Problem looking to resolve ................................................................................................. 2

II. EC crawling process and the metadata ................................................................................... 3
   2.1. Notifying a link for Crawling ............................................................................................. 4
   2.2. Crawling process ................................................................................................................ 5
   2.3. Metadata ............................................................................................................................ 5
       2.2.1. Canonical tag ............................................................................................................... 8

III. Crawler statistics ..................................................................................................................... 10
   3.1. Problem looking to resolve ............................................................................................... 10
       3.1.1 Usability ....................................................................................................................... 10
       3.1.2. Administrative ............................................................................................................ 11
   3.2. List of Crawled Events ........................................................................................................ 12
   3.3. Crawler Event Statistics .................................................................................................... 14
I. Goal

The present document has the purpose to explain the **Crawler functionality** currently implemented in the SDG links repository and presents the Crawler’s features and the **Crawler Statistics** module that oversees its constant run and would allow to achieve the expected benefits.

1.1. Problem looking to resolve

The first months of experience with the SDG links repository have shown some drawbacks and limitations on the way links can be added into the system. In particular, Member States that had many links to notify needed to make a great effort in adding all the data in the system. Several proposals have been raised to tackle this, like offering the possibility to import the links into the system:

- By means of file import functionality;
- Having a web service that can ingest links directly from Member States’ IT systems.

There was a need for more, and by adding a **Crawler functionality** to the existing links repository we gave Member States the possibility to only notify the web folders containing the links that needed to be added into the system. The Crawler will be in charge of adding/updating/deleting the links from the notified web folders.
Below is a high-level representation of Crawler interaction with Member states websites:

II. EC crawling process and the metadata

In order for the EC crawler to be able to get the required information from the pages that had been notified by the Member States, a number of meta-tags will need to be present in the generated html code of the web pages. We tried to use Dublin Core (https://www.dublincore.org/specifications/dublin-core/dcmi-terms/) standard elements whenever suitable.

The crawler is an automated system that takes notified web folders and searches through all the child links’ for the ones that contain all the required metadata.

In order to cover all the technologies used to generate websites we have developed two different crawlers:

- the default Crawler which is in charge of all the static web sites
- the JavaScript Crawler for web sites that are generated dynamically using JavaScript.

Note: The user agent for the SDG crawler is EC-SDG-Crawler.
The crawler **runs continuously** in order to keep the list of metadata links (Web pages) up to date.

* The JavaScript crawler needs manual intervention in order to start. We are trying to run it at regular intervals but if you need updates to your datasets please send us an email and we will run the crawler for your dataset.

### 2.1. Notifying a link for Crawling

Only Web Folder (parent links) can be notified for crawling. This can be done when a link is added in the repository or when an existing link is edited.

There are two types of Crawlers available, depending on your websites' technical platform. You can choose the appropriate crawler from the Add/Edit link interface:

- If website contains **static pages**, then the best solution would be to select **“Should this URL be crawled?”**
- If website is built on top of a javascript framework like Angular, React, VueJS, etc., then the best solution would be to select **“Should this URL be crawled? (JavaScript Crawler)”**

*See image below:*
Once the websites have been notified, the Crawler functionality will pick them up on the next scheduled Crawl event and details on what was crawled can be seen in Crawler Statistics module.

### 2.2. Crawling process

1) Retrieves the list of all the active countries in the system
2) Iterates on the retrieved dataset and start with the first member state (e.g. Austria)
3) Retrieves for each member state all the notified links marked for crawling
4) Notified links are being crawled **one by one** looking for web pages that contain the following attributes:

- Language (lang / meta.language)
- SDG tag (sdg-tag)
- Country Code (DC.ISO3166)
- Sub-national location (DC.Location)
- Content type (DC.Service)
- Classification information code (policy-code)
- Classification information description (DC.Policy)
- Canonical tag (rel=canonical)

5) After the crawling process has been completed for a notified link, all the valid web pages that have been collected will be stored in the links repository alongside with the collected metadata.

### 2.3. Metadata

In order for the EC crawler to be able to get the required information from the pages that had been notified by the Member States, a number of meta-tags will need to be present in the generated html code of the web pages.

Below you can find the list of the available **meta-tags**:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Attribute name</th>
<th>Mandatory?</th>
<th>Details</th>
<th>Controlled dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td>Language</td>
<td></td>
<td>Yes (either solution is)</td>
<td>It can be found in the following structures:</td>
<td></td>
</tr>
<tr>
<td>Field</td>
<td>Example</td>
<td>Accepted</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>-------------------</td>
<td>------------------</td>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>lang</td>
<td>lang=&quot;en&quot;&lt;html&gt;</td>
<td></td>
<td>For the Member States that have properly formatted websites in the metadata section</td>
<td></td>
</tr>
<tr>
<td>meta.language</td>
<td>&lt;meta name=&quot;DC.language&quot; content=&quot;en&quot; /&gt;</td>
<td></td>
<td>The sdg-tag will contain the text sdg identifying the page as part of the Single Digital Gateway</td>
<td></td>
</tr>
<tr>
<td>sdg tag</td>
<td>sdg-tag</td>
<td>Yes</td>
<td>It will contain the two characters ISO 3166-1 representation of names of countries.</td>
<td></td>
</tr>
<tr>
<td>Country Code</td>
<td>DC.ISO3166</td>
<td>No</td>
<td>The EEA countries</td>
<td></td>
</tr>
<tr>
<td>Sub-national location</td>
<td>DC.Location</td>
<td>only if applicable</td>
<td>Describes the NUTS or LAU location id for which the content on the page is valid.</td>
<td></td>
</tr>
</tbody>
</table>

The NUTS and LAU location ids

The complete NUTS 1-3 and LAU datasets in excel format: SDG-nuts-1-3.xlsx
EU-28-LAU-2019-NUTS-2016.xlsx

The NUTS 1-3 and LAU datasets csv format (to be uploaded soon)

https://dublincore.org/specifications/dublin-core/dcmi-terms/#Location
<table>
<thead>
<tr>
<th>Content type</th>
<th>DC.Service</th>
<th>Yes</th>
<th>It will contain the information about the type of content present on the page <a href="https://dublincore.org/specifications/dublin-core/dcmi-terms/#Service">https://dublincore.org/specifications/dublin-core/dcmi-terms/#Service</a></th>
</tr>
</thead>
</table>
| Classification information code | policy-code | Yes | It will contain the information about the code of the content area covered by the page according to Annex I and II.  
**Only lowest level categories are accepted** (the codes containing the letter of the category and the number of the area) |
| | | | It can be found in the following structure:  
<meta name="policy-code" content="C2"/>  
In case the content covers multiple content areas we can put multiple codes in the content attribute of the content-area-code meta tag separated by ";" |
| Classification information description | DC.Policy | No, but recommended | It will contain the information about the name of the content area covered by the page according to Annex I and II [https://dublincore.org/specifications/dublin-core/dcmi-terms/#Policy](https://dublincore.org/specifications/dublin-core/dcmi-terms/#Policy) |
| | | | It can be found in the following structure:  
<meta name="DC.Policy" content="acquiring and renewing a driving licence"/> |
| Canonical tag | rel=canonical | only if | The canonical tag is recommended to be[^1] |
*if one of the tags needs to contain more than one value than these values need to be separated by a semicolon sign (;).

For example:

```html
<meta name="DC.Service" content="information;procedure"/>
```

In order for the crawler to be able to find the right meta tags and extract the information from the web pages the DC prefix needs to be used whenever the meta tag name is part of the Dublin Core elements.

The servers that host the websites notified in the Links repository for crawling will need to allow access to the crawler.

### 2.2.1. Canonical tag

The canonical tag is recommended to be used on the web pages that will need to be crawled to make sure we are not registering duplicated pages. For example, on Your Europe we are using filters to display the relevant contact points depending from which section the users arrive to the contact points pages. https://europa.eu/youreurope/citizens/national-contact-points/ireland/index_en.htm?contacts=65891 and https://europa.eu/youreurope/citizens/national-contact-points/ireland/index_en.htm?contacts=65641 are both instances of the original page https://europa.eu/youreurope/citizens/national-contact-points/ireland/index_en.htm but
display filtered content depending on the users' journey towards the contact points page. The html content of all three pages is identical although we have here three unique URLs.

In order for the commercial search engine to understand which page needs to be indexed, we are using the canonical tag with the value https://europa.eu/youreurope/citizens/national-contact-points/ireland/index_en.htm. The other two pages are just different presentations of the same content.

The EC crawler will use the canonical URL in a similar fashion to exclude saving in the repository duplicated pages.

- **Tagging Example 1**

  If we consider the Belgian portal https://www.belgium.be and we want to tag the information about acquiring and renewing your driving licence (C2 in Annex I) the tagging will look like the example below. For this theoretical exercise, we assumed that the information on the page only applies for the Brussels region (BE1).

  Web Page: https://www.belgium.be/fr/mobilite/permis_de_conduire/obtenir_un_nouveau_permis

  ```html
  <html lang="fr"/>
  <meta name="sdg-tag" content="sdg"/>
  <meta name="DC.ISO3166" content="BE"/>
  <meta name="DC.Location" content="BE1"/>
  <meta name="DC.Service" content="information"/>
  <meta name="policy-code" content="C2"/>
  <meta name="DC.Policy" content="acquiring and renewing a driving licence"/>
  ```

- **Tagging Example 2**

  If we consider the Belgian portal https://www.belgium.be and we want to tag the information and the procedure about opening a business (J1 in Annex I and W1 in Annex II) the tagging will look like the example below. For this theoretical exercise, we assumed that the information on
the page only applies for the Brussels region (BE1) and on the page, we have information and the procedure.

Web Page: https://www.belgium.be/fr/economie/entreprise/creation/etapes_principales

<html lang="fr"/>
<meta name="sdg-tag" content="sdg"/>
<meta name="DC.ISO3166" content="BE"/>
<meta name="DC.Location" content="BE1"/>
<meta name="DC.Service" content="information;procedure"/>
<meta name="policy-code" content="J1;W1"/>
<meta name="DC.Policy" content="registering, changing the legal form of or closing a business (registration procedures and legal forms for carrying out business);General registration of business activity, excluding procedures concerning the constitution of companies or firms within the meaning of the second paragraph of Article 54 TFEU"/>

*please note that there may not be an exact match between the example above and the specific situations in your country. Therefore, the instance illustrated above is only for exemplification purposes.

III. Crawler statistics

3.1. Problem looking to resolve

The Crawler functionality is a service that runs alongside the Repository of Links and constantly ingest links, without having a tool to have an overall picture of the crawling process.

Several problems have been identified:

3.1.1 Usability

- Statistics about the crawler activity could be gathered only from logs
• Issues and problems were again collected from logs
• Member states had no access to crawler stats or information

3.1.2. Administrative

• Not knowing when crawler was running or blocked
• No control over the crawler lifecycle

The solution was to implement an interface that would tackle all the issues and concerns raised and would give the application user more control over this important functionality.
**Crawler Statistics module** is located on the left navigation bar by expanding **Statistics** module and after clicking on ‘**Crawler Statistics’ icon.**

See image below:

![Image of Crawler Statistics module](image)

This module is available for all national users as follows:

- **National Coordinators / National Observers** - have access to all the information specific for their Member state.
- **National Service Providers** - have access to information on the links for which they are responsible.

### 3.2. List of Crawled Events

After clicking on ‘Crawler statistics’ icon, the user will see the ‘List of Crawled Events’ page, where all present and past crawl events are listed, for each of them some information being available in a table format:
**Crawler ID** - which is the unique ID assigned to each event.

**Status** – represents the status of the crawler event, which can be:
- ‘In Progress’
- ‘Completed’
- ‘Interrupted’

**Note:** The last status ‘Interrupted’ occurs very rarely, but what it means is that for some reasons the Crawler event has been interrupted for a short amount of time.

**Note:** If value ‘Pending’ is present per event then the data is currently being processed and there is no available info yet.

**Time duration** - which tells us how much time this event took.

**Last update** - the time when the information was last updated for the crawl event.

See image below:

![List of Crawled Events]

**Note:** JavaScript events are highlighted in blue while the crawler ID has a dedicated pattern containing JS label at the end.

**Example below:**

![Example Table]
In the left corner of the list of items page, the user has also the possibility to filter crawl events by a specific interval of time:

*See image below:*

![List of Crawled Events](image)

### 3.3. Crawler Event Statistics

If the user selects a crawl event by clicking the View icon, it will then be redirected to the ‘Crawler Event Statistics’ page, which is split into three tabs:

- General
- Critical issues
- Medium issues

**Note:** In order to have the latest information about the Crawler it is recommended to select one of the recent ones.

- **General tab**

On the ‘General tab’ all notified links that have been marked for Crawl will be listed, each of them coming with some extra information attached such as:

- **Pages Crawled** - which shows the total amount of metadata links visited by the Crawler for that notified link.

- **Pages with metadata** - where two information are displayed, the first being the ‘Total amount of metadata links found’ while the other is the ‘Percentage of metadata links found’ out of the total visited links.

- **Time duration** - which represents the total time it took for the notified link to be crawled.

- **Last update** – the time when the information was last updated for this notified link
**Actions** - there is a log that can be downloaded where the user can see the order and path the Crawler took on the website in order to find metadata links.

*See image below:*

![Crawler Event Statistics](image)

At the bottom of the table, highlighted in blue, the total values for each column are displayed.

➢ **Critical issues tab**

This tab contains a list with all the notified websites that are considered as having critical issues, highlighted with red.

The criteria for a link to be listed in this tab are as follows:

- If a Website has less than two pages visited;
- If a website takes more than 3h 30 min to crawl;
- If the number of saved pages with metadata is less than 10% from the total crawled pages.

Critical information table contains:

- **URL** – notified link marked for JS or regular crawling;
- **Crawled?** – evaluating if the link has been crawled or not.

Possible options:

- Yes (if we have more than 2 pages visited) or,
• No (having less than 2 pages visited);

**Pages with Metadata** – the total number of metadata links found for the notified parent link;

**Time Duration** – the total time it took to crawl the notified parent website;

**Actions** – the user can download the log file used to debug the crawling process for a parent website.

*See image below:*

<table>
<thead>
<tr>
<th>Crawler Event Statistics</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>URL</strong></td>
</tr>
<tr>
<td><a href="https://europeanddirect.aspx">https://europeanddirect.aspx</a></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><a href="https://europeanddirect.aspx">https://europeanddirect.aspx</a></td>
</tr>
<tr>
<td><a href="https://europeanddirect.aspx">https://europeanddirect.aspx</a></td>
</tr>
<tr>
<td><a href="https://europeanddirect.aspx">https://europeanddirect.aspx</a></td>
</tr>
</tbody>
</table>

➢ **Medium issues tab**

‘Medium issues tab’ - contains a list with all notified websites that considered as having minor issues highlighted with **orange**.

The criteria for a link to be listed here are as follows:

⇒ If a Website takes more than **2h** to be crawled.
⇒ If the number of saved pages with metadata is less than **30%** from the total crawled pages

The information presented in the Medium issues tab is:

- **URL** – notified link marked for JS or regular crawling
- **Pages with Metadata** – the total number of metadata links found for the notified parent link
**Time Duration** – the total time it took to crawl the notified parent website

**Actions** – the user can download the log file used to debug the crawling process for a parent website.

*See image below:*