

Metadata for URLs that are part of the Single Digital Gateway

The metadata information requirements present in the implementing act for the web pages notified in the scope of the SDG can be transferred to the EC in several ways:

Metadata to be collected by the EC crawler

In order for the EC crawler to be able to get the required information from the pages that were 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. Below you can find the list of the meta tags that will need to be implemented:

Meta tag name	Mandatory?	Description	Controlled dataset
sdg-tag	yes	The sdg-tag will contain the text sdg identifying the page as part of the Single Digital Gateway	{sdg}
DC. ISO3166	yes	It will contain the two characters ISO 3166-1 representation of names of countries.	The EEA countries
DC. Location	only if applicable	Describes the NUTS or LAU location id for which the content on the page is valid (http://purl.org/dc/terms/Location)	The NUTS and LAU location ids The complete NUTS 1-3 and LAU datasets in excel format EU-28-LAU-2019-NUTS-2016.xlsx SDG-nuts-1-3.xlsx The NUTS 1-3 and LAU datasets csv format (to be uploaded soon)
DC. Service	yes	It will contain the information about the type of content present on the page (http://purl.org/dc/dcmitype/Service)	{information, procedure, assistance}
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 or the full name of the assistance service in Annex III	Annex I and II area codes annex I and II. xlsx Annex III service names (to be uploaded)

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 (http://purl.org/dc/terms/Policy)	Annex I and II area names annex I and II. xlsx annex I and II.csv
---------------	------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------

*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:

```
<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 where the websites that were notified in the Links repository for crawling are hosted will need to allow access to the crawler.

Language

The information about the **language of the page** is a special case because although the best practices state that it should be put in the html tag as the value of the lang property (eg. `<html lang="en">`), we will also cater for the situation where because of CMS's restriction the required information cannot be put there. In such cases, the language information can be placed in a meta tag with the name "DC.Language" and the value property containing the ISO 639-1 Code of the language. For example:

```
<meta name="DC.Language" content="en"/>
```

Providing the information about the language of the page is mandatory.

Canonical tag

The canonical tag is recommended to be used in 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 3 pages is identical although we have here 3 unique urls.

In order for the commercial search engine not to penalize Your Europe for duplicated content we are using the canonical tag with the value https://europa.eu/youreurope/citizens/national-contact-points/ireland/index_en.htm which is the original url to let search engine know that the 2 page pointing to contact points are just duplication of an existing page and they should not be indexed,

The EC crawler will use the canonical url in a similar fashion to exclude duplicated pages from the registration into the system.

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 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"/>
```

NB 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.

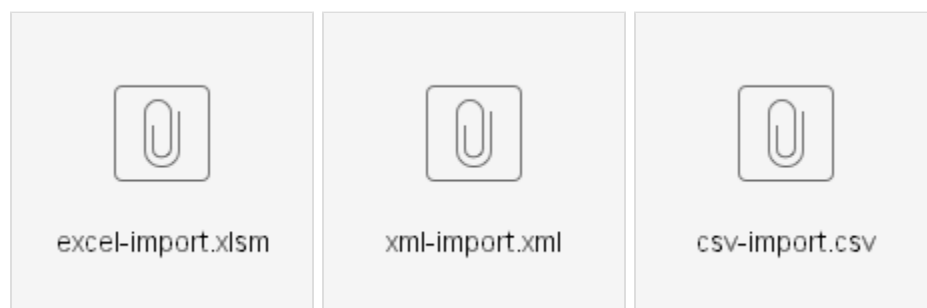
Manual insert into the links repository UI

Information will follow soon

Excel/csv/xml file upload functionality in the links repository

****This option is available in Acceptance for testing but is currently still being refined based on the feedback received from the people testing the feature so small changes might occur before it's released in production. All the updates will be posted here****

Users can upload using an excel/csv/xml file websites/web folders or web pages with their metadata in the links repository interface. A web folder can be an entire website eg: <https://gov.eu> or a sub folder of that website eg: <https://gov.eu/information>. For this process the following files can be used:



The excel file contains the following columns that will need to be completed by the Member States:

title - the title you want to give to your website or web page you want to notify

url - the url of the web folder or web page you want to notify

description - the title you want to give to your website or web page you want to notify

type- the type of information present in the content which can be { Information or Procedure } . There is already a dropdown present that will give you the possible options for this column.

categories- the areas in Annex I or II that are covered by the content of the notified web folder or web page. This column can have multiple options selected and there is a macro that will allow you to select the areas in the two annexes that you consider necessary.

language - in case you are notifying a web page and its metadata this column will present the language of the web page

url type - this column will specify if the notified url is a web folder or an individual web page. There is already a dropdown present that will give you the possible options for this column.

national code - here you can specify NUTS or LAU location id for which the content on the page is valid if this scenario is applicable for the notified url

parent url - in this column you should specify the url(web folder) where a notified web page and its metadata is located in case that url(web folder) has been previously notified to the Repository of links. If you are notifying web folders or stand alone pages that are not attached to any prenotified web folder no value needs to be provided.

excluded paths - here you can add for web folders that you want excluded from the search results for your notified urls. So for example if you have notified <https://gov.eu> and you don't want pages from the folder news to appear in the search results you can add here <https://gov.eu/news/>. This field is optional.

ignore params - in order to help the crawler, in case you have inserted the metadata information directly into your HTML pages, you can add here the parameters that if found in certain urls will alert the crawler to ignore those pages. For example if your news pages contain a parameter **articleId** you can add it here and everytime a url contains this parameter will be ignored by the crawler. This field is optional.

delete - in this column you can mark the items that you want to be deleted from the links repository. **n** for **No** and **y** for **Yes**.

Similarly, you can complete the csv or the xml files following the rules described above.

Using the web services of the links repository

****This option is available in Acceptance for testing but is currently still being refined based on the feedback received from the people testing the feature so small changes might occur before it's released in production. All the updates will be posted here****

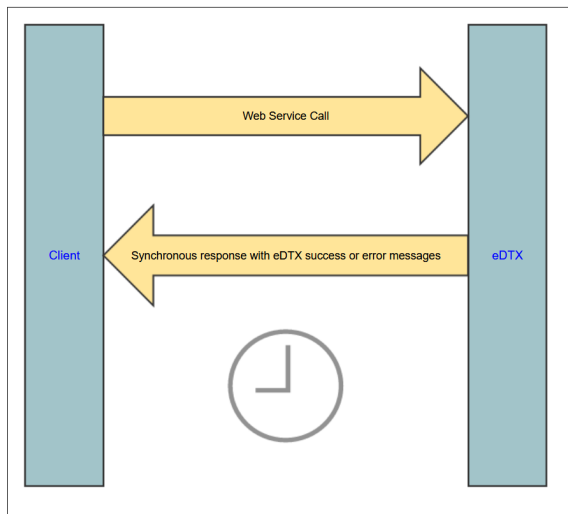
1 Introduction

This section describes how to submit an update request eDTX SDG internal service via a Web Service call. This is a typical B2B integration scenario, where a client application needs to send an automated update request and receive the result. The interaction with the eDTX SDG Web Service is asynchronous. This means that the client sends a translation request and is notified back at a later point in time when the text snippet or document is translated. Thus, calling the web service does not block the client. After update is completed, eDTX sends the update result by a notification.

The following steps illustrate this interaction:

1. The client sends a create or update request to the eDTX web service;
2. eDTX web service will update or create the specified links after the validation;
3. eDTX web service will send a confirmation message of success or an error describing the invalid fields.

The above approach decouples the client from the eDTX server. It's the responsibility of the Client to submit the initial request and react to the callback notification from the eDTX server.



2 Accessing the web service

To obtain the credentials needed to access the service you will need to request them from [G ROW-SINGLE-DIGITAL-GATEWAY@ec.europa.eu](mailto:ROW-SINGLE-DIGITAL-GATEWAY@ec.europa.eu).

3 WSEndPointHandlerService

The service is accessible via the following URL : <https://webgate.ec.europa.eu/youreurope/sdg/public/dtx/import>



It is a good idea to carry out some initial tests of service requests using the SOAP UI tool available here: <https://www.soapui.org/>

If you are unfamiliar with this tool, see <https://www.soapui.org/soap-and-wsdl/getting-started.html> for basic instructions.

The structure of the XML message to send a request is below. Sample translation request for a text snippet:

```

        <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap
/envelope/"
                                xmlns:imp="http://ec.europa.eu/grow/sdg/links
/dtx/soap/import">
    <soapenv:Header/>
    <soapenv:Body>
        <imp:SetImportLinksDetailsRequest>
            <imp:secureToken>CLIENT_SECURE_CODE</imp:secureToken>
            <imp:countryCode>CLIENT_COUNTRY_CODE</imp:countryCode>
            <imp:author>CLIENT_AUTHOR</imp:author>
            <imp:links>
                <!--Zero or more repetitions:-->
                <imp:LinkDetails>
                    <imp:title>LINK_TITLE</imp:title>
                    <imp:url>LINK_URL</imp:url>
                    <!-- Optional -->
                    <imp:description>LINK_DESCRIPTION</imp:description>
                    <!-- Link type can be Information
or Procedure -->
                    <imp:type>LINK_TYPE</imp:type>
                    <imp:categories>
                        <!--Zero or more repetitions:-->
                        <imp:LinkCategoryDetails>
                            <imp:category>LINK_CATEGORY</imp:category>
                            </imp:LinkCategoryDetails>
                        </imp:categories>
                        <!-- Mandatory for URL_TYPE that
has the value WEB PAGE -->
                        <imp:language>LANGUAGE</imp:
language>
                        <imp:urlType>URL_TYPE</imp:urlType>
                        <!-- Optional -->
                        <imp:nationalCode>NATIONAL_CODE<
/imp:nationalCode>
                        <!-- Optional -->
                        <imp:parentUrl>PARENT_URL</imp:
parentUrl>
                        <imp:excludedPaths>
                            <!--Zero or more repetitions:-->
                            <imp:LinkExcludedPaths>
                                <imp:excludedPath>EXCLUDED_PATH</imp:
excludedPath>
                                </imp:LinkExcludedPaths>
                            </imp:excludedPaths>
                            <imp:ignoreParams>
                                <!--Zero or more repetitions:-->
                                <imp:LinkIgnoreParams>
                                    <imp:ignoreParam>IGNORE_PARAM</imp:
ignoreParam>
                                    </imp:LinkIgnoreParams>
                                </imp:ignoreParams>
                                <imp:delete>DELETE</imp:delete>
                                </imp:LinkDetails>
                            </imp:links>
                        </imp:SetImportLinksDetailsRequest>
                    </soapenv:Body>
                </soapenv:Envelope>

```

- **SECURE_CORE** : this will be shared initially on the wiki
- **COUNTRY_CODE** : author need to be authorized to upload links for this specific country
- **CLIENT_AUTHOR** : will be the author of the links that are to be imported through the web service
- **LINK_DESCRIPTION** : is optional field

- **LINK_TYPE** : can have the value **Information** or **Procedure**
- **LINK_CATEGORY** : are category codes described in the ANNEX 1
- **LANGUAGE** : language of the web page
- **URL_TYPE** : the type of the notified URL. Possible values **WEB PAGE** or **WEB FOLDER**
- **NATIONAL_CODE** : NUTS or LAU location id for which the content on the page is valid
- **PARENT_URL** : the URL of the parent of a notified web page
- **EXCLUDED_PATH** : web folders that you want excluded from the search results for your notified urls. So for example if you have notified <https://gov.eu> and you don't want pages from the folder news to appear in the search results you can add here <https://gov.eu/news/>.
- **IGNORE_PARAM** : in order to help the crawler, in case you have inserted the metadata information directly into your HTML pages, you can add here the parameters that if found in certain urls will alert the crawler to ignore those pages. For example if your news pages contain a parameter **articleId** you can add it here and everytime a url contains this parameter will be ignored by the crawler. This field is optional.
- **DELETE** : marks the items that you want to be deleted from the links repository. **n** for **No** and **y** for **Yes**

Sample for update request :

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
                    xmlns:imp="http://ec.europa.eu/grow/sdg/links/dtx/soap"
/soapenv:Envelope>
  <soapenv:Header/>
  <soapenv:Body>
    <imp:SetImportLinksDetailsRequest>
      <imp:secureToken>SECURE_CODE</imp:secureToken>
      <imp:countryCode>BE</imp:countryCode>
      <imp:author>CLIENT_AUTHOR</imp:author>
      <imp:links>
        <!--Zero or more repetitions-->
        <imp:LinkDetails>
          <imp:title>Visa for Belgium</imp:title>
          <imp:url>https://diplomatie.belgium.be/en/services
/travel_to_belgium/visa_for_belgium</imp:url>
          <imp:description>List of Belgian embassies and
consulates competent for visa application</imp:description>
          <imp:type>Information</imp:type>
          <imp:categories>
            <!--Zero or more repetitions-->
            <imp:LinkCategoryDetails>
              <imp:category>D1</imp:
category>
              </imp:LinkCategoryDetails>
            <imp:LinkCategoryDetails>
              <imp:category>D2</imp:
category>
              </imp:LinkCategoryDetails>
            <imp:LinkCategoryDetails>
              <imp:category>D5</imp:
category>
              </imp:LinkCategoryDetails>
            </imp:categories>
          <imp:language>en</imp:language>
          <imp:urlType>WEB FOLDER</imp:
urlType>
          <imp:nationalCode>NATIONAL CODE
SOAP 2</imp:nationalCode>
          <imp:parentUrl></imp:parentUrl>
          <imp:excludedPaths>
```



```

                <!--Zero or more repetitions-->
                <imp:LinkExcludedPaths>
                    <imp:excludedPath>https://eu.com/path3<
/imp:
excludedPath>

                </imp:LinkExcludedPaths>
                <imp:LinkExcludedPaths>
                    <imp:excludedPath>https://eu.com/path4<
/imp:
excludedPath>

                </imp:LinkExcludedPaths>
            </imp:excludedPaths>
                <imp:ignoreParams>
                <!--Zero or more repetitions-->
                <imp:LinkIgnoreParams>
                    <imp:ignoreParam>p3</imp:
ignoreParam>

                </imp:LinkIgnoreParams>
                <imp:LinkIgnoreParams>
                    <imp:ignoreParam>p4</imp:
ignoreParam>

                </imp:LinkIgnoreParams>
            </imp:ignoreParams>
                <imp:delete>n</imp:delete>
        </imp:LinkDetails>
    </imp:links>
</imp:SetImportLinksDetailsRequest>
</soapenv:Body>
</soapenv:Envelope>

```

4 RESTful Secured Web Service

eDTX exposes a RESTful web service available at this address: <https://webgate.ec.europa.eu/youreurope/sdg/public/dtx/import>.

The RESTful service must be called using POST, and the request body must contain the translation request as JSON format.

Sample import request for a text snippet :

```

{
  "secureToken": "SECURE_TOKEN",
  "countryCode": "BE",
  "author": "CLIENT_AUTHOR",
  "linkData":
    [
      {
        "title": "Residence procedure",
        "url": "https://www.belgium.be/en/family/residence_documents_for_belgium",
        "description": "Description of the procedure for people to come to Belgium as foreigners",
        "type": "Information",
        "categories":
          [
            {
              "category": "D1"
            },
            {
              "category": "D2"
            },
            {
              "category": "D4"
            }
          ],
        "language": "en",
        "urlType": "web folder",
        "nationalCode": "NAT CODE REST",
        "parentUrl": "",
        "excludedPaths":
          [
            {
              "excludedPath": "https://eu.com/path1"
            },
            {
              "excludedPath": "https://eu.com/path2"
            }
          ],
        "ignoreParams":
          [
            {
              "ignoreParam": "k1"
            },
            {
              "ignoreParam": "k2"
            }
          ],
        "delete": "n"
      }
    ]
}

```

5 Error messages:

The below list displays all error messages and their description which the client could receive synchronously in case the request is invalid.

Error messages are detailed describing exactly what is missing from the request

```
{
  "status": 400,
  "errors": [
    "Please provide a Title"
  ],
  "errorFields": [
    "linkData[0].title"
  ]
}
```

- **status** : 400 (Bad Request) - status used to notify that there is something wrong in the payload
- **errors** : Description of the issue (e.g "Please provide a Title")
- **errorFields** : Details the location of the encountered issues (e.g. linkData[0] describes which **n'th** link in the payload has an error)