
sphinxcontrib-opendataservices

Open Data Services

May 11, 2021

CONTENTS

- 1 Installation** **3**
- 2 JSON Include Directives** **5**
 - 2.1 Standard JSON Include 5
 - 2.2 Flat JSON Include 6
- 3 JSON Schema Directives** **9**
 - 3.1 CSS 10
- 4 Misc Directives** **11**
 - 4.1 Directives in other repositories 12

Open Data Services' Sphinx directives.

INSTALLATION

Add this line to your `requirements.txt`:

```
-e git+https://github.com/OpenDataServices/sphinxcontrib-opendataservices.  
→git@23ce17656feaa237584af8822bd57ac39b498f93#egg=sphinxcontrib-opendataservices
```

Then run `pip install -r requirements.txt`.

Edit your doc's `conf.py` and add `sphinxcontrib.opendataservices` to the `extensions` array.

e.g.

```
extensions = ['sphinxcontrib.opendataservices']
```


JSON INCLUDE DIRECTIVES

2.1 Standard JSON Include

```
.. rst:directive:: jsoninclude

    .. markdown::

        Include a section of a JSON file, given a [jsonpointer] (https://tools.ietf.org/html/rfc6901).

        e.g. using [this json file](_static/example.json):

    ..

    .. literal-and-parsed-markdown::

        ``eval_rst
        .. jsoninclude:: _static/example.json
           :jsonpointer: /a/0/b
           :expand: e
        ```

 The ``expand`` option is needed to expand a list when the json is folded by javascript. For more info see :ref:`jsoninclude_javascript` below.

 ``exclude`` option:

 .. literal-and-parsed-markdown::

 ``eval_rst
 .. jsoninclude:: _static/example.json
 :jsonpointer: /a/0/b
 :exclude: e,g
        ```

    ``include_only`` option:

    .. literal-and-parsed-markdown::

        ``eval_rst
        .. jsoninclude:: _static/example.json
           :jsonpointer: /a/0/b
```

(continues on next page)

(continued from previous page)

```
        :include_only: e,g
        :expand: e
    ...

.. _jsoninclude_javascript:
```

2.1.1 Javascript

To fold the JSON with javascript, you need to include the following files:

- `renderjson.css`
- `renderjson.js`
- `json-example-format.js`

You need to add the files to a `_static` folder within your docs, and then add the following to `_templates/layout.html`.

```
{% extends "!layout.html" %}
{% set css_files = css_files + ["_static/renderjson.css"] %}
{% set script_files = script_files + ["_static/renderjson.js", "_static/json-example-
↪format.js"] %}
```

The option `expand` can be used to control which lists are expanded initially. The option `title` can be used to give the json include a title; only one of consecutive includes will be shown, with a select box to switch:

```
.. literal-and-parsed-markdown::

    ``eval_rst
    .. jsoninclude:: _static/example.json
       :jsonpointer:
       :title: collapsed

    .. jsoninclude:: _static/example.json
       :jsonpointer:
       :expand: a,b,e,h
       :title: expanded
    ...
```

2.2 Flat JSON Include

```
.. rst:directive:: jsoninclude-flat

    .. markdown::

        Include a section of a JSON file, flattened into a table representation,  

        ↪given a [jsonpointer] (https://tools.ietf.org/html/rfc6901).

        Examples, using [this json file](_static/example.json):
```

(continues on next page)

(continued from previous page)

```

..

.. literal-and-parsed-markdown::

    ``eval_rst
    .. jsoninclude-flat:: _static/example.json
       :jsonpointer: /a/0/b
    ``

``recursive`` (include nested dicts and lists):

.. literal-and-parsed-markdown::

    ``eval_rst
    .. jsoninclude-flat:: _static/example.json
       :jsonpointer: /a/0/b
       :recursive:
    ``

List of items directly under the json pointer:

.. literal-and-parsed-markdown::

    ``eval_rst
    .. jsoninclude-flat:: _static/example.json
       :jsonpointer: /h
    ``

Remove part of the path from the headings uwsing ``ignore_path``:

.. literal-and-parsed-markdown::

    ``eval_rst
    .. jsoninclude-flat:: _static/example.json
       :jsonpointer: /a/0/b
       :ignore_path: /a/0/b/
    ``

``jsoninclude-flat`` also has the options ``exclude`` and ``include_only``, the
same as for ``jsoninclude`` (see above).

```


JSON SCHEMA DIRECTIVES

```
.. rst:directive:: jsonschema

    .. markdown::

        The core `jsonschema` directive, maintained in [the sphinxcontrib-
        ↪jsonschema repository] (https://github.com/OpenDataServices/sphinxcontrib-
        ↪jsonschema).

        The exact output of this is somewhat specific to the OCDS usecases, so
        we subclass the directive to create alternatives for other data
        standards, see below.

    .. literal-and-parsed-markdown::

        ```eval_rst
 .. jsonschema:: _static/example_schema.json
        ```

.. rst:directive:: jsonschema-titles

    Display titles, but not field names.

    Currently only used for 360Giving.

    .. literal-and-parsed-markdown::

        ```eval_rst
 .. jsonschema-titles:: _static/example_schema.json
        ```

    .. literal-and-parsed-markdown::

        ```eval_rst
 .. jsonschema-titles:: _static/example_schema.json
 :child: subthings
        ```

.. rst:directive:: jsonschema-title-fieldname-map

    Display the mapping between titles and field names.

    Currently only used for 360Giving.
```

(continues on next page)

(continued from previous page)

```
.. literal-and-parsed-markdown::

    ``eval_rst
    .. jsonschema-title-fieldname-map:: _static/example_schema.json
    ``

.. rst:directive:: jsonschema-array

    Handle a jsonschema where the top element is an array. Don't display titles
    in the table.

    Currently only used for OpenReferral.

    .. literal-and-parsed-markdown::

        ``eval_rst
        .. jsonschema-array:: _static/example_schema_array.json
        ``
```

3.1 CSS

To display the tables better, add this css to your site: [jsonschema.css](#). You need to add the file to a `_static` folder within your docs, and then add the following to `_templates/layout.html`.

```
{% extends "!layout.html" %}
{% set css_files = css_files + ["_static/jsonschema.css"] %}
```

MISC DIRECTIVES

```
.. rst:directive:: csv-table-no-translate

    .. markdown::

        Like
        [csv-table] (http://docutils.sourceforge.net/docs/ref/rst/directives.html#csv-table),
        but the text inside the table is not translated. Useful when
        translation is handled by an external process, e.g. codelists.

        | ``included_cols`` is the optional list of indices of columns to include.
        |

.. rst:directive:: markdown

    A directive that renders its contents as markdown, using Reccommonmark.

    .. literal-and-parsed-markdown::

        ```eval_rst
 .. markdown::

 Some markdown [a URL] (http://example.org), `single backtick literals`.
        ```

    This is not so useful by itself, but allows markdown to be embedded inside
    other directives, e.g.:

    .. literal-and-parsed-markdown::

        ```eval_rst
 .. admonition:: Worked example
 :class: hint

 .. markdown::

 Some markdown [a URL] (http://example.org), `single backtick_
 literals`.
        ```

.. rst:directive:: directory_list
```

(continues on next page)

```

Return a bullet list for files in a directory.

| ``path`` is the path of the directory.
| ``url`` is a url prefix to form the links
|

Example:

.. literal-and-parsed-markdown::

    ```eval_rst
 .. directory_list::
 :path: example_dir
 :url: https://github.com/OpenDataServices/sphinxcontrib-
↪ opendataservices/blob/master/docs/example_dir/
    ```

.. rst:directive:: localization-note

    Create a note admonition that only will appear in languages that have "translated"
↪ it. This will not appear in the base language (normally English). If a translator
↪ wants to mark they have seen the message but do not want to add a note then they
↪ can leave a single hyphen '-'. The contents of the translation will be treated as
↪ markdown. The text within the directive should contain information useful for the
↪ translator and instruct what to do when they encounter this. For example::

    ```eval_rst
 .. localization-note::

 DO NOT TRANSLATE THIS MESSAGE DIRECTLY

 Instead put some language specific context as to how to interpret this page.

 Put a '-' if you do not want this note to appear in this language.
    ```

```

4.1 Directives in other repositories

- `ocds_sphinx_directives` contains extensions that are specific to OCDS docs sites. Currently they all relate to extensions.
- `OpenReferral's JSON Table Schema include`, because this is the only docs site we maintain that uses JSON Table Schema.