sphinxcontrib.datatemplates Documentation

Release 0.0.0

Doug Hellmann

Jan 16, 2020

Contents

`sphinxcontrib.datatemplates` helps you use static data in machine readable format in your documentation by letting you define Jinja2 templates to turn JSON, YAML, XML, or CSV data into reStructuredText for Sphinx to render as part of its output.

- Repo: https://github.com/sphinxcontrib/sphinxcontrib.datatemplates
- Docs: http://sphinxcontribdatatemplates.readthedocs.io/

CHAPTER 1

Using datatemplate

The `datatemplate` directive is the interface between the data source and the rendering template.

`.. datatemplate:json:: source-path`
: Load file at `source-path` (relative to the documentation build directory) via `json.load()` and render using `template` given in directive body.

`.. datatemplate:yaml:: source-path`
: Load file at `source-path` (relative to the documentation build directory) via PyYAML (`yaml.safe_load`) and render using `template` given in directive body.

`.. datatemplate:xml:: source-path`
: Load file at `source-path` (relative to the documentation build directory) via `xml.etree.ElementTree.parse()` (actually using `defusedxml`) and render using `template` given in directive body.

`.. datatemplate:import-module:: module-name`
: Load module `module-name` (must be importable in `conf.py`) via `importlib.import_module()` and render using `template` given in directive body.

`.. datatemplate:csv:: source-path`
: Load file at `source-path` (relative to the documentation build directory) via `csv.reader()` or `csv.DictReader` depending on `header` and render using `template` given in directive body.

`.. datatemplate:dbm:: source-path::`
: Load DB at `source-path` (relative to the documentation build directory) via `dbm.open()` and render using `template` given in directive body.

1.1 Template Context

When a `datatemplate` directive is processed, the data is passed to the template through its context so that the symbol `data` is available as a global variable.

Important: The data is loaded from the source and passed directly to the template. No pre-processing is done on the data, so the template needs to handle aspects like `None` values and fields that have values that may interfere with

parsing reStructuredText.

1.2 Template Helpers

These helper functions are exposed using their short name (without the module prefix) in the template context.

sphinxcontrib.datatemplates.helpers.**make_list_table**(*headers*, *data*, *title=''*, *columns=None*)

Build a list-table directive.

Parameters

- **headers** – List of header values.
- **data** – Iterable of row data, yielding lists or tuples with rows.
- **title** – Optional text to show as the table title.
- **columns** – Optional widths for the columns.

sphinxcontrib.datatemplates.helpers.**make_list_table_from_mappings**(*headers*, *data*, *title*, *columns=None*)

Build a list-table directive.

Parameters

- **headers** – List of tuples containing header title and key value.
- **data** – Iterable of row data, yielding mappings with rows.
- **title** – Optional text to show as the table title.
- **columns** – Optional widths for the columns.

CHAPTER 2

JSON Samples

2.1 Data File

```
{
    "key1": "value1",
    "key2": [
        "list item 1",
        "list item 2",
        "list item 3"
    ],
    "nested-list": [
        ["a", "b", "c"],
        ["A", "B", "C"]
    ],
    "mapping-series": [
        {"cola": "a", "colb": "b", "colc": "c"},
        {"cola": "A", "colb": "B", "colc": "C"}
    ]
}
```

2.2 Template File

```
.. -*- mode: rst -*-

Static Heading
--------------

Individual Item
~~~~~~~~~~~~~~~

{{ data['key1'] }}
```

(continues on next page)

(continued from previous page)

```
List of Items
~~~~~~~~~~~~~

{% for item in data['key2'] %}
- {{item}}
{% endfor %}

Nested List Table
~~~~~~~~~~~~~~~~~

Rendering a table from a list of nested sequences using hard-coded
headers.

{{ make_list_table(
    ['One', 'Two', 'Three'],
    data['nested-list'],
    title='Table from nested lists',
    ) }}

Mapping Series Table
~~~~~~~~~~~~~~~~~~~~

Rendering a table from a list of nested dictionaries using dynamic
headers.

{{ make_list_table_from_mappings(
    [('One', 'cola'), ('Two', 'colb'), ('Three', 'colc')],
    data['mapping-series'],
    title='Table from series of mappings',
    ) }}
```

2.3 Loading the Template

```
.. datatemplate:json:: sample.json
   :template: sample.tmpl
```

2.4 Rendered Output

2.4.1 Static Heading

Individual Item

value1

List of Items

- list item 1
- list item 2

- list item 3

Nested List Table

Rendering a table from a list of nested sequences using hard-coded headers.

Table 1: Table from nested lists

One	Two	Three
a	b	c
A	B	C

Mapping Series Table

Rendering a table from a list of nested dictionaries using dynamic headers.

Table 2: Table from series of mappings

One	Two	Three
a	b	c
A	B	C

CHAPTER 3

YAML Samples

3.1 Single Document

3.1.1 Data File

```
---
key1: value1
key2:
  - list item 1
  - list item 2
  - list item 3
nested-list:
  - ['a', 'b', 'c']
  - ['A', 'B', 'C']
mapping-series:
  - cola: a
    colb: b
    colc: c
  - cola: A
    colb: B
    colc: C
```

3.1.2 Template File

```
.. -*- mode: rst -*-

Static Heading
--------------

Individual Item
~~~~~~~~~~~~~~~
```

(continues on next page)

(continued from previous page)

```
{{ data['key1'] }}

List of Items
~~~~~~~~~~~~~

{% for item in data['key2'] %}
- {{item}}
{% endfor %}

Nested List Table
~~~~~~~~~~~~~~~~~

Rendering a table from a list of nested sequences using hard-coded
headers.

{{ make_list_table(
    ['One', 'Two', 'Three'],
    data['nested-list'],
    title='Table from nested lists',
    ) }}

Mapping Series Table
~~~~~~~~~~~~~~~~~~~~

Rendering a table from a list of nested dictionaries using dynamic
headers.

{{ make_list_table_from_mappings(
    [('One', 'cola'), ('Two', 'colb'), ('Three', 'colc')],
    data['mapping-series'],
    title='Table from series of mappings',
    ) }}
```

3.1.3 Loading the Template

```
.. datatemplate:yaml:: sample.yaml
   :template: sample.tmpl
```

3.1.4 Rendered Output

Static Heading

Individual Item

value1

List of Items

- list item 1
- list item 2

- list item 3

Nested List Table

Rendering a table from a list of nested sequences using hard-coded headers.

Table 1: Table from nested lists

One	Two	Three
a	b	c
A	B	C

Mapping Series Table

Rendering a table from a list of nested dictionaries using dynamic headers.

Table 2: Table from series of mappings

One	Two	Three
a	b	c
A	B	C

3.2 Multiple Documents

3.2.1 Data File

```
key1: value1

---
key: value
key1: different value
```

3.2.2 Template File

```
.. -*- mode: rst -*-

Static Heading
--------------

Individual Item
~~~~~~~~~~~~~~~

{{ data[0]|tojson }}

List of Items
~~~~~~~~~~~~~
```

(continues on next page)

(continued from previous page)

```
{% for item in data %}
- {{item|tojson}}

  - {{item.key}}
  - {{item.key1}}
{% endfor %}


Mapping Series Table
~~~~~~~~~~~~~~~~~~~~

Rendering a table from a list of nested dictionaries using dynamic
headers.

{{ make_list_table_from_mappings(
    [('Key', 'key'), ('Key One', 'key1')],
    data,
    title='Table from series of mappings',
    ) }}
```

3.2.3 Loading the Template

```
.. datatemplate:yaml:: sample-multiple.yaml
   :template: sample-multiple.tmpl
   :multiple-documents:
```

3.2.4 Rendered Output

Static Heading

Individual Item

{“key1”: “value1”}

List of Items

- {“key1”: “value1”}
  -
  - value1
- {“key”: “value”, “key1”: “different value”}
  - value
  - different value

Mapping Series Table

Rendering a table from a list of nested dictionaries using dynamic headers.

Table 3: Table from series of mappings

Key	Key One
None	value1
value	different value

CHAPTER 4

XML Samples

4.1 Data File

```
<sample>
    <key1>value1</key1>
    <key2>
        <item>list item 1</item>
        <item>list item 2</item>
        <item special='yes'>list item 3</item>
    </key2>
    <mappingseries>
        <mapping>
            <cola special='yes'>a</cola>
            <colb>b</colb>
            <colc>c</colc>
        </mapping>
        <mapping>
            <cola>A</cola>
            <colb special='yes'>B</colb>
            <colc>C</colc>
        </mapping>
    </mappingseries>
</sample>
```

4.2 Template File

```
.. -*- mode: rst -*-

Static Heading
--------------
```

(continues on next page)

(continued from previous page)

```
Individual Item
~~~~~~~~~~~~~~~

{{ data.find('key1').text }}

List of Items
~~~~~~~~~~~~~

{% for item in data.find('key2') %}
- {{item.text}}
{% endfor %}

XPath for Items
~~~~~~~~~~~~~~~

See `XPath support <https://docs.python.org/3/library/xml.etree.elementtree.html
↪#xpath-support>`_

{% for item in data.findall(".//*[@special='yes']") %}
- {{item.text}}
{% endfor %}
```

4.3 Loading the Template

```
.. datatemplate:xml:: sample.xml
   :template: xml-sample.tmpl
```

4.4 Rendered Output

4.4.1 Static Heading

Individual Item

value1

List of Items

- list item 1
- list item 2
- list item 3

XPath for Items

See XPath support

- list item 3
- a

- B

CHAPTER 5

Import-Module Samples

5.1 Template File

```
.. -*- mode: rst -*-

Static Heading
--------------

List of Directory Entries
~~~~~~~~~~~~~~~~~~~~~~~~~

{% for item in data.scandir() %}
- {{item.name}}'s size is {{item.stat().st_size}} Bytes
{% endfor %}

File Path of the Null Device
~~~~~~~~~~~~~~~~~~~~~~~~~~~

``{{data.devnull}}``
```

5.2 Loading the Template

```
.. datatemplate:import-module:: os
   :template: import-module-sample.tmpl
```

5.3 Rendered Output

5.3.1 Static Heading

List of Directory Entries

- dbm.rst's size is 433 Bytes
- sample.xml's size is 483 Bytes
- inline.rst's size is 928 Bytes
- legacy.rst's size is 411 Bytes
- index.rst's size is 765 Bytes
- using.rst's size is 4476 Bytes
- sampledbm.dat's size is 519 Bytes
- yaml.rst's size is 954 Bytes
- csv.rst's size is 532 Bytes
- conf.py's size is 15655 Bytes
- make_dbm.py's size is 114 Bytes
- sample.csv's size is 36 Bytes
- _templates's size is 4096 Bytes
- sampledbm.dir's size is 29 Bytes
- sample-multiple.yaml's size is 53 Bytes
- json.rst's size is 410 Bytes
- _build's size is 4096 Bytes
- sample.yaml's size is 212 Bytes
- import-module.rst's size is 419 Bytes
- xml.rst's size is 416 Bytes
- _static's size is 4096 Bytes
- sample.json's size is 319 Bytes

File Path of the Null Device

```
/dev/null
```

CHAPTER 6

CSV Samples

6.1 Data File

```
a         b         c
Eins         Zwei         Drei
1         2         3
I         II          III
```

6.2 Template File

```
.. -*- mode: rst -*-

Static Heading
--------------

Individual Cell in Row
~~~~~~~~~~~~~~~~~~~~~~

{{ data[0].a }}

List of Cells in Row
~~~~~~~~~~~~~~~~~~~~

{% for item in data[0].items() %}
- {{item[0]}}: {{item[1]}}
{% endfor %}

Mapping Series Table
~~~~~~~~~~~~~~~~~~~~

Rendering a table from a list of nested dictionaries using dynamic
```

(continues on next page)

(continued from previous page)

```
headers.

{{ make_list_table_from_mappings(
    [('One', 'a'), ('Two', 'b'), ('Three', 'c')],
    data,
    title='Table from series of mappings',
    ) }}
```

6.3 Loading the Template

```
.. datatemplate:csv:: sample.csv
    :template: csv-sample.tmpl
    :headers:
    :dialect: excel-tab
```

6.4 Rendered Output

6.4.1 Static Heading

Individual Cell in Row

Eins

List of Cells in Row

- a: Eins
- b: Zwei
- c: Drei

Mapping Series Table

Rendering a table from a list of nested dictionaries using dynamic headers.

Table 1: Table from series of mappings

One	Two	Three
Eins	Zwei	Drei
1	2	3
I	II	III

CHAPTER 7

DBM Samples

7.1 Creating Data File

```
import dbm.dumb

with dbm.dumb.open("sampledbm", "c") as db:
    db[b"Hi"] = b"Hello"
    db[b"Bye"] = b"Goodbye"
```

7.2 Template File

```
.. -*- mode: rst -*-

Static Heading
--------------

Individual Item
~~~~~~~~~~~~~~~

- With decoding {{ data['Hi'].decode('ascii') }}
- Without decoding {{ data['Hi'] }}

List of Items
~~~~~~~~~~~~~

{% for item in data.items() %}
- {{item[0]}} -> {{item[1]}}
{% endfor %}
```

7.3 Loading the Template

```
.. datatemplate:dbm:: sampledbm
   :template: dbm-sample.tmpl
```

7.4 Rendered Output

7.4.1 Static Heading

Individual Item

- With decoding Hello
- Without decoding b'Hello'

List of Items

- b'Hi' -> b'Hello'
- b'Bye' -> b'Goodbye'

CHAPTER 8

Inline Sample (JSON)

8.1 Data File

```
{
    "key1": "value1",
    "key2": [
        "list item 1",
        "list item 2",
        "list item 3"
    ],
    "nested-list": [
        ["a", "b", "c"],
        ["A", "B", "C"]
    ],
    "mapping-series": [
        {"cola": "a", "colb": "b", "colc": "c"},
        {"cola": "A", "colb": "B", "colc": "C"}
    ]
}
```

8.2 Template File

```
Individual Item
~~~~~~~~~~~~~~~

{{ data['key1'] }}

List of Items
~~~~~~~~~~~~~

{% for item in data['key2'] %}
```

(continues on next page)

(continued from previous page)

```
- {{item}}
{% endfor %}
```

8.3 Loading the Template

```
.. datatemplate:json::
   :source: sample.json

   Individual Item
   ~~~~~~~~~~~~~~~

   {{ data['key1'] }}

   List of Items
   ~~~~~~~~~~~~~

   {% for item in data['key2'] %}
   - {{item}}
   {% endfor %}
```

8.4 Rendered Output

8.4.1 Individual Item

value1

8.4.2 List of Items

- list item 1
- list item 2
- list item 3

CHAPTER 9

Legacy Samples

The `datatemplate` directive is should no longer be used. It is deprecated, and will be removed in the next release.

9.1 Data File

```
---
key1: value1
key2:
  - list item 1
  - list item 2
  - list item 3
nested-list:
  - ['a', 'b', 'c']
  - ['A', 'B', 'C']
mapping-series:
  - cola: a
    colb: b
    colc: c
  - cola: A
    colb: B
    colc: C
```

9.2 Template File

```
.. -*- mode: rst -*-

Static Heading
--------------

Individual Item
```

(continues on next page)

(continued from previous page)

```
~~~~~~~~~~~~~~~

{{ data['key1'] }}

List of Items
~~~~~~~~~~~~~

{% for item in data['key2'] %}
- {{item}}
{% endfor %}

Nested List Table
~~~~~~~~~~~~~~~~~

Rendering a table from a list of nested sequences using hard-coded
headers.

{{ make_list_table(
    ['One', 'Two', 'Three'],
    data['nested-list'],
    title='Table from nested lists',
    ) }}

Mapping Series Table
~~~~~~~~~~~~~~~~~~~~

Rendering a table from a list of nested dictionaries using dynamic
headers.

{{ make_list_table_from_mappings(
    [('One', 'cola'), ('Two', 'colb'), ('Three', 'colc')],
    data['mapping-series'],
    title='Table from series of mappings',
    ) }}
```

9.3 Rendered Output

9.3.1 Static Heading

Individual Item

value1

List of Items

- list item 1
- list item 2
- list item 3

Nested List Table

Rendering a table from a list of nested sequences using hard-coded headers.

Table 1: Table from nested lists

One	Two	Three
a	b	c
A	B	C

Mapping Series Table

Rendering a table from a list of nested dictionaries using dynamic headers.

Table 2: Table from series of mappings

One	Two	Three
a	b	c
A	B	C

CHAPTER 10

Indices and tables

- genindex
- modindex
- search

Python Module Index

S

Index

D

M

S