cliff-tablibDocumentation

Release 1.0

Doug Hellmann

April 09, 2015

Contents

1 Introduction **3**

2 List Output Formatters **5**
- 2.1 html . . . 5
- 2.2 json . . . 5
- 2.3 yaml . . . 5

3 Show Output Formatters **7**
- 3.1 html . . . 7
- 3.2 json . . . 7
- 3.3 yaml . . . 8

4 Installation **9**
- 4.1 Python Versions . . . 9
- 4.2 Basic Installation . . . 9
- 4.3 Source Code . . . 9
- 4.4 Reporting Bugs . . . 9

5 For Developers **11**
- 5.1 Building Documentation . . . 11
- 5.2 Running Tests . . . 12

6 Release History **13**

7 Indices and tables **15**

cliff-tablib includes formatters to be used in applications based on the cliff framework.

Contents:

CHAPTER 1

Introduction

The cliff framework is meant to be used to create multi-level commands such as subversion and git, where the main program handles some basic argument parsing and then invokes a sub-command to do the work. This package adds JSON, YAML, and HTML output formatters to those commands.

CHAPTER 2

List Output Formatters

cliff-tablib delivers several new output formatters for list commands.

2.1 html

The `html` formatter uses tablib to produce HTML output as a table.

```
(.venv)$ cliffdemo files -f html
<table>
<thead>
<tr><th>Name</th>
<th>Size</th></tr>
</thead>
<tr><td>build</td>
<td>136</td></tr>
<tr><td>cliffdemo.log</td>
<td>3252</td></tr>
<tr><td>Makefile</td>
<td>5569</td></tr>
<tr><td>requirements.txt</td>
<td>33</td></tr>
<tr><td>source</td>
<td>782</td></tr>
</table>
```

2.2 json

The `json` formatter uses tablib to produce JSON output.

```
(.venv)$ cliffdemo files -f json
[{"Name": "build", "Size": 136}, {"Name": "cliffdemo.log", "Size":
3461}, {"Name": "Makefile", "Size": 5569}, {"Name":
"requirements.txt", "Size": 33}, {"Name": "source", "Size": 782}]
```

2.3 yaml

The `yaml` formatter uses tablib to produce YAML output as a sequence of mappings.

```
(.venv)$ cliffdemo files -f yaml
- {Name: build, Size: 136}
- {Name: cliffdemo.log, Size: 3043}
- {Name: Makefile, Size: 5569}
- {Name: requirements.txt, Size: 33}
- {Name: source, Size: 816}
```

CHAPTER 3

Show Output Formatters

cliff is delivered with output formatters for show commands. `ShowOne` adds a command line switch to let the user specify the formatter they want, so you don't have to do any extra work in your application.

3.1 html

The `html` formatter uses tablib to produce HTML output as a table.

```
(.venv)$ cliffdemo file -f html setup.py
<table>
<thead>
<tr><th>Field</th>
<th>Value</th></tr>
</thead>
<tr><td>Name</td>
<td>setup.py</td></tr>
<tr><td>Size</td>
<td>6373</td></tr>
<tr><td>UID</td>
<td>527</td></tr>
<tr><td>GID</td>
<td>501</td></tr>
<tr><td>Modified Time</td>
<td>1336353173.0</td></tr>
</table>
```

3.2 json

The `json` formatter uses tablib to produce JSON output.

```
(.venv)$ cliffdemo file -f json setup.py
[{"Field": "Name", "Value": "setup.py"}, {"Field": "Size",
"Value": 6373}, {"Field": "UID", "Value": 527}, {"Field": "GID",
"Value": 501}, {"Field": "Modified Time", "Value": 1336353173.0}]
```

3.3 yaml

The `yaml` formatter uses tablib to produce YAML output as a sequence of mappings.

```
(.venv)$ cliffdemo file -f yaml setup.py
- {Field: Name, Value: setup.py}
- {Field: Size, Value: 6373}
- {Field: UID, Value: 527}
- {Field: GID, Value: 501}
- {Field: Modified Time, Value: 1336353173.0}
```

CHAPTER 4

Installation

4.1 Python Versions

cliff-tablib is being developed under Python 2.7 and tested with Python 3.2.

4.2 Basic Installation

cliff-tablib should be installed into the same site-packages area where the application and extensions are installed (either a virtualenv or the global site-packages). You may need administrative privileges to do that. The easiest way to install it is using pip:

```
$ pip install cliff-tablib
```

or:

```
$ sudo pip install cliff-tablib
```

4.3 Source Code

The source is hosted on github: https://github.com/dreamhost/cliff-tablib

4.4 Reporting Bugs

Please report bugs through the github project: https://github.com/dreamhost/cliff-tablib/issues

CHAPTER 5

For Developers

If you would like to contribute to cliff-tablib directly, these instructions should help you get started. Patches, bug reports, and feature requests are all welcome through the GitHub project. Contributions in the form of patches or pull requests are easier to integrate and will receive priority attention.

5.1 Building Documentation

The documentation for cliff-tablib is written in reStructuredText and converted to HTML using Sphinx. The build itself is driven by make. You will need the following packages in order to build the docs:

- Sphinx
- docutils

Once all of the tools are installed into a virtualenv using pip, run `make docs` to generate the HTML version of the documentation:

```
$ make docs
(cd docs && make clean html)
sphinx-build -b html -d build/doctrees   source build/html
Running Sphinx v1.1.3
loading pickled environment... done
building [html]: targets for 1 source files that are out of date
updating environment: 1 added, 1 changed, 0 removed
reading sources... [100%] index
looking for now-outdated files... none found
pickling environment... done
done
preparing documents... done
writing output... [100%] index
writing additional files... genindex search
copying static files... done
dumping search index... done
dumping object inventory... done
build succeeded, 2 warnings.

Build finished. The HTML pages are in build/html.
```

The output version of the documentation ends up in `./docs/build/html` inside your sandbox.

5.2 Running Tests

The test suite for cliff-tablib uses tox, which must be installed separately (`pip install tox`).

To run the tests under Python 2.7 and 3.2, run `tox` from the top level directory of the git repository.

To run tests under a single version of Python, specify the appropriate environment when running tox:

```
$ tox -e py27
```

Add new tests by modifying an existing file or creating new script in the `tests` directory.

CHAPTER 6

Release History

1.1

- Remove the use of distribute for installation.

1.0

- Initial public release, created by pulling the tablib integration out of cliff and repackaging it.

CHAPTER 7

Indices and tables

- *genindex*
- *modindex*
- *search*