Dictdiffer Documentation

Release 0.2.0

Fatih Erikli

September 29, 2014

Dictdiffer is a helper module that helps you to diff and patch dictionaries.

Installation

Dictdiffer is on PyPI so all you need is:

```
$ pip install dictdiffer
```

Usage

Let's start with an example on how to find the diff between two dictionaries using `diff()` method:

```python
from dictdiffer import diff, patch, swap, revert

first = {
    "title": "hello",
    "fork_count": 20,
    "stargazers": ["/users/20", "/users/30"],
    "settings": {
        "assignees": [100, 101, 201],
    }
}

second = {
    "title": "hellooo",
    "fork_count": 20,
    "stargazers": ["/users/20", "/users/30", "/users/40"],
    "settings": {
        "assignees": [100, 101, 202],
    }
}

result = diff(first, second)

assert list(result) == [
    ('change', ['settings', 'assignees', 2], (201, 202)),
    ('add', 'stargazers', [(2, '/users/40')]),
    ('change', 'title', ('hello', 'hellooo'))]
```

Now we can apply the diff result with `patch()` method:

```python
result = diff(first, second)
patched = patch(result, first)

assert patched == second
```

Also we can swap the diff result with `swap()` method:

```python
result = diff(first, second)
swapped = swap(result)

assert list(swapped) == [
    ('change', ['settings', 'assignees', 2], (202, 201)),
```

```
        ('remove', 'stargazers', [(2, '/users/40')]),
        ('change', 'title', ('hellooo', 'hello'))]
```

Let's revert the last changes:

```
result = diff(first, second)
reverted = revert(result, patched)
assert reverted == first
```

API

Dictdiffer is a helper module to diff and patch dictionaries.

dictdiffer.**diff**(*first*, *second*, *node=None*)
Compare two dictionary object, and returns a diff result.

Return iterator with differences between two dictionaries.

```
>>> result = diff({'a':'b'}, {'a':'c'})
>>> list(result)
[('change', 'a', ('b', 'c'))]
```

dictdiffer.**patch**(*diff_result*, *destination*)
Patch the diff result to the old dictionary.

dictdiffer.**swap**(*diff_result*)
Swap the diff result.

It uses following mapping:

> •remove -> add
>
> •add -> remove

In addition, swap the changed values for *change* flag.

```
>>> swapped = swap([('add', 'a.b.c', ('a', 'b'))])
>>> next(swapped)
('remove', 'a.b.c', ('a', 'b'))

>>> swapped = swap([('change', 'a.b.c', ('a', 'b'))])
>>> next(swapped)
('change', 'a.b.c', ('b', 'a'))
```

dictdiffer.**revert**(*diff_result*, *destination*)
Call swap function to revert patched dictionary object.

Usage example:

```
>>> first = {'a': 'b'}
>>> second = {'a': 'c'}
>>> revert(diff(first, second), second)
{'a': 'b'}
```

dictdiffer.**dot_lookup**(*source*, *lookup*, *parent=False*)
Allow you to reach dictionary items with string or list lookup.

Recursively find value by lookup key split by '.'.

```
>>> dot_lookup({'a': {'b': 'hello'}}, 'a.b')
'hello'
```

If parent argument is True, returns the parent node of matched object.

```
>>> dot_lookup({'a': {'b': 'hello'}}, 'a.b', parent=True)
{'b': 'hello'}
```

If node is empty value, returns the whole dictionary object.

```
>>> dot_lookup({'a': {'b': 'hello'}}, '')
{'a': {'b': 'hello'}}
```

Changes

Version 0.2.0 (released 2014-09-29):

- Fix for empty list instructions. (#30)
- Regression test for empty list instructions.

Version 0.1.0 (released 2014-09-01):

- Fix for list removal issues during patching caused by wrong iteration. (#10)
- Fix for issues with multiple value types for the same key. (#10)
- Fix for issues with strings handled as iterables. (#6)
- Fix for integer keys. (#12)
- Regression test for complex dictionaries. (#4)
- Better testing with Travis CI, tox, pytest, code coverage. (#10)
- Initial release of documentation on ReadTheDocs. (#21 #24)
- Support for Python 3. (#15)

Version 0.0.4 (released 2014-01-04):

- List diff behavior treats lists as lists instead of sets. (#3)
- Differed typed objects are flagged as *changed* now.
- Swap function refactored.

Version 0.0.3 (released 2013-05-26):

- Initial public release on PyPI.

Contributing

Bug reports, feature requests, and other contributions are welcome. If you find a demonstrable problem that is caused by the code of this library, please:

1. Search for already reported problems.

2. Check if the issue has been fixed or is still reproducible on the latest *master* branch.

3. Create an issue with **a test case**.

If you create a feature branch, you can run the tests to ensure everything is operating correctly:

```
$ ./run-tests.sh

...

Name                  Stmts   Miss  Cover   Missing
---------------------------------------------------
dictdiffer/__init__      88      0   100%
dictdiffer/version        2      0   100%
---------------------------------------------------
TOTAL                    90      0   100%

...

52 passed, 2 skipped in 0.44 seconds
```

License

Dictdiffer is free software; you can redistribute it and/or modify it under the terms of the MIT License quoted below.

Copyright (C) 2013 Fatih Erikli. Copyright (C) 2013, 2014 CERN.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

In applying this license, CERN does not waive the privileges and immunities granted to it by virtue of its status as an Intergovernmental Organization or submit itself to any jurisdiction.

Authors

Dictdiffer was originally developed by Fatih Erikli. It is now being developed and maintained by the Invenio collaboration. You can contact us at info@invenio-software.org.

Contributors:

- Fatih Erikli <fatiherikli@gmail.com>

- Brian Rue <brianrue@gmail.com>

- Lars Holm Nielsen <lars.holm.nielsen@cern.ch>

- Tibor Simko <tibor.simko@cern.ch>

- Jiri Kuncar <jiri.kuncar@cern.ch>

- Jason Peddle <jwpeddle@gmail.com>

d