brunns-row Documentation

Release 2.0.0

Simon Brunning

Oct 19, 2022

CONTENTS:

Convenience wrapper for DB API and csv.DictReader rows, and similar, inspired by Greg Stein's lovely dtuple module.

CHAPTER

ONE

API REFERENCE

1.1 The "rowwapper" module

class brunns.row.rowwrapper.**RowWrapper**(*description*, *force_lower_case_ids=False*)

Build lightweight row tuples for DB API and csv.DictReader rows.

Inspired by Greg Stein's lovely dtuple module, which I can't find online any longer, isn't on pypi, and doesn't support Python 3 without some fixes.

Initializer takes a sequence of column descriptions, either names, or tuples of names and other metadata (which will be ignored). For instance, it's happy to take a DB API cursor description, or a csv.DictReader's fieldnames property. Provides a wrap(row) method for wrapping rows.

Some characters which are illegal in identifiers will be replaced when building the row tuples - currently "-" and " " characters will be replaced with "_"s.

```
>>> cursor = conn.cursor()
>>> cursor.execute("SELECT kind, rating FROM sausages ORDER BY rating DESC;")
>>> wrapper = RowWrapper(cursor.description)
>>> rows = [wrapper.wrap(row) for row in cursor.fetchall()]
```

```
>>> reader = csv.DictReader(csv_file)
>>> wrapper = RowWrapper(reader.fieldnames)
>>> rows = [wrapper.wrap(row) for row in reader]
```

wrap(*row*)

Return row tuple for row.

wrap_all(*rows*)

Return row tuple for each row in rows.

CHAPTER

TWO

INSTALLATION

Install from Pypi as usual, using pip , tox, or `setup.py`.

CHAPTER

THREE

USAGE

The basic approach is to create a wrapper object from some kind of description - typically a DBAPI cursor's description, or a csv.DictReader's fieldnames attribute - then to use the wrapper's `wrap(row)` method to wrap each row. `wrap(row)` returns an object from which you can access the row's fields as attributes. A couple of simple examples:

3.1 DB API

```
cursor = conn.cursor()
cursor.execute("SELECT kind, rating FROM sausages ORDER BY rating DESC;")
wrapper = RowWrapper(cursor.description)
rows = [wrapper.wrap(row) for row in cursor.fetchall()]
for row in rows:
    print(row.kind, row.rating)
```

3.2 csv.DictReader

```
reader = csv.DictReader(csv_file)
wrapper = RowWrapper(reader.fieldnames)
rows = [wrapper.wrap(row) for row in reader]
for row in rows:
    print(row.kind, row.rating)
```

Attributes names are simply the column names where possible, converted to valid identifiers where necessary by replacing invalid characters with "_"s, prefixing any leading numerics with "a_", and de-duplicating where necessary by adding numeric suffixes.

CHAPTER

FOUR

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

b

INDEX