
pyLoDStorage

Mar 28, 2023

Contents:

1	sampledata package	1
1.1	Module contents	1
2	setup module	3
3	tests package	5
3.1	Submodules	5
3.2	tests.basetest module	5
3.3	tests.testEntityManager module	6
3.4	tests.testJson module	6
3.5	tests.testPandas module	7
3.6	tests.testSPARQL module	7
3.7	tests.testSqlite3 module	9
3.8	tests.testTrulyTabular module	10
3.9	tests.test_Plot module	11
3.10	tests.test_Tabulate module	12
3.11	tests.test_csv module	12
3.12	tests.test_lod module	13
3.13	tests.test_queries module	13
3.14	Module contents	15
4	Indices and tables	17
	Python Module Index	19
	Index	21

CHAPTER 1

sampledata package

1.1 Module contents

CHAPTER 2

setup module

3.1 Submodules

3.2 tests.basetest module

Created on 2021-08-19

@author: wf

```
class tests.basetest.Basetest (methodName='runTest')
```

```
    Bases: unittest.case.TestCase
```

```
    base test case
```

```
    Create an instance of the class that will use the named test method when executed. Raises a ValueError if the instance does not have a method with the specified name.
```

```
    static inPublicCI ()
```

```
        are we running in a public Continuous Integration Environment?
```

```
    static isUser (name: str)
```

```
        Checks if the system has the given name
```

```
    setUp (debug=False, profile=True)
```

```
        setUp test environment
```

```
    tearDown ()
```

```
        Hook method for deconstructing the test fixture after testing it.
```

```
class tests.basetest.Profiler (msg, profile=True)
```

```
    Bases: object
```

```
    simple profiler
```

```
    construct me with the given msg and profile active flag
```

```
        Parameters
```

- **msg** (*str*) – the message to show if profiling is active
- **profile** (*bool*) – True if messages should be shown

time (*extraMsg=""*)
time the action and print if profile is active

3.3 tests.testEntityManager module

Created on 2021-07-23

@author: wf

```
class tests.testEntityManager.TestEntityManager (methodName='runTest')
    Bases: tests.basetest.Basetest

    test the entity manager wrapper

    Create an instance of the class that will use the named test method when executed. Raises a ValueError if the
    instance does not have a method with the specified name.

    checkItem (item1, item2, attrs, msg)

    configure (config: lodstorage.storageconfig.StorageConfig)

    testEntityManager ()
        test the entity Manager handling

    testStoreMode ()
        test store mode display
```

3.4 tests.testJson module

Created on 2020-09-12

@author: wf

```
class tests.testJson.ServerConfig
    Bases: lodstorage.jsonable.JSONAble

class tests.testJson.TestJsonAble (methodName='runTest')
    Bases: tests.basetest.Basetest

    test JSON serialization with JsonAble mixin

    Create an instance of the class that will use the named test method when executed. Raises a ValueError if the
    instance does not have a method with the specified name.

    check (manager, manager1, listName, debugLimit)
        check that the list of the two managers are the same

    dumpListOfDicts (listOfDicts, limit)

    setUp ()
        setUp test environment

    testIssue22 ()
        https://github.com/WolfgangFahl/pyLoDStorage/issues/22 Regression: storeToJsonFile and restoreFromJ-
        sonFile missing in JSONAble
```

```

testIssue27_Lookup ()
    add Lookup map option

testIssue30_SampleLimited ()
    tests if the json export is correctly limited to the fields that are used in the samples

testJsonAble ()
    test JSONAble

testPluralName ()

testRestoreFromJsonStr ()
    Tests restoring a JsonAbleList form a json string

testRoyals ()
    test Royals example

testSingleQuoteToDoubleQuoteStackoverflow ()
    see
        • https://stackoverflow.com/a/63862387/1497139
        • https://stackoverflow.com/a/50257217/1497139

testSingleToDoubleQuote ()

testStoreAndRestore ()
    test storing and restoring from a JSON file https://github.com/WolfgangFahl/pyLoDStorage/issues/21

```

3.5 tests.testPandas module

Created on 2021-06-07

@author: mk

```

class tests.testPandas.TestPandas (methodName='runTest')
    Bases: tests.basetest.Basetest

```

Create an instance of the class that will use the named test method when executed. Raises a ValueError if the instance does not have a method with the specified name.

```

testIssue25 ()
    see https://github.com/WolfgangFahl/pyLoDStorage/issues/25

```

3.6 tests.testSPARQL module

Created on 2020-08-14

@author: wf

```

class tests.testSPARQL.TestSPARQL (methodName='runTest')
    Bases: tests.basetest.Basetest

```

Test SPARQL access e.g. Apache Jena via Wrapper

Create an instance of the class that will use the named test method when executed. Raises a ValueError if the instance does not have a method with the specified name.

checkErrors (*errors*, *expected=0*)

check the given list of errors - print any errors if there are some and after that assert that the length of the list of errors is zero

Parameters **errors** (*list*) – the list of errors to check

getJena (*mode='query'*, *debug=False*, *typedLiterals=False*, *profile=False*)

get the jena endpoint for the given mode

Parameters

- **mode** (*string*) – query or update
- **debug** (*boolean*) – True if debug information should be output
- **typedLiterals** (*boolean*) – True if INSERT DATA SPARQL commands should use typed literals
- **profile** (*boolean*) – True if profile/timing information should be shown

testControlEscape ()

check the control-escaped version of an UTF-8 string

testDob ()

test the DOB (date of birth) function that converts from ISO-Date to datetime.date

testEscapeStringContent ()

test handling of double quoted strings

testIssue20And76 ()

see <https://github.com/WolfgangFahl/pyLoDStorage/issues/20> add fixNone option to SPARQL results (same functionality as in SQL)

<https://github.com/WolfgangFahl/pyLoDStorage/issues/76>

SPARQL GET method support

testIssue7 ()

test conversion of dates with timezone info

testJenaInsert ()

test a Jena INSERT DATA

testJenaQuery ()

test Apache Jena Fuseki SPARQL endpoint with example SELECT query

testListOfDictInsert ()

test inserting a list of Dicts and retrieving the values again using a person based example instead of [https://en.wikipedia.org/wiki/FOAF_\(ontology\)](https://en.wikipedia.org/wiki/FOAF_(ontology))

we use an object oriented derivate of FOAF with a focus on datatypes

testListOfDictSpeed ()

test the speed of adding data

testSPARQLErrorMessage ()

test error handling see <https://stackoverflow.com/questions/63486767/how-can-i-get-the-fuseki-api-via-sparqlwrapper-to-properly-report-a-detailed-err>

testStackoverflow55961615Query ()

see <https://stackoverflow.com/questions/55961615/how-to-integrate-wikidata-query-in-python>
<https://stackoverflow.com/a/69771615/1497139>

```

testStackoverflow71444069 ()
    https://stackoverflow.com/questions/71444069/create-csv-from-result-of-a-for-google-colab/71548650#71548650

testWikidata ()
    check wikidata

test_query_with_authentication ()
    tests querying an endpoint that requires authentication

```

3.7 tests.testSqlite3 module

Created on 2020-08-24

@author: wf

```
class tests.testSqlite3.TestSQLDB (methodName='runTest')
```

Bases: *tests.basetest.Basetest*

Test the SQLDB database wrapper

Create an instance of the class that will use the named test method when executed. Raises a ValueError if the instance does not have a method with the specified name.

```
checkListOfRecords (listOfRecords, entityName, primaryKey=None, executeMany=True,  
                    fixNone=False, fixDates=False, debug=False, doClose=True)  
    check the handling of the given list of Records
```

Parameters

- **listOfRecords** (*list*) – a list of dicts that contain the data to be stored
- **entityName** (*string*) – the name of the entity type to be used as a table name
- **primaryKey** (*string*) – the name of the key / column to be used as a primary key
- **executeMany** (*boolean*) – True if executeMany mode of sqlite3 should be used
- **fixNone** (*boolean*) – fix dict entries that are undefined to have a “None” entry
- **debug** (*boolean*) – True if debug information e.g. CREATE TABLE and INSERT INTO commands should be shown
- **doClose** (*boolean*) – True if the connection should be closed

```
static getSampleTableDB (withDrop=False, debug=False, failIfTooFew=False, sample-  
                        Size=1000)
```

```
testBackup ()  
    test creating a backup of the SQL database
```

```
testBindingError ()  
    test list of Records with incomplete record leading to “You did not supply a value for binding 2” see  
    https://bugs.python.org/issue41638
```

```
testCopy ()  
    test copying databases into another database
```

```
testEntityInfo ()  
    test creating entityInfo from the sample record
```

```
testIssue110 ()  
    https://github.com/WolfgangFahl/pyLoDStorage/issues/110
```

testIssue13_setNoneValue ()
 <https://github.com/WolfgangFahl/pyLoDStorage/issues/13> set None value for undefined LoD entries

testIssue14_execute ()
 <https://github.com/WolfgangFahl/pyLoDStorage/issues/14>
 offer execute wrapper directly via sqlDB

testIssue15 ()
 <https://github.com/WolfgangFahl/pyLoDStorage/issues/15>
 auto create view ddl in mergeschema

testIssue16 ()
 <https://github.com/WolfgangFahl/pyLoDStorage/issues/16> allow to only warn if samplerrecordcount is higher than number of available records

testIssue18 ()
 <https://github.com/WolfgangFahl/pyLoDStorage/issues/18>

testIssue41 ()
 <https://github.com/WolfgangFahl/pyLoDStorage/issues/41> improve error message when create table command fails

testIssue87AllowUsingQueryWithGenerator ()
 test the query gen approach

testListOfCities ()
 test sqlite3 with some 120000 city records

testQueryParams ()
 test Query Params

testSqlite3 ()
 test sqlite3 with a few records from the royal family

testSqlite3Speed ()
 test sqlite3 speed with some 100000 artificial sample records consisting of two columns with a running index

testUniqueConstraint ()
 test for <https://github.com/WolfgangFahl/pyLoDStorage/issues/4> sqlite3.IntegrityError: UNIQUE constraint failed: ... show debug info

3.8 tests.testTrulyTabular module

Created on 2022-03-4

@author: wf

class tests.testTrulyTabular.**TestTrulyTabular** (*methodName='runTest'*)

 Bases: unittest.case.TestCase

 test Truly tabular analysis

 Create an instance of the class that will use the named test method when executed. Raises a ValueError if the instance does not have a method with the specified name.

documentQuery (*tt, query, show=True, formats=['mediawiki']*)

 document the given query for the given TrueTabular instance

handleServiceUnavailable (*ex*, *endpointConf*)

handle service unavailable

Parameters

- **ex** (*Exception*) – the exception to handle
- **endpointConf** (*Endpoint*) – the endpoint for which there is a problem

setUp ()

Hook method for setting up the test fixture before exercising it.

tearDown ()

Hook method for deconstructing the test fixture after testing it.

testCount ()

test the count function of truly tabular

testGenerateSparqlQuery ()

test Generating a SPARQL query

testGetFirst ()

test the get First helper function

testGetItemsByLabel ()

try getting items by label

testGetPropertiesById ()

try getting properties by label

testGetPropertiesByLabel ()

try getting properties by label

testMostFrequentProperties ()

test getting the most frequent properties for some Wikidata Item types

testSyntaxHighlighting ()

<https://github.com/WolfgangFahl/pyLoDStorage/issues/81>

testTrulyTabularTables ()

test Truly Tabular for different tabular queries

3.9 tests.test_Plot module

Created on 2020-07-05

@author: wf

class tests.test_Plot.**TestPlot** (*methodName='runTest'*)

Bases: *tests.basetest.Basetest*

test the Plot helper class

Create an instance of the class that will use the named test method when executed. Raises a ValueError if the instance does not have a method with the specified name.

testPlot ()

test a plot based on a Counter

3.10 tests.test_Tabulate module

Created on 2021-06-13

@author: wf

class tests.test_Tabulate.**TestTabulate** (*methodName='runTest'*)

Bases: *tests.basetest.Basetest*

test tabulate support/compatibility

Create an instance of the class that will use the named test method when executed. Raises a ValueError if the instance does not have a method with the specified name.

testIssue24_IntegrateTabulate ()

<https://github.com/WolfgangFahl/pyLoDStorage/issues/24>

test <https://pypi.org/project/tabulate/> support

3.11 tests.test_csv module

class tests.test_csv.**TestCSV** (*methodName='runTest'*)

Bases: *tests.basetest.Basetest*

Tests functionalities for the conversion between csv and list of dicts (LoD)

Create an instance of the class that will use the named test method when executed. Raises a ValueError if the instance does not have a method with the specified name.

setUp ()

setUp test environment

tearDown () → None

Hook method for deconstructing the test fixture after testing it.

testCsvFromJSONable ()

tests generation of csv from list of JSONable object

testCsvFromJSONableExcludeFields ()

tests generation of csv from list of JSONable object with excluding specific fields (negative list)

testCsvFromJSONableIncludeFields ()

tests generation of csv from list of JSONable object with including only specified fields (positive list)

testRoyals ()

test conversion of royals

test_from_csv ()

tests if the csv is correctly parsed to an LoD

test_from_csv_without_header ()

tests if csv string without embedded headers is parsed correctly

test_restore_from_csvfile ()

tests if the lod is correctly restored from csv file

test_round_trip ()

tests the csv round trip: dict -> csv -> dict Note: the initial dict has missing values it is expected that the final dict has the missing keys with None as value


```

test_store_to_csvfile()
    tests if LoD is correctly stored as csv file

test_to_csv()
    tests if LoD is correctly converted to csv str

test_to_csv_delimiter_in_value()
    tests if delimiter in dict value will not result in incorrect values

test_to_csv_incomplete_dicts()
    tests if the LoD is correctly converted to csv even if some dicts are incomplete Note: incomplete dicts can
    lead to changes of the column orders of the csv string

```

3.12 tests.test_lod module

Created on 2021-06-11

@author: wf

```

class tests.test_lod.TestLOD (methodName='runTest')
    Bases: tests.basetest.Basetest

    test list of dicts base functionality

    Create an instance of the class that will use the named test method when executed. Raises a ValueError if the
    instance does not have a method with the specified name.

    checkHandleListTypeResult (lod, expectedLen, expected)
        check the result of the handleListType function

        Parameters
        • lod (list) – the list of dicts to check
        • expectedLen (int) – the expected Length
        • expected (str) – the expected entry for the München,Munich Q1524 record with a list

    testGetFields ()
        tests field extraction from list of JSONable objects and LoD

    testGetLookupIssue31And32 ()
        test for https://github.com/WolfgangFahl/pyLoDStorage/issues/31 test for https://github.com/WolfgangFahl/pyLoDStorage/issues/32

    testListHandlingIssue33 ()
        test for handling list

    testListIntersect ()
        test a list intersection

```

3.13 tests.test_queries module

Created on 2021-01-29

@author: wf

```

class tests.test_queries.TestEndpoints (methodName='runTest')
    Bases: tests.basetest.Basetest

```

tests Endpoint

Create an instance of the class that will use the named test method when executed. Raises a ValueError if the instance does not have a method with the specified name.

testEndpoints()
tests getting and rawQuerying Endpoints

class tests.test_queries.**TestQueries** (*methodName='runTest'*)

Bases: *tests.basetest.Basetest*

Test query handling

Create an instance of the class that will use the named test method when executed. Raises a ValueError if the instance does not have a method with the specified name.

captureQueryMain (*args*)
run queryMain and capture stdout

Parameters *args* (*list*) – command line arguments

Returns the stdout content of the command line call

Return type str

testCommandLineUsage()
test the command line usage

testIssue111()
<https://github.com/WolfgangFahl/pyLoDStorage/issues/111> add basicauth support for endpoints

testIssue115Limit()
test the limit argument

testIssue61()
tests different query path

see <https://github.com/WolfgangFahl/pyLoDStorage/issues/61>

testIssue69showEndpoints()
test the listEndpoints option

<https://github.com/WolfgangFahl/pyLoDStorage/issues/69>

testIssue70showQuery()
test the showQuery option

<https://github.com/WolfgangFahl/pyLoDStorage/issues/70>

testIssue73Formatting()
test formatting

testIssue73ReadFormats()
test reading the valueFormatters

testIssue89()
test fix TypeError('Object of type datetime is not JSON serializable') #89

testQueryCommandLine()
test the sparql query command line

testQueryDocumentation()
test QueryDocumentation

testQueryEndpoints()
tests the sparql endpoint commandline endpoint selection

```
testSQLQueries ()  
    see https://github.com/WolfgangFahl/pyLoDStorage/issues/19  
  
testSparqlQueries ()  
    test SPARQL queries  
  
testUnicode2LatexWorkaround ()  
    test the uniCode2Latex conversion workaround
```

3.14 Module contents

CHAPTER 4

Indices and tables

- `genindex`
- `modindex`
- `search`

S

`sampledata`, 1

t

`tests`, 15

`tests.basetest`, 5

`tests.test_csv`, 12

`tests.test_lod`, 13

`tests.test_Plot`, 11

`tests.test_queries`, 13

`tests.test_Tabulate`, 12

`tests.testEntityManager`, 6

`tests.testJson`, 6

`tests.testPandas`, 7

`tests.testSPARQL`, 7

`tests.testSqlite3`, 9

`tests.testTrulyTabular`, 10

B

Basetest (class in tests.basetest), 5

C

captureQueryMain()
 (tests.test_queries.TestQueries method), 14
 check() (tests.testJson.TestJsonAble method), 6
 checkErrors() (tests.testSPARQL.TestSPARQL method), 7
 checkHandleListTypeResult()
 (tests.test_lod.TestLOD method), 13
 checkItem() (tests.testEntityManager.TestEntityManager method), 6
 checkListOfRecords()
 (tests.testSqlite3.TestSQLDB method), 9
 configure() (tests.testEntityManager.TestEntityManager method), 6

D

documentQuery() (tests.testTrulyTabular.TestTrulyTabular method), 10
 dumpListOfDicts() (tests.testJson.TestJsonAble method), 6

G

getJena() (tests.testSPARQL.TestSPARQL method), 8
 getSampleTableDB() (tests.testSqlite3.TestSQLDB static method), 9

H

handleServiceUnavailable()
 (tests.testTrulyTabular.TestTrulyTabular method), 10

I

inPublicCI() (tests.basetest.Basetest static method), 5
 isUser() (tests.basetest.Basetest static method), 5

P

Profiler (class in tests.basetest), 5

S

sampladata (module), 1
 ServerConfig (class in tests.testJson), 6
 setUp() (tests.basetest.Basetest method), 5
 setUp() (tests.test_csv.TestCSV method), 12
 setUp() (tests.testJson.TestJsonAble method), 6
 setUp() (tests.testTrulyTabular.TestTrulyTabular method), 11

T

tearDown() (tests.basetest.Basetest method), 5
 tearDown() (tests.test_csv.TestCSV method), 12
 tearDown() (tests.testTrulyTabular.TestTrulyTabular method), 11
 test_from_csv() (tests.test_csv.TestCSV method), 12
 test_from_csv_without_header()
 (tests.test_csv.TestCSV method), 12
 test_query_with_authentication()
 (tests.testSPARQL.TestSPARQL method), 9
 test_restore_from_csvfile()
 (tests.test_csv.TestCSV method), 12
 test_round_trip() (tests.test_csv.TestCSV method), 12
 test_store_to_csvfile()
 (tests.test_csv.TestCSV method), 12
 test_to_csv() (tests.test_csv.TestCSV method), 13
 test_to_csv_delimiter_in_value()
 (tests.test_csv.TestCSV method), 13
 test_to_csv_incomplete_dicts()
 (tests.test_csv.TestCSV method), 13
 testBackup() (tests.testSqlite3.TestSQLDB method), 9
 testBindingError() (tests.testSqlite3.TestSQLDB method), 9

`testCommandLineUsage()`
 (*tests.test_queries.TestQueries* *method*), 14

`testControlEscape()`
 (*tests.testSPARQL.TestSPARQL* *method*), 8

`testCopy()` (*tests.testSqlite3.TestSQLDB* *method*), 9

`testCount()` (*tests.testTrulyTabular.TestTrulyTabular* *method*), 11

`TestCSV` (*class in tests.test_csv*), 12

`testCsvFromJSONAble()` (*tests.test_csv.TestCSV* *method*), 12

`testCsvFromJSONAbleExcludeFields()`
 (*tests.test_csv.TestCSV* *method*), 12

`testCsvFromJSONAbleIncludeFields()`
 (*tests.test_csv.TestCSV* *method*), 12

`testDob()` (*tests.testSPARQL.TestSPARQL* *method*), 8

`TestEndpoints` (*class in tests.test_queries*), 13

`testEndpoints()` (*tests.test_queries.TestEndpoints* *method*), 14

`testEntityInfo()` (*tests.testSqlite3.TestSQLDB* *method*), 9

`TestEntityManager` (*class in tests.testEntityManager*), 6

`testEntityManager()`
 (*tests.testEntityManager.TestEntityManager* *method*), 6

`testEscapeStringContent()`
 (*tests.testSPARQL.TestSPARQL* *method*), 8

`testGenerateSparqlQuery()`
 (*tests.testTrulyTabular.TestTrulyTabular* *method*), 11

`testGetFields()` (*tests.test_lod.TestLOD* *method*), 13

`testGetFirst()` (*tests.testTrulyTabular.TestTrulyTabular* *method*), 11

`testGetItemsByLabel()`
 (*tests.testTrulyTabular.TestTrulyTabular* *method*), 11

`testGetLookupIssue31And32()`
 (*tests.test_lod.TestLOD* *method*), 13

`testGetPropertiesById()`
 (*tests.testTrulyTabular.TestTrulyTabular* *method*), 11

`testGetPropertiesByLabel()`
 (*tests.testTrulyTabular.TestTrulyTabular* *method*), 11

`testIssue110()` (*tests.testSqlite3.TestSQLDB* *method*), 9

`testIssue111()` (*tests.test_queries.TestQueries* *method*), 14

`testIssue115Limit()`
 (*tests.test_queries.TestQueries* *method*), 14

`testIssue13_setNoneValue()`
 (*tests.testSqlite3.TestSQLDB* *method*), 9

`testIssue14_execute()`
 (*tests.testSqlite3.TestSQLDB* *method*), 10

`testIssue15()` (*tests.testSqlite3.TestSQLDB* *method*), 10

`testIssue16()` (*tests.testSqlite3.TestSQLDB* *method*), 10

`testIssue18()` (*tests.testSqlite3.TestSQLDB* *method*), 10

`testIssue20And76()`
 (*tests.testSPARQL.TestSPARQL* *method*), 8

`testIssue22()` (*tests.testJson.TestJsonAble* *method*), 6

`testIssue24_IntegrateTabulate()`
 (*tests.test_Tabulate.TestTabulate* *method*), 12

`testIssue25()` (*tests.testPandas.TestPandas* *method*), 7

`testIssue27_Lookup()`
 (*tests.testJson.TestJsonAble* *method*), 6

`testIssue30_SampleLimited()`
 (*tests.testJson.TestJsonAble* *method*), 7

`testIssue41()` (*tests.testSqlite3.TestSQLDB* *method*), 10

`testIssue61()` (*tests.test_queries.TestQueries* *method*), 14

`testIssue69showEndpoints()`
 (*tests.test_queries.TestQueries* *method*), 14

`testIssue7()` (*tests.testSPARQL.TestSPARQL* *method*), 8

`testIssue70showQuery()`
 (*tests.test_queries.TestQueries* *method*), 14

`testIssue73Formatting()`
 (*tests.test_queries.TestQueries* *method*), 14

`testIssue73ReadFormats()`
 (*tests.test_queries.TestQueries* *method*), 14

`testIssue87AllowUsingQueryWithGenerator()`
 (*tests.testSqlite3.TestSQLDB* *method*), 10

`testIssue89()` (*tests.test_queries.TestQueries* *method*), 14

`testJenaInsert()` (*tests.testSPARQL.TestSPARQL* *method*), 8

`testJenaQuery()` (*tests.testSPARQL.TestSPARQL* *method*), 8

`TestJsonAble` (*class in tests.testJson*), 6

`testJsonAble()` (*tests.testJson.TestJsonAble* *method*), 7

testListHandlingIssue33() (tests.test_lod.TestLOD method), 13
 testListIntersect() (tests.test_lod.TestLOD method), 13
 testListOfCities() (tests.testSqlite3.TestSQLDB method), 10
 testListOfDictInsert() (tests.testSPARQL.TestSPARQL method), 8
 testListOfDictSpeed() (tests.testSPARQL.TestSPARQL method), 8
 TestLOD (class in tests.test_lod), 13
 testMostFrequentProperties() (tests.testTrulyTabular.TestTrulyTabular method), 11
 TestPandas (class in tests.testPandas), 7
 TestPlot (class in tests.test_Plot), 11
 testPlot() (tests.test_Plot.TestPlot method), 11
 testPluralName() (tests.testJson.TestJsonAble method), 7
 TestQueries (class in tests.test_queries), 14
 testQueryCommandLine() (tests.test_queries.TestQueries method), 14
 testQueryDocumentation() (tests.test_queries.TestQueries method), 14
 testQueryEndpoints() (tests.test_queries.TestQueries method), 14
 testQueryParams() (tests.testSqlite3.TestSQLDB method), 10
 testRestoreFromJsonStr() (tests.testJson.TestJsonAble method), 7
 testRoyals() (tests.test_csv.TestCSV method), 12
 testRoyals() (tests.testJson.TestJsonAble method), 7
 tests (module), 15
 tests.basetest (module), 5
 tests.test_csv (module), 12
 tests.test_lod (module), 13
 tests.test_Plot (module), 11
 tests.test_queries (module), 13
 tests.test_Tabulate (module), 12
 tests.testEntityManager (module), 6
 tests.testJson (module), 6
 tests.testPandas (module), 7
 tests.testSPARQL (module), 7
 tests.testSqlite3 (module), 9
 tests.testTrulyTabular (module), 10
 testSingleQuoteToDoubleQuoteStackoverflow() (tests.testJson.TestJsonAble method), 7
 testSingleToDoubleQuote() (tests.testJson.TestJsonAble method), 7
 TestSPARQL (class in tests.testSPARQL), 7
 testSPARQLErrorMessage() (tests.testSPARQL.TestSPARQL method), 8
 testSparqlQueries() (tests.test_queries.TestQueries method), 15
 TestSQLDB (class in tests.testSqlite3), 9
 testSqlite3() (tests.testSqlite3.TestSQLDB method), 10
 testSqlite3Speed() (tests.testSqlite3.TestSQLDB method), 10
 testSQLQueries() (tests.test_queries.TestQueries method), 14
 testStackoverflow55961615Query() (tests.testSPARQL.TestSPARQL method), 8
 testStackoverflow71444069() (tests.testSPARQL.TestSPARQL method), 8
 testStoreAndRestore() (tests.testJson.TestJsonAble method), 7
 testStoreMode() (tests.testEntityManager.TestEntityManager method), 6
 testSyntaxHighlighting() (tests.testTrulyTabular.TestTrulyTabular method), 11
 TestTabulate (class in tests.test_Tabulate), 12
 TestTrulyTabular (class in tests.testTrulyTabular), 10
 testTrulyTabularTables() (tests.testTrulyTabular.TestTrulyTabular method), 11
 testUnicode2LatexWorkaround() (tests.test_queries.TestQueries method), 15
 testUniqueConstraint() (tests.testSqlite3.TestSQLDB method), 10
 testWikidata() (tests.testSPARQL.TestSPARQL method), 9
 time() (tests.basetest.Profiler method), 6