
ABCUnit Backend

Release v1.2.0

Jonathan Haigh

Jun 22, 2021

CONTENTS

| | | |
|----------|------------------------------|-----------|
| 1 | Database Backend | 3 |
| 2 | File System Backend | 5 |
| 3 | API | 7 |
| 3.1 | Backend | 7 |
| 4 | History | 11 |
| 4.1 | 1.2.0 (2021-03-17) | 11 |
| 4.2 | 1.1.0 (2021-02-09) | 11 |
| 4.3 | 1.0.0 (2020-11-06) | 11 |
| | Python Module Index | 13 |
| | Index | 15 |

readthedocs

Storage backends for an ABCUnit Framework, logging success and failures of process units.

DATABASE BACKEND

To use this backend you will need to contact the JASMIN help desk (support@jasmin.ac.uk) and ask them to setup a postgresql database for you. Specify a name for the database and a username to login with. JASMIN support will get back to you with the user password and host name.

After you have got your database, you'll need to export an environment variable called `$ABCUNIT_DB_SETTINGS` and set it to a connection string for `psycpg2`:

```
ABCUNIT_DB_SETTINGS="dbname=<name> user=<user> host=<host> password=<pwd>"
```

DatabaseHandler class construction looks like this:

```
DatabaseHandler(table_name="results")
```

Where

- `table_name` is the name of the table logs will be insert into

Connects to an existing database and creates a table to store results:

```
<table_name> (id varchar(255) PRIMARY KEY, result varchar(255) NOT NULL)
```

| id | result |
|----------------------|----------|
| facet1.facet2.facet3 | success |
| facet1.facet2.facet3 | bad_file |
| ⋮ | ⋮ |

FILE SYSTEM BACKEND

FileSystemHandler class construction looks like this:

```
FileSystemHandler(base_log_dir, n_facets, sep)
```

Where;

- `base_log_dir` is the string path to top level directory for logs
- `n_facets` is the number of facets used to describe each unit result
- `sep` is the separator used for a result identifier

Uses the file system to create log files marking success and failures, categorised by directory structure:

```
<log_base_dir>/success/facet1/facet2/facet3
```

```
<log_base_dir>/failure/error_type/facet1/facet2/facet3
```


3.1 Backend

```
class abcunit_backend.database_handler.DatabaseHandler(table_name='results')  
    Bases: abcunit_backend.base_handler.BaseHandler
```

```
count_failures()
```

Returns (int) Number of failed results in the table

```
count_results()
```

Returns (int) Number of results in the table

```
count_successes()
```

Returns (int) Number of successful results in the table

```
delete_all_results()
```

Deletes all entries from the table

```
delete_result(identifier)
```

Deletes entry specified by the given identifier from the database

Parameters **identifier** – (str) Identifier of the job

```
get_all_results()
```

Returns (dict) Dictionary of all job identifiers mapped to their respective results

```
get_failed_runs()
```

Returns (dict) Dictionary of error types mapped to lists of job identifiers which result in them

```
get_result(identifier)
```

Selects the result of the job with the identifier parsed and returns it

Parameters **identifier** – (str) Identifier of the job result

Returns (str) Result of job

```
get_successful_runs()
```

Returns (str list) Returns a list of the identifiers of all successful runs

insert_failure(*identifier*, *error_type*='failure')

Inserts an entry into the table with a given identifier and erroneous result

Parameters

- **identifier** – (str) Identifier of the job
- **error_type** – (str) Result of the job

insert_success(*identifier*)

Inserts an entry into the table with a given identifier and the result 'success'

Parameters **identifier** – (str) Identifier of the job

ran_successfully(*identifier*)

Returns true / false on whether the result with this identifier is successful

Parameters **identifier** – (str) Identifier of the job result

Returns (bool) Boolean on if job ran successfully

class abcunit_backend.file_system_handler.**FileSystemHandler**(*base_log_dir*, *n_facets*, *sep*)

Bases: abcunit_backend.base_handler.BaseHandler

count_failures()

Returns (int) Number of failure result files

count_results()

Returns (int) Number of results in the table

count_successes()

Returns (int) Number of successful result files

delete_all_results()

Deletes all result files in the file system under the base_log_dir

delete_result(*identifier*)

Deletes result file from the file system given its identifier

Parameters **identifier** – (str) Identifier of the job

get_all_results()

Returns (dict) Dictionary of all job identifiers mapped to their respective results

get_failed_runs()

Returns (dict) Dictionary of error types mapped to lists of job identifiers which result in them

get_result(*identifier*)

Returns the value of a result given its identifier

Parameters **identifier** – (str) Identifier of the job

Returns (str) Result of job

get_successful_runs()

Returns (str list) Returns a list of the identifiers of all successful runs

insert_failure(*identifier*, *error_type*)

Creates a failure result file using the identifier and error type parsed

Parameters

- **identifier** – (str) Identifier of the job
- **error_type** – (str) Result of the job

insert_success(*identifier*)

Creates a successful result file with the identifier passed

Parameters **identifier** – (str) Identifier of the job

ran_successfully(*identifier*)

Returns true / false on whether the result with this identifier is successful

Parameters **identifier** – (str) Identifier of the job result

Returns (bool) Boolean on if job ran successfully

validate()

Decorator to check if an identifier is of the correct format

HISTORY

4.1 1.2.0 (2021-03-17)

- Database connections now opened and closed for each method call

4.2 1.1.0 (2021-02-09)

- Changed module name from `backend` to `abcunit_backend`
- Removed need to parse `error_types` to either handler

4.3 1.0.0 (2020-11-06)

- First release

PYTHON MODULE INDEX

a

`abcunit_backend.database_handler`, [7](#)

`abcunit_backend.file_system_handler`, [8](#)

INDEX

A

abcunit_backend.database_handler
module, 7
abcunit_backend.file_system_handler
module, 8

C

count_failures() (abcunit_backend.database_handler.DatabaseHandler
method), 7
count_failures() (abcunit_backend.file_system_handler.FileSystemHandler
method), 8
count_results() (abcunit_backend.database_handler.DatabaseHandler
method), 7
count_results() (abcunit_backend.file_system_handler.FileSystemHandler
method), 8
count_successes() (abcunit_backend.database_handler.DatabaseHandler
method), 7
count_successes() (abcunit_backend.file_system_handler.FileSystemHandler
method), 8

D

DataBaseHandler (class in abcunit_backend.database_handler), 7
delete_all_results() (abcunit_backend.database_handler.DatabaseHandler
method), 7
delete_all_results() (abcunit_backend.file_system_handler.FileSystemHandler
method), 8
delete_result() (abcunit_backend.database_handler.DatabaseHandler
method), 7
delete_result() (abcunit_backend.file_system_handler.FileSystemHandler
method), 8

F

FileSystemHandler (class in abcunit_backend.file_system_handler), 8

G

get_all_results() (abcunit_backend.database_handler.DatabaseHandler
method), 7
get_all_results() (abcunit_backend.file_system_handler.FileSystemHandler
method), 8
get_failed_runs() (abcunit_backend.database_handler.DatabaseHandler
method), 7
get_failed_runs() (abcunit_backend.file_system_handler.FileSystemHandler
method), 8
get_result() (abcunit_backend.database_handler.DatabaseHandler
method), 7
get_result() (abcunit_backend.file_system_handler.FileSystemHandler
method), 8
get_successful_runs() (abcunit_backend.database_handler.DatabaseHandler
method), 7
get_successful_runs() (abcunit_backend.file_system_handler.FileSystemHandler
method), 8

I

insert_failure() (abcunit_backend.database_handler.DatabaseHandler
method), 7
insert_failure() (abcunit_backend.file_system_handler.FileSystemHandler
method), 9
insert_success() (abcunit_backend.database_handler.DatabaseHandler
method), 8
insert_success() (abcunit_backend.file_system_handler.FileSystemHandler
method), 9

M

module

 abcunit_backend.database_handler, 7
 abcunit_backend.file_system_handler, 8

R

ran_successfully() (abcunit_backend.database_handler.DatabaseHandler
 method), 8

ran_successfully() (abcunit_backend.file_system_handler.FileSystemHandler
 method), 9

V

validate() (abcunit_backend.file_system_handler.FileSystemHandler
 method), 9