

---

# **RoboTA-common-errors**

***Release 1.1.1***

**Peter Crowther**

**Oct 18, 2022**



# CONTENTS

<b>1</b>	<b>Installation</b>	<b>3</b>
<b>2</b>	<b>RoboTA Config</b>	<b>5</b>
2.1	API Reference . . . . .	5
2.1.1	RoboTA data types and data sources . . . . .	5
2.1.2	Getting started with RoboTA common errors . . . . .	6
2.1.3	Running the report . . . . .	6
2.1.4	Using common errors as a library . . . . .	6
2.1.5	Adding new common errors . . . . .	6
2.1.6	robota_common_errors . . . . .	7
	<b>Python Module Index</b>	<b>13</b>
	<b>Index</b>	<b>15</b>



RoboTA (Robot Teaching Assistant) is a group of Python packages that provide a framework for the assessment of software engineering practices. The focus of RoboTA is the assessment of student software engineering coursework, though it has a wider scope in the assessment of general good practice in software engineering.

The robota-core package collects information about a project from a number of sources, git repositories, issue trackers, ci-servers. It is designed to be provider agnostic, for example repository data can come from GitLab or GitHub.

The robota-common-errors package then uses this information to assess the project against a number of common software engineering errors or bad practices. Examples include committing non-project files to a git repository or merging a git branch into the remote tracking branch instead of the local one. The included errors are designed to be general and project agnostic but it would be easy to add new methods to enforce project or group specific standards such as a standard format for git commit messages or ensuring that issues are always assigned to an individual as soon as they are opened.

RoboTA was developed in the [Computer Science](#) department at the [University of Manchester](#). From 2018 to March 2021, development of RoboTA was funded by the [Institute of Coding](#).



## **INSTALLATION**

To install as a Python module, type

```
python -m pip install .
```

from the root directory. For developers, you should install in linked .egg mode using

```
python -m pip install robota-core -e
```

If you are using a Python virtual environment, you should activate this first before using the above commands.





## ROBOTA CONFIG

RoboTA requires access to a number of data sources to collect data to operate on.

Details of these data sources and information required to connect to them is provided in the robota config yaml file.

Documentation on the config file can be found in the [RoboTA data types and data sources](#) section of the documentation.

An example robota config file can be found in the [root of the robota-common-errors repository](#).

## 2.1 API Reference

### 2.1.1 RoboTA data types and data sources

RoboTA reads in various types of data from different sources. The main data source documentation is included in the [robota-core documentation](#). This documentation is supplementary to that, so you should read that documentation first.

robota-common-errors only makes use of a subset of the possible data types in robota-core and adds a single additional data type.

#### Data types

Issues, ci, repository and remote\_provider are valid data types from robota-core. They are documented in the [robota-core documentation](#).

There is one additional data type for robota-common-errors:

#### common-errors

The location of the yaml file that defines the common errors.

Valid sources: local\_path, gitlab

required keys:

- file\_name - The path of the file containing the mapping between names and email addresses

**This keys may specify sub-folder(s) in the git repository, e.g.**

file\_name: config\_files/error\_definitions.yaml

### Data Sources

Data sources are [unchanged from robota-core](#).

### 2.1.2 Getting started with RoboTA common errors

Once the robota-common-errors package has been installed you can either run the provided report which will output detected common errors to a HTML page, or you can import the functions and incorporate the outputs into your own workflow.

### 2.1.3 Running the report

The common-errors report is not intended as the primary method for displaying the detected common errors, but is provided as a simple example of a possible output.

To run the common errors report, from the common-errors root directory run:

```
python robota_common_errors/report.py
```

This will produce a report in the **webpages** folder called **common-error-report.html**.

Optional parameters for the script are:

- `--config_path` - This is the path to the robota config file. The default is **./robota-config.yaml**
- `--output` - This is the path where the report will be output. The default is **./webpages**
- `--start` - This is the start date of the report. Artifacts (commits, issues etc.) with a creation date older than this will not be considered in the report. The default is **2020-01-01**
- `--end` - This is the end date of the report, Artifacts newer than this will not be considered in the report. The default is **the current date**

### 2.1.4 Using common errors as a library

If you want to analyse common errors and use the data in your own scripts, you should use the `robota_common_errors.report.identify_common_errors()` function. This function takes the path to the robota config, start and end times as arguments and outputs a list of `CommonError`. These can then be parsed into whatever display format you want to use.

### 2.1.5 Adding new common errors

Common errors are described in the **error\_definitions.yaml** file. The **marking\_function** key in the error description is the name of the method in the code that will be called to assess the error. The code that actually assesses the error is located in the **common\_error\_functions.py** file. Functions to assess a common error must have the signature **marking\_function\_name(data\_source: DataServer, common\_error: CommonError)** and should populate the `CommonError` with the result and return this.

## 2.1.6 robota\_common\_errors

robota\_common\_errors package

Subpackages

robota\_common\_errors.output\_templates package

Submodules

robota\_common\_errors.output\_templates.build\_webpages module

robota\_common\_errors.output\_templates.build\_webpages.**build\_webpages**(*template\_dir: Path*,  
*output\_dir: Path*)

Create ancillary pages from templates and copy them to *output\_dir*. :param *template\_dir*: The path to load page templates from. :param *output\_dir*: The path to output generated pages to.

Module contents

Submodules

robota\_common\_errors.analysis module

An example of a simple data analysis script that might be run on the JSON output of common\_error\_collection.py.

robota\_common\_errors.analysis.**main**(*file\_name: str*)

robota\_common\_errors.analysis.**read\_json**(*file\_path: Path*) → DataFrame

robota\_common\_errors.common\_error\_collection module

**class** robota\_common\_errors.common\_error\_collection.**CommonErrorReport**(*course: str*, *year: str*,  
*team\_name: str*,  
*error\_name: str*,  
*detail\_titles: List[str]*,  
*error\_details: List[List[str]]*,  
*num\_occurrences: int*)

Bases: object

**course:** str

**detail\_titles:** List[str]

**error\_details:** List[List[str]]

**error\_name:** str

**num\_occurrences:** int

**team\_name:** str

**year:** str

robota\_common\_errors.common\_error\_collection.**main**(*config\_path: str, output\_name: str*)

robota\_common\_errors.common\_error\_collection.**output\_to\_json**(*file\_name: str,*  
*common\_error\_reports:*  
*List[CommonErrorReport]*)

### **robota\_common\_errors.common\_error\_functions module**

These functions specifically test the common errors.

robota\_common\_errors.common\_error\_functions.**commit\_unnecessary\_files**(*data\_source: DataServer,*  
*common\_error:*  
*CommonError*) →  
*CommonError*

Certain files shouldn't be committed to a Git repo, and should probably be specified in a .gitignore file

robota\_common\_errors.common\_error\_functions.**committing\_comments**(*data\_source: DataServer,*  
*common\_error: CommonError*)  
→ *CommonError*

Students often commit commented out code which is bad. This function checks the diffs for any java style commenting syntax.

robota\_common\_errors.common\_error\_functions.**committing\_conflict\_markup**(*data\_source:*  
*DataServer,*  
*common\_error:*  
*CommonError*) →  
*CommonError*

Conflicts should be resolved before committing. Any commit conflict markup which gets committed shows that the conflict has not been properly resolved.

robota\_common\_errors.common\_error\_functions.**confusing\_branch\_names**(*data\_source: DataServer,*  
*common\_error:*  
*CommonError*) →  
*CommonError*

Some branch names are obviously confusing and should be avoided. e.g. a local branch called origin/branch would have a remote tracking branch called origin/origin/branch

robota\_common\_errors.common\_error\_functions.**get\_comment\_line\_number**(*diff, line\_num\_in\_diff*)

Return starting line number of hunk which contains line\_num\_in\_diff

robota\_common\_errors.common\_error\_functions.**get\_commit\_diffs**(*commits: List[Commit],*  
*data\_source: DataServer*) →  
*Dict[Commit, List[Diff]]*

Get the diffs for each commit in commits.

While it would be fewer API calls to do a diff of all commits at once, the API refuses to return the diff if it is too big. This means doing it one commit at a time is safer.

robota\_common\_errors.common\_error\_functions.**get\_line\_num\_in\_diff**(*diff\_up\_to\_match: Diff, match:*  
*Match*)

Return line number in diff of a regex match object

`robota_common_errors.common_error_functions.get_subject_errors(message: str) → List[str]`

Detect errors with the subject line of git commit messages.

**Parameters**

**message** – Commit message

**Returns**

List of detected problems.

`robota_common_errors.common_error_functions.good_commit_messages(data_source: DataServer,  
common_error: CommonError) →  
CommonError`

Git commit messages should have a standardised formatting.

`robota_common_errors.common_error_functions.merge_remote_branch(data_source: DataServer,  
common_error: CommonError)  
→ CommonError`

Identify cases where students have merged a remote tracking branch. They should be pulling *before* making local commits or rebasing their local branch onto the remote branch.

`robota_common_errors.common_error_functions.repeated_revert(data_source: DataServer,  
common_error: CommonError) →  
CommonError`

Identify cases where students have repeatedly attempted to revert commits. This is often because they think that a commit revert is like an undo command.

`robota_common_errors.common_error_functions.wrong_use_of_estimate(data_source: DataServer,  
common_error: CommonError) →  
CommonError`

Identify cases where students have misused the `/estimate` command when trying to record time estimates for a gitlab issue.

`robota_common_errors.common_error_functions.wrong_use_of_spend(data_source: DataServer,  
common_error: CommonError)  
→ CommonError`

Identify cases where students have misused the `/spend` command when trying to record time spent working on a gitlab issue.

`robota_common_errors.common_error_functions.wrong_way_merge(data_source: DataServer,  
common_error: CommonError) →  
CommonError`

Identify cases where students have merged develop into their feature branch rather than the feature branch into master.

**robota\_common\_errors.common\_errors module**

Common errors are suboptimal student behaviors that occur regularly. If a common error is identified, feedback is given on how to correct it or avoid it in future.

Common errors are much like Tasks except they are not assigned any credit.

**class** robota\_common\_errors.common\_errors.**CommonError**(*common\_error: dict*)

Bases: object

Representation of a common error that could be identified

**Variables**

- **name** – The name of the common error.
- **marking\_function** – The name of the function used to assess the common error.
- **description** – A textual description of the common error.
- **detail\_titles** – Titles of a table to be printed on the marking report.
- **error\_details** – Rows of a table to be printed on the marking report.

**add\_feedback**(*feedback: List[List[str]]*)

**add\_feedback\_titles**(*feedback\_titles: List[str]*)

**count\_errors**() → int

Sum the occurrences of this error by checking error details.

robota\_common\_errors.common\_errors.**assess\_common\_errors**(*data\_source: DataServer, common\_errors: List[CommonError]*) → List[CommonError]

Go through all of the possible common errors, running the assessment function for each one.

robota\_common\_errors.common\_errors.**get\_error\_descriptions**(*robota\_config: dict*) → List[CommonError]

Get the textual description of common errors to identify.

robota\_common\_errors.common\_errors.**validate\_data\_sources**(*data\_source: DataServer, common\_error: CommonError*)

Check the data sources that are available from the DataServer.

**robota\_common\_errors.report module**

Generates a HTML report of common errors for a git repo.

robota\_common\_errors.report.**count\_errors**(*common\_errors: List[CommonError]*) → int

Count how many different common error types were detected.

robota\_common\_errors.report.**identify\_common\_errors**(*robota\_config: dict, start: datetime, end: datetime*) → List[CommonError]

The main function which gets the common errors.

robota\_common\_errors.report.**output\_html\_report**(*common\_errors: List[CommonError], data\_source\_info: dict, common\_error\_summary: dict, output\_dir: str*)

`robota_common_errors.report.run_html_error_report(start: str, end: str, config_path: str, output_dir: str, substitution_variables: dict)`

Get common errors and output them in the form of a HTML report.

`robota_common_errors.report.summarise_common_errors(common_errors: List[CommonError])`

`robota_common_errors.report.summarise_data_sources(robota_config: dict, start: datetime, end: datetime) → dict`

`robota_common_errors.report.update_template(common_errors: List[CommonError], data_source_info: dict, common_error_summary: dict)`

Produce the HTML report by writing the marking results to a HTML template.

#### Parameters

- **common\_errors** – A list of common errors and feedback on them.
- **data\_source\_info** – A dictionary of information about the data sources that were used.
- **common\_error\_summary** – A dictionary of stats about the common errors that is printed in the report.

## Module contents

- `genindex`





## PYTHON MODULE INDEX

### r

- `robota_common_errors`, 11
- `robota_common_errors.analysis`, 7
- `robota_common_errors.common_error_collection`,  
7
- `robota_common_errors.common_error_functions`,  
8
- `robota_common_errors.common_errors`, 10
- `robota_common_errors.output_templates`, 7
- `robota_common_errors.output_templates.build_webpages`,  
7
- `robota_common_errors.report`, 10



## INDEX

### A

`add_feedback()` (rob-  
     `ota_common_errors.common_errors.CommonError`  
     method), 10

`add_feedback_titles()` (rob-  
     `ota_common_errors.common_errors.CommonError`  
     method), 10

`assess_common_errors()` (in module rob-  
     `ota_common_errors.common_errors`), 10

### B

`build_webpages()` (in module rob-  
     `ota_common_errors.output_templates.build_webpages`),  
     7

### C

`commit_unnecessary_files()` (in module rob-  
     `ota_common_errors.common_error_functions`),  
     8

`committing_comments()` (in module rob-  
     `ota_common_errors.common_error_functions`),  
     8

`committing_conflict_markup()` (in module rob-  
     `ota_common_errors.common_error_functions`),  
     8

`CommonError` (class in rob-  
     `ota_common_errors.common_errors`), 10

`CommonErrorReport` (class in rob-  
     `ota_common_errors.common_error_collection`),  
     7

`confusing_branch_names()` (in module rob-  
     `ota_common_errors.common_error_functions`),  
     8

`count_errors()` (in module rob-  
     `ota_common_errors.report`), 10

`count_errors()` (rob-  
     `ota_common_errors.common_errors.CommonError`  
     method), 10

`course` (robota\_common\_errors.common\_error\_collection.  
     attribute), 7

### D

`detail_titles` (robota\_common\_errors.common\_error\_collection.CommonError.  
     attribute), 7

### E

`error_details` (robota\_common\_errors.common\_error\_collection.CommonError.  
     attribute), 7

`error_name` (robota\_common\_errors.common\_error\_collection.CommonError.  
     attribute), 7

### G

`get_comment_line_number()` (in module rob-  
     `ota_common_errors.common_error_functions`),  
     8

`get_commit_diffs()` (in module rob-  
     `ota_common_errors.common_error_functions`),  
     8

`get_error_descriptions()` (in module rob-  
     `ota_common_errors.common_errors`), 10

`get_line_num_in_diff()` (in module rob-  
     `ota_common_errors.common_error_functions`),  
     8

`get_subject_errors()` (in module rob-  
     `ota_common_errors.common_error_functions`),  
     8

`good_commit_messages()` (in module rob-  
     `ota_common_errors.common_error_functions`),  
     9

### I

`identify_common_errors()` (in module rob-  
     `ota_common_errors.report`), 10

### M

`main()` (in module robota\_common\_errors.analysis), 7

`main()` (in module rob-  
     `ota_common_errors.common_error_collection`),  
     8

`merge_or_remove_branch()` (in module rob-  
     `ota_common_errors.common_error_functions`),  
     9

module

robota\_common\_errors, 11  
robota\_common\_errors.analysis, 7  
robota\_common\_errors.common\_error\_collection, 7  
robota\_common\_errors.common\_error\_functions, 8  
robota\_common\_errors.common\_errors, 10  
robota\_common\_errors.output\_templates, 7  
robota\_common\_errors.output\_templates.build\_webpages, 7  
robota\_common\_errors.report, 10

**N**  
num\_occurrences (robota\_common\_errors.common\_error\_collection.CommonErrorReport attribute), 7

**O**  
output\_html\_report() (in module robota\_common\_errors.report), 10  
output\_to\_json() (in module robota\_common\_errors.common\_error\_collection), 8

**R**  
read\_json() (in module robota\_common\_errors.analysis), 7  
repeated\_revert() (in module robota\_common\_errors.common\_error\_functions), 9  
robota\_common\_errors module, 11  
robota\_common\_errors.analysis module, 7  
robota\_common\_errors.common\_error\_collection module, 7  
robota\_common\_errors.common\_error\_functions module, 8  
robota\_common\_errors.common\_errors module, 10  
robota\_common\_errors.output\_templates module, 7  
robota\_common\_errors.output\_templates.build\_webpages module, 7  
robota\_common\_errors.report module, 10  
run\_html\_error\_report() (in module robota\_common\_errors.report), 10

**S**  
summarise\_common\_errors() (in module robota\_common\_errors.report), 11  
summarise\_data\_sources() (in module robota\_common\_errors.report), 11

**T**  
team\_name(robota\_common\_errors.common\_error\_collection.CommonErrorReport attribute), 7  
update\_template() (in module robota\_common\_errors.report), 11  
validate\_data\_sources() (in module robota\_common\_errors.common\_errors), 10  
wrong\_use\_of\_estimate() (in module robota\_common\_errors.common\_error\_functions), 9  
wrong\_use\_of\_spend() (in module robota\_common\_errors.common\_error\_functions), 9  
wrong\_way\_merge() (in module robota\_common\_errors.common\_error\_functions), 9  
year(robota\_common\_errors.common\_error\_collection.CommonErrorReport attribute), 8