
terraformtestinglib Documentation

Release 1.4.5

Costas Tyfoxylos

Jun 08, 2021

Contents

1	terraformtestinglib	3
1.1	Development Workflow	3
1.2	Important Information	4
1.3	Project Features	4
2	Installation	5
3	Usage	7
4	Contributing	9
4.1	Submit Feedback	9
5	terraformtestinglib	11
5.1	terraformtestinglib package	11
6	Credits	25
6.1	Development Lead	25
6.2	Contributors	25
7	History	27
8	0.1.0 (24-05-2018)	29
9	1.0.0 (16-10-2018)	31
10	1.0.3 (17-10-2018)	33
11	1.0.4 (25-10-2018)	35
12	1.1.0 (07-01-2019)	37
13	1.1.1 (14-01-2019)	39
14	1.1.2 (18-01-2019)	41
15	1.2.0 (19-01-2019)	43
16	1.2.1 (20-01-2019)	45

17	1.2.2 (22-01-2019)	47
18	1.2.3 (22-01-2019)	49
19	1.3.0 (06-02-2019)	51
20	1.4.0 (07-02-2019)	53
21	1.4.1 (07-02-2019)	55
22	1.4.2 (22-10-2019)	57
23	1.4.3 (22-10-2019)	59
24	1.4.4 (22-05-2020)	61
25	1.4.5 (25-05-2020)	63
26	Indices and tables	65
	Python Module Index	67
	Index	69

Contents:

A library that implements linting and testing of terraform resources based on rules.

- Documentation: <https://terraformtestinglib.readthedocs.org/en/latest>

1.1 Development Workflow

The workflow supports the following steps

- lint
- test
- build
- document
- upload
- graph

These actions are supported out of the box by the corresponding scripts under `_CI/scripts` directory with sane defaults based on best practices. Sourcing `setup_aliases.ps1` for windows powershell or `setup_aliases.sh` in bash on Mac or Linux will provide with handy aliases for the shell of all those commands prepended with an underscore.

The bootstrap script creates a `.venv` directory inside the project directory hosting the virtual environment. It uses `pipenv` for that. It is called by all other scripts before they do anything. So one could simple start by calling `_lint` and that would set up everything before it tried to actually lint the project

Once the code is ready to be delivered the `_tag` script should be called accepting one of three arguments, `patch`, `minor`, `major` following the semantic versioning scheme. So for the initial delivery one would call

```
$ _tag --minor
```

which would bump the version of the project to 0.1.0 tag it in git and do a push and also ask for the change and automagically update `HISTORY.rst` with the version and the change provided.

So the full workflow after git is initialized is:

- repeat as necessary (of course it could be test - code - lint :)) * code * lint * test
- commit and push
- develop more through the code-lint-test cycle
- tag (with the appropriate argument)
- build
- upload (if you want to host your package in pypi)
- document (of course this could be run at any point)

1.2 Important Information

This template is based on pipenv. In order to be compatible with requirements.txt so the actual created package can be used by any part of the existing python ecosystem some hacks were needed. So when building a package out of this **do not** simple call

```
$ python setup.py sdist bdist_egg
```

as this will produce an unusable artifact with files missing. Instead use the provided build and upload scripts that create all the necessary files in the artifact.

1.3 Project Features

- Lints terraform files based on provided rules
- Enforces positioning of resources on files based on provided rules

CHAPTER 2

Installation

At the command line:

```
$ pip install terraformtestinglib
```

Or, if you have virtualenvwrapper installed:

```
$ mkvirtualenv terraformtestinglib  
$ pip install terraformtestinglib
```

Or, if you are using pipenv:

```
$ pipenv install terraformtestinglib
```


CHAPTER 3

Usage

To develop on terraformtestinglib:

```
# The following commands require pipenv as a dependency

# To lint the project
_CI/scripts/lint.py

# To execute the testing
_CI/scripts/test.py

# To create a graph of the package and dependency tree
_CI/scripts/graph.py

# To build a package of the project under the directory "dist/"
_CI/scripts/build.py

# To see the package version
_CI/scripts/tag.py

# To bump semantic versioning [--major|--minor|--patch]
_CI/scripts/tag.py --major|--minor|--patch

# To upload the project to a pypi repo if user and password are properly provided
_CI/scripts/upload.py

# To build the documentation of the project
_CI/scripts/document.py
```

To use terraformtestinglib in a project for linting:

```
from terraformtestinglib import Stack
stack = Stack('path_to_tf_files',
              'path_to_naming_yaml',
              'optional_path_to_positioning_yaml',
```

(continues on next page)

(continued from previous page)

```

        'optional_path_to_global.tfvars')
stack.validate()
for error in stack.errors:
    print(error)

```

```

# naming.yaml should follow the following schema
#
# Schema([{'resource': basestring,
#         'regex': is_valid_regex,
#         Optional('fields'): [{'value': basestring,
#                               'regex': is_valid_regex}]}])
#
# Example
---
- resource: terraform_resource_name
  regex: .* # regex to lint terraform id
  fields:
    - value: tags.Name
      regex: ^cust[dtaps](?:ewl)-pc[0-9]{2}$ # regex to lint the name of the tag
    - value: tags.Other
      regex: ^cust[dtaps](?:ewl)-other[0-9]{2}$ # regex to lint the name of the tag

```

```

# positioning.yaml should follow the following schema
#
# Schema({And(basestring, lambda x: x.endswith('.tf')): [basestring]})
#
# Example
---
_data.tf:
  - terraform
  - data
_provider.tf:
  - provider
_variables.tf:
  - variable
_compute.tf:
  - azurerm_app_service
  - azurerm_app_service_plan
  - azurerm_virtual_machine
  - aws_instance

```

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given.

4.1 Submit Feedback

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.

4.1.1 Get Started!

Ready to contribute? Here's how to set up *terraformtestinglib* for local development.

1. Clone your fork locally:

```
$ git clone https://github.com/schubergphilis/terraformtestinglib.git
```

2. Install your local copy into a virtualenv. Assuming you have virtualenvwrapper installed, this is how you set up your clone for local development:

```
$ mkvirtualenv terraformtestinglib
$ cd terraformtestinglib/
$ python setup.py develop
```

3. Create a branch for local development:

```
$ git checkout -b name-of-your-bugfix-or-feature
```

Now you can make your changes locally.

4. Commit your changes and push your branch to the server:

```
$ git add .  
$ git commit -m "Your detailed description of your changes."  
$ git push origin name-of-your-bugfix-or-feature
```

5. Submit a merge request

5.1 terraformtestinglib package

5.1.1 Subpackages

terraformtestinglib.linting package

Submodules

terraformtestinglib.linting.linting module

Main code for linting.

```
class terraformtestinglib.linting.linting.LintingResource (filename, re-  
 source_type, name,  
 data, original_data)
```

Bases: object

Manages a resource and provides validation capabilities..

register_positioning_set (*positioning_set*)

Registers the set of rules with the Resource.

Parameters **positioning_set** (*dict*) – A dictionary with the rules for the positioning convention

Returns None

register_rules_set (*rules_set*)

Registers the set of rules with the Resource.

Parameters **rules_set** (*dict*) – A dictionary with the rules for the naming convention

Returns None

validate()

Validates the resource according to the appropriate rule.

Returns True upon completion

class terraformtestinglib.linting.linting.**Rule**(*data*)

Bases: object

Handles the rule object providing validation capabilities.

errors

List of errors found.

Returns: The errors found

validate(*resource_type, resource_name, resource_data, original_data*)

Validates the given resource based on the ruleset.

Parameters

- **resource_type** (*basestring*) – The type of the resource
- **resource_name** (*basestring*) – The name of the resource
- **resource_data** (*dict*) – The interpolated data of the resource
- **original_data** (*dict*) – The original data of the resource, before the interpolation

Returns True on successful validation, False otherwise

class terraformtestinglib.linting.linting.**RuleSet**(*rules*)

Bases: object

Manages the rules as a group and can search them by name.

get_rule_for_resource(*resource_name*)

Retrieves the rule for the resource name.

Parameters **resource_name** (*basestring*) – The resource type to retrieve the rule for

Returns The rule corresponding with the resource type if found, None otherwise

class terraformtestinglib.linting.linting.**Stack**(*configuration_path, naming_file_path, positioning_file_path=None, global_variables_file_path=None, file_to_skip_for_positioning=None, raise_on_missing_variable=True, environment_variables=None*)

Bases: *terraformtestinglib.terraformtestinglib.Parser*

Manages a stack as a collection of resources that can be checked for name convention.

errors

The errors of the validation of the resources of the stack.

Returns list of possible linting errors

Return type errors (ResourceError|FilenameError)

validate()

Validates all the resources of the stack.

Returns None

Module contents

terraformtestinglib.linting package.

Import all parts from terraformtestinglib.linting here

terraformtestinglib.testing package

Submodules

terraformtestinglib.testing.testing module

Main code for testing.

class terraformtestinglib.testing.testing.**Attribute** (*resource, name, value*)

Bases: object

Models the attribute.

resource_name

Exposes the name of the parent resource object.

Returns The name of the parent resource

Return type name (basestring)

resource_type

Exposes the type of the parent resource object.

Returns The type of the parent resource

Return type type (basestring)

class terraformtestinglib.testing.testing.**AttributeList** (*validator, attributes*)

Bases: object

Object containing attribute objects and providing validation methods for them.

attribute (*name*)

Filters attributes on matching the provided name.

Parameters **name** (*basestring*) – The name to match the attribute with

Returns A container of attribute objects

Return type *AttributeList*

if_has_attribute_with_value (*attribute, value*)

Filters the AttributeList based on the provided attribute and value.

Parameters

- **attribute** – The attribute to filter on
- **value** – the value of the attribute to filter on

Returns A container of attribute objects

Return type *AttributeList*

if_not_has_attribute_with_value (*attribute, value*)

Filters the AttributeList based on the provided attribute and value.

Parameters

- **attribute** – The attribute to filter on
- **value** – the value of the attribute to filter on

Returns A container of attribute objects

Return type *AttributeList*

should_be_valid_json()

Checks whether the value for the attribute is valid json.

Raises `AssertionError` – If any errors are found on the check

Returns `None`

should_equal(*value*)

Checks for equality for the provided value from all contained attributes.

Parameters **value** – The value to match with

Raises `AssertionError` – If any errors are found on the check

Returns `None`

should_have_attributes(*attributes*)

Checks for existence for the provided attribute from all contained attributes.

Parameters **attributes** – An attribute or list of attributes to check for

Raises `AssertionError` – If any errors are found on the check

Returns `None`

should_match_regex(*regex*)

Checks for regular expression match from all contained attributes.

Parameters **regex** (*basestring*) – A regular expression to match with

Raises `AssertionError` – If any errors are found on the check

Returns `None`

should_not_equal(*value*)

Checks for inequality for the provided value from all contained attributes.

Parameters **value** – The value to not match with

Raises `AssertionError` – If any errors are found on the check

Returns `None`

should_not_have_attributes(*attributes*)

Checks for lack for the provided attribute from all contained attributes.

Parameters **attributes** – An attribute or list of attributes to check for

Raises `AssertionError` – If any errors are found on the check

Returns `None`

should_not_match_regex(*regex*)

Checks for regular expression not matching from all contained attributes.

Parameters **regex** (*basestring*) – A regular expression to not match with

Raises `AssertionError` – If any errors are found on the check

Returns `None`

class terraformtestinglib.testing.testing.**Container** (*validator_instance, entities*)

Bases: object

An object handling the exposing of attributes of different resources of terraform.

attribute (*name*)

Filters attributes based on the provided name.

Parameters **name** (*basestring*) – The name to match against

Raises `AssertionError` – If any errors are calculated

Returns An object containing any attributes matching the check

Return type *AttributeList* (list)

attribute_matching_regex (*regex*)

Filters attributes based on the provided regex.

Parameters **regex** (*basestring*) – A basestring of a valid regular expression to match against

Raises `AssertionError` – If any errors are calculated

Returns An object containing any attributes matching the check

Return type *AttributeList* (list)

if_has_attribute (*attribute*)

Filters the entities based on the provided attribute.

Parameters **attribute** (*basestring*) – The attribute to filter the resources on

Returns An entities list object with all resources following the pattern

Return type (list)

if_has_attribute_with_regex_value (*attribute, regex*)

Filters the entities based on the provided attribute and value.

Parameters

- **attribute** (*basestring*) – The attribute to filter the entities on if the value matches the regex provided
- **regex** – The regex to match with

Returns An entities list object with all entities following the pattern

Return type (list))

if_has_attribute_with_value (*attribute, value*)

Filters the entities based on the provided attribute and value.

Parameters

- **attribute** (*basestring*) – The attribute to filter the entities on
- **value** – The value to match with

Returns An entities list object with all entities following the pattern

Return type (list)

if_has_subattribute (*parent_attribute, attribute*)

Filters the entities based on the provided parent and child attribute.

Parameters

- **parent_attribute** (*basestring*) – The parent attribute to filter the resources on
- **attribute** (*basestring*) – The child attribute to filter the entities on if it exists

Returns An entities list object with all entities following the pattern

Return type (list)

if_has_subattribute_with_regex_value (*parent_attribute, attribute, regex*)

Filters the entities based on the provided parent and child attribute and regex for value matching.

Parameters

- **parent_attribute** (*basestring*) – The parent attribute to filter the entities on
- **attribute** (*basestring*) – The child attribute to filter the entities on
- **regex** – The regex to match with for the child attribute's value

Returns An entities list object with all entities following the pattern

Return type (list)

if_has_subattribute_with_value (*parent_attribute, attribute, value*)

Filters the entities based on the provided parent and child attribute and value.

Parameters

- **parent_attribute** (*basestring*) – The parent attribute to filter the entities on
- **attribute** (*basestring*) – The child attribute to filter the entities on
- **value** – The value to match with for the child attribute

Returns An entities list object with all entities following the pattern

Return type (list)

if_not_has_attribute (*attribute*)

Filters the entities based on the non existence of the provided attribute.

Parameters **attribute** (*basestring*) – The attribute to filter the resources on

Returns An entities list object with all entities following the pattern

Return type (list)

if_not_has_attribute_with_regex_value (*attribute, regex*)

Filters the entities based on the provided attribute and value.

Parameters

- **attribute** (*basestring*) – The attribute to filter the entities on if the value does not match the regex
- **regex** – The regex not to match with

Returns An entities list object with all entities following the pattern

Return type (list)

if_not_has_attribute_with_value (*attribute, value*)

Filters the entities based on the provided attribute and value.

Parameters

- **attribute** (*basestring*) – The attribute to filter the resources on
- **value** – The value to not match

Returns An entities list object with all entities following the pattern

Return type (list)

if_not_has_subattribute (*parent_attribute, attribute*)

Filters the entities based on the provided parent and child attribute.

Parameters

- **parent_attribute** (*basestring*) – The parent attribute to filter the entities on
- **attribute** (*basestring*) – The child attribute to filter the entities on if it does not exists

Returns An entities list object with all entities following the pattern

Return type (list)

if_not_has_subattribute_with_regex_value (*parent_attribute, attribute, regex*)

Filters the entities based on the provided parent and child attribute and regex for value matching.

Parameters

- **parent_attribute** (*basestring*) – The parent attribute to filter the entities on
- **attribute** (*basestring*) – The child attribute to filter the entities on
- **regex** – The regex to not match with for the child attribute's value

Returns An entities list object with all entities following the pattern

Return type (list)

if_not_has_subattribute_with_value (*parent_attribute, attribute, value*)

Filters the entities based on the provided parent and child attribute and value.

Parameters

- **parent_attribute** (*basestring*) – The parent attribute to filter the entities on
- **attribute** (*basestring*) – The child attribute to filter the entities on
- **value** – The value to not match with for the child attribute

Returns An entities list object with all entities following the pattern

Return type (list)

should_have_attributes (*attributes_list*)

Validates that the resource has the provided arguments which are always cast to a list.

Parameters **attributes_list** (*list*) – A list of strings for attributes to check against

Raises `AssertionError` – If any errors are calculated

Returns None

should_not_have_attributes (*attributes_list*)

Validates that the resource does not have the provided arguments which are always cast to a list.

Parameters **attributes_list** (*list*) – A list of strings for attributes to check against

Raises `AssertionError` – If any errors are calculated

Returns None

```

class terraformtestinglib.testing.testing.Data (type: str, name: str, data: Any)
    Bases: terraformtestinglib.testing.testing.Entity

    Basic model of a data object exposing required attributes.

class terraformtestinglib.testing.testing.DataList (validator_instance, entities)
    Bases: terraformtestinglib.testing.testing.Container

    A list of data objects being capable to filter on specific requirements.

class terraformtestinglib.testing.testing.Entity (type: str, name: str, data: Any)
    Bases: object

    Basic model of an entity exposing required attributes.

class terraformtestinglib.testing.testing.Provider (type: str, name: str, data: Any)
    Bases: terraformtestinglib.testing.testing.Entity

    Basic model of a provider object exposing required attributes.

class terraformtestinglib.testing.testing.ProviderList (validator_instance, entities)
    Bases: terraformtestinglib.testing.testing.Container

    A list of provider objects being capable to filter on specific requirements.

class terraformtestinglib.testing.testing.Resource (type: str, name: str, data: Any)
    Bases: terraformtestinglib.testing.testing.Entity

    Basic model of a resource exposing required attributes.

class terraformtestinglib.testing.testing.ResourceList (validator_instance, entities)
    Bases: terraformtestinglib.testing.testing.Container

    A list of resource objects being capable to filter on specific requirements.

resources (type_)
    Filters resources based on resource type which is always cast to list.

        Parameters type (list/basestring) – The type of resources to filter on. Always gets
            cast to list.

        Raises AssertionError – If any errors are calculated

        Returns An object containing any resources matching the type

        Return type ResourceList (list)

class terraformtestinglib.testing.testing.Terraform (type: str, name: str, data: Any)
    Bases: terraformtestinglib.testing.testing.Entity

    Basic model of a provider object exposing required attributes.

class terraformtestinglib.testing.testing.TerraformList (validator_instance, enti-
    ties)
    Bases: terraformtestinglib.testing.testing.Container

    A list of terraform objects being capable to filter on specific requirements.

class terraformtestinglib.testing.testing.Validator (configuration_path,
    global_variables_file_path=None,
    raise_on_missing_variable=True,
    environment_variables=None)
    Bases: terraformtestinglib.terraformtestinglib.Parser

    Object exposing resources and variables of terraform plans.

```

data (*type_*)

Filters data based on data type which is always cast to list.

Parameters **type** (*basestring/list*) – The type of data attributes to filter on. Always gets cast to a list.

Returns An object containing the data matching the type provided

Return type *DataList*

get_variable_value (*variable*)

Retrieves the variable value from the global view state.

Parameters **variable** (*basestring*) – The variable to retrieve the value for

Returns The value of the retrieved variable

Return type *value*

provider (*type_*)

Filters providers based on provider type which is always cast to list.

Parameters **type** (*basestring/list*) – The type of provider to filter on. Always gets cast to a list.

Returns An object containing the providers matching the type provided

Return type *ProviderList*

resources (*type_*)

Filters resources based on resource type which is always cast to list.

Parameters **type** (*basestring/list*) – The type of resources to filter on. Always gets cast to a list.

Returns An object containing the resources matching the type provided

Return type *ResourceList*

terraform (*type_*)

Filters terraform entries based on provided type which is always cast to list.

Parameters **type** (*basestring/list*) – The type of terraform attributes to filter on. Always gets cast to a list.

Returns An object containing the terraform objects matching the type provided

Return type *TerraformList*

static to_list (*value*)

Casts to list the provided argument if not a list already.

Parameters **value** (*basestring/list*) – Casts the provided value to list if not already

Returns A list of the value or values

Return type *value (list)*

variable (*name*)

Returns a variable object of the provided name.

Parameters **name** (*basestring*) – The name of the variable to retrieve

Returns An object modeling a variable

Return type *Variable*

class terraformtestinglib.testing.testing.**Variable** (*name, value*)

Bases: object

Models a variable and exposes basic test for it.

value_equals (*value*)

Checks that the value equals the provided value.

Raises AssertionError – If any errors are found on the check

Returns None

value_exists ()

Checks that the value exists.

Raises AssertionError – If any errors are found on the check

Returns None

value_matches_regex (*regex*)

Checks that the value matches the provided regex.

Raises AssertionError – If any errors are found on the check

Returns None

terraformtestinglib.testing.testing.**assert_on_error** (*func*)

Raises assertion error exceptions if the wrapped method returned any errors.

Module contents

terraformtestinglib.testing package.

Import all parts from terraformtestinglib.testing here

terraformtestinglib.utils package

Submodules

terraformtestinglib.utils.errortypes module

Main code for errortypes.

class terraformtestinglib.utils.errortypes.**ConfigurationError** (*resource_type,*
entity, *field,*
regex, *value,*
original_value)

Bases: tuple

entity

Alias for field number 1

field

Alias for field number 2

original_value

Alias for field number 5

regex

Alias for field number 3

resource_type
Alias for field number 0

value
Alias for field number 4

class terraformtestinglib.utils.errortypes.**FilenameError** (*filename, resource, target*)

Bases: object

Models the Filename error and provides a nice printed version.

class terraformtestinglib.utils.errortypes.**ResourceError** (*filename, resource, entity, field, regex, value, original_value*)

Bases: object

Models the Resource error and provides a nice printed version.

class terraformtestinglib.utils.errortypes.**RuleError** (*resource_type, entity, field, regex, value, original_value*)

Bases: tuple

entity
Alias for field number 1

field
Alias for field number 2

original_value
Alias for field number 5

regex
Alias for field number 3

resource_type
Alias for field number 0

value
Alias for field number 4

terraformtestinglib.utils.utils module

Main code for utils.

class terraformtestinglib.utils.utils.**RecursiveDictionary**

Bases: dict

Implements recursively updating dictionary.

RecursiveDictionary provides the methods update and iter_rec_update that can be used to update member dictionaries rather than overwriting them.

iter_rec_update (*iterator*)
Updates recursively.

update (*other, **third*)
Implements the recursion.

Recursively update the dictionary with the contents of other and third like dict.update() does - but don't overwrite sub-dictionaries.

Module contents

terraformtestinglib.utils package.

Import all parts from terraformtestinglib.utils here

5.1.2 Submodules

5.1.3 terraformtestinglib.configuration module

Main code for configuration.

`terraformtestinglib.configuration.is_valid_regex(value)`
Validates a regex.

5.1.4 terraformtestinglib.terraformtestinglib module

Main code for terraformtestinglib.

```
class terraformtestinglib.terraformtestinglib.HclFileResource(filename, resource_type, resource_name, data)

Bases: tuple

data
    Alias for field number 3

filename
    Alias for field number 0

resource_name
    Alias for field number 2

resource_type
    Alias for field number 1

class terraformtestinglib.terraformtestinglib.HclView(hcl_resources, global_variables=None, raise_on_missing_variable=True, environment_variables=None)
```

Bases: object

Object representing the global view of hcl resources along with any global variables.

get_counter_resource_data_by_type(resource_type, resource_name)
Retrieves the data of a resource from the global hcl state based on its type that has a count.

Parameters

- **resource_type** (*basestring*) – The resource type to retrieve the data for
- **resource_name** (*basestring*) – The resource name to retrieve the data for

Returns Original non interpolated data (dict) for the provided resource name and resource type

get_resource_data_by_type(resource_type, resource_name)
Retrieves the data of a resource from the global hcl state based on its type.

Parameters

- **resource_type** (*basestring*) – The resource type to retrieve the data for
- **resource_name** (*basestring*) – The resource name to retrieve the data for

Returns Interpolated data (dict) for the provided resource name and resource type

get_variable_value (*variable*)

Retrieves the value of a variable from the global view of variables.

Parameters () (*variable*) – The variable to look for

Raises MissingValue – If the value does not exist

Returns The value retrieved

Return type value (str)

```
class terraformtestinglib.terraformtestinglib.Parser (configuration_path,
                                                    global_variables_file_path=None,
                                                    raise_on_missing_variable=True,
                                                    environment_variables=None)
```

Bases: object

Manages the parsing of terraform files and creating the global hcl view from them.

```
terraformtestinglib.terraformtestinglib.warning_on_one_line (message, category,
                                                            filename, lineno,
                                                            line=None)
```

Warning formatting method.

5.1.5 terraformtestinglib.terraformtestinglibexceptions module

Custom exception code for terraformtestinglib.

exception terraformtestinglib.terraformtestinglibexceptions.**InvalidNaming**

Bases: Exception

The rules file provided was invalid.

exception terraformtestinglib.terraformtestinglibexceptions.**InvalidPositioning**

Bases: Exception

The structure file provided was invalid.

exception terraformtestinglib.terraformtestinglibexceptions.**MissingVariable**

Bases: Exception

The variable is missing.

5.1.6 Module contents

terraformtestinglib package.

Import all parts from terraformtestinglib here

CHAPTER 6

Credits

6.1 Development Lead

- Costas Tyfoxylos <ctyfoxylos@schubergphilis.com>

6.2 Contributors

None yet. Why not be the first?

CHAPTER 7

History

CHAPTER 8

0.1.0 (24-05-2018)

- First release

CHAPTER 9

1.0.0 (16-10-2018)

- Implemented variable, count attribute and format method interpolation on both linting and testing capabilities
- Implemented testing capabilities with conditional filtering for resources
- Ported the pipeline portion to python 3.7
- Dropped official support for python2.7

CHAPTER 10

1.0.3 (17-10-2018)

- Implemented interactive setting of the changelog in HISTORY.rst file on tagging

CHAPTER 11

1.0.4 (25-10-2018)

- Updated template and dependencies

CHAPTER 12

1.1.0 (07-01-2019)

- Added support for attributes with same name and filtering attributes on value

CHAPTER 13

1.1.1 (14-01-2019)

- Correctly handle lists in resource data.

CHAPTER 14

1.1.2 (18-01-2019)

- Casting to string for replacement in case it is a number

CHAPTER 15

1.2.0 (19-01-2019)

- Added support for “length” method and multi variable strings

CHAPTER 16

1.2.1 (20-01-2019)

- fixed bug where count was a string breaking the range calculation

CHAPTER 17

1.2.2 (22-01-2019)

- added support for multiple same keys that end up being handled as a list internally.

CHAPTER 18

1.2.3 (22-01-2019)

- added capabilities to skip a test based on a “skip-testing” tag on the resource

CHAPTER 19

1.3.0 (06-02-2019)

- implemented all terraform supported entities like, data, terraform and provider.

CHAPTER 20

1.4.0 (07-02-2019)

- implemented skipping positioning checking for a disaster_recovery.tf file. Refactored container object to expose filtering.

CHAPTER 21

1.4.1 (07-02-2019)

- fixed instantiation of Stack object

CHAPTER 22

1.4.2 (22-10-2019)

- Updated template and bumped dependencies.

CHAPTER 23

1.4.3 (22-10-2019)

- Fixed yaml deprecation errors and breakage of format method.

CHAPTER 24

1.4.4 (22-05-2020)

- Bumped dependencies, getting terraform 12 compatibility.

CHAPTER 25

1.4.5 (25-05-2020)

- bumped dependencies

CHAPTER 26

Indices and tables

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

t

- `terraformtestinglib`, [23](#)
- `terraformtestinglib.configuration`, [22](#)
- `terraformtestinglib.linting`, [13](#)
- `terraformtestinglib.linting.linting`, [11](#)
- `terraformtestinglib.terraformtestinglib`,
[22](#)
- `terraformtestinglib.terraformtestinglibexceptions`,
[23](#)
- `terraformtestinglib.testing`, [20](#)
- `terraformtestinglib.testing.testing`, [13](#)
- `terraformtestinglib.utils`, [22](#)
- `terraformtestinglib.utils.errortypes`,
[20](#)
- `terraformtestinglib.utils.utils`, [21](#)

A

`assert_on_error()` (in module `terraformtestinglib.testing.testing`), 20

`Attribute` (class in `terraformtestinglib.testing.testing`), 13

`attribute()` (`terraformtestinglib.testing.testing.AttributeList` method), 13

`attribute()` (`terraformtestinglib.testing.testing.Container` method), 15

`attribute_matching_regex()` (`terraformtestinglib.testing.testing.Container` method), 15

`AttributeList` (class in `terraformtestinglib.testing.testing`), 13

C

`ConfigurationError` (class in `terraformtestinglib.utils.errortypes`), 20

`Container` (class in `terraformtestinglib.testing.testing`), 14

D

`Data` (class in `terraformtestinglib.testing.testing`), 17

`data` (`terraformtestinglib.terraformtestinglib.HclFileResource` attribute), 22

`data()` (`terraformtestinglib.testing.testing.Validator` method), 18

`DataList` (class in `terraformtestinglib.testing.testing`), 18

E

`Entity` (class in `terraformtestinglib.testing.testing`), 18

`entity` (`terraformtestinglib.utils.errortypes.ConfigurationError` attribute), 20

`entity` (`terraformtestinglib.utils.errortypes.RuleError` attribute), 21

`errors` (`terraformtestinglib.linting.linting.Rule` attribute), 12

`errors` (`terraformtestinglib.linting.linting.Stack` attribute), 12

F

`field` (`terraformtestinglib.utils.errortypes.ConfigurationError` attribute), 20

`field` (`terraformtestinglib.utils.errortypes.RuleError` attribute), 21

`filename` (`terraformtestinglib.terraformtestinglib.HclFileResource` attribute), 22

`FilenameError` (class in `terraformtestinglib.utils.errortypes`), 21

G

`get_counter_resource_data_by_type()` (`terraformtestinglib.terraformtestinglib.HclView` method), 22

`get_resource_data_by_type()` (`terraformtestinglib.terraformtestinglib.HclView` method), 22

`get_rule_for_resource()` (`terraformtestinglib.linting.linting.RuleSet` method), 12

`get_variable_value()` (`terraformtestinglib.terraformtestinglib.HclView` method), 23

`get_variable_value()` (`terraformtestinglib.testing.testing.Validator` method), 19

H

`HclFileResource` (class in `terraformtestinglib.terraformtestinglib`), 22

`HclView` (class in `terraformtestinglib.terraformtestinglib`), 22

I

`if_has_attribute()` (`terraformtestinglib.testing.testing.Container` method),

15
[if_has_attribute_with_regex_value\(\)](#) (terraformtestinglib.testing.testing.Container method), 15
[if_has_attribute_with_value\(\)](#) (terraformtestinglib.testing.testing.AttributeList method), 13
[if_has_attribute_with_value\(\)](#) (terraformtestinglib.testing.testing.Container method), 15
[if_has_subattribute\(\)](#) (terraformtestinglib.testing.testing.Container method), 15
[if_has_subattribute_with_regex_value\(\)](#) (terraformtestinglib.testing.testing.Container method), 16
[if_has_subattribute_with_value\(\)](#) (terraformtestinglib.testing.testing.Container method), 16
[if_not_has_attribute\(\)](#) (terraformtestinglib.testing.testing.Container method), 16
[if_not_has_attribute_with_regex_value\(\)](#) (terraformtestinglib.testing.testing.Container method), 16
[if_not_has_attribute_with_value\(\)](#) (terraformtestinglib.testing.testing.AttributeList method), 13
[if_not_has_attribute_with_value\(\)](#) (terraformtestinglib.testing.testing.Container method), 16
[if_not_has_subattribute\(\)](#) (terraformtestinglib.testing.testing.Container method), 17
[if_not_has_subattribute_with_regex_value\(\)](#) (terraformtestinglib.testing.testing.Container method), 17
[if_not_has_subattribute_with_value\(\)](#) (terraformtestinglib.testing.testing.Container method), 17
[InvalidNaming](#), 23
[InvalidPositioning](#), 23
[is_valid_regex\(\)](#) (in module terraformtestinglib.configuration), 22
[iter_rec_update\(\)](#) (terraformtestinglib.utils.utils.RecursiveDictionary method), 21

L

[LintingResource](#) (class in terraformtestinglib.linting.linting), 11

M

[MissingVariable](#), 23

O

[original_value](#) (terraformtestinglib.utils.erroratypes.ConfigurationError attribute), 20

[original_value](#) (terraformtestinglib.utils.erroratypes.RuleError attribute), 21

P

[Parser](#) (class in terraformtestinglib.terraformtestinglib), 23

[Provider](#) (class in terraformtestinglib.testing.testing), 18

[provider\(\)](#) (terraformtestinglib.testing.testing.Validator method), 19

[ProviderList](#) (class in terraformtestinglib.testing.testing), 18

R

[RecursiveDictionary](#) (class in terraformtestinglib.utils.utils), 21

[regex](#) (terraformtestinglib.utils.erroratypes.ConfigurationError attribute), 20

[regex](#) (terraformtestinglib.utils.erroratypes.RuleError attribute), 21

[register_positioning_set\(\)](#) (terraformtestinglib.linting.linting.LintingResource method), 11

[register_rules_set\(\)](#) (terraformtestinglib.linting.linting.LintingResource method), 11

[Resource](#) (class in terraformtestinglib.testing.testing), 18

[resource_name](#) (terraformtestinglib.terraformtestinglib.HclFileResource attribute), 22

[resource_name](#) (terraformtestinglib.testing.testing.Attribute attribute), 13

[resource_type](#) (terraformtestinglib.terraformtestinglib.HclFileResource attribute), 22

[resource_type](#) (terraformtestinglib.testing.testing.Attribute attribute), 13

[resource_type](#) (terraformtestinglib.utils.erroratypes.ConfigurationError attribute), 20

[resource_type](#) (terraformtestinglib.utils.erroratypes.RuleError attribute), 21

[ResourceError](#) (class in terraformtestinglib.utils.erroratypes), 21

ResourceList (class in terraformtestinglib.testing.testing), 18

resources() (terraformtestinglib.testing.testing.ResourceList method), 18

resources() (terraformtestinglib.testing.testing.Validator method), 19

Rule (class in terraformtestinglib.linting.linting), 12

RuleError (class in terraformtestinglib.utils.errortypes), 21

RuleSet (class in terraformtestinglib.linting.linting), 12

S

should_be_valid_json() (terraformtestinglib.testing.testing.AttributeList method), 14

should_equal() (terraformtestinglib.testing.testing.AttributeList method), 14

should_have_attributes() (terraformtestinglib.testing.testing.AttributeList method), 14

should_have_attributes() (terraformtestinglib.testing.testing.Container method), 17

should_match_regex() (terraformtestinglib.testing.testing.AttributeList method), 14

should_not_equal() (terraformtestinglib.testing.testing.AttributeList method), 14

should_not_have_attributes() (terraformtestinglib.testing.testing.AttributeList method), 14

should_not_have_attributes() (terraformtestinglib.testing.testing.Container method), 17

should_not_match_regex() (terraformtestinglib.testing.testing.AttributeList method), 14

Stack (class in terraformtestinglib.linting.linting), 12

T

Terraform (class in terraformtestinglib.testing.testing), 18

terraform() (terraformtestinglib.testing.testing.Validator method), 19

TerraformList (class in terraformtestinglib.testing.testing), 18

terraformtestinglib (module), 23

terraformtestinglib.configuration (module), 22

terraformtestinglib.linting (module), 13

terraformtestinglib.linting.linting (module), 11

terraformtestinglib.terraformtestinglib (module), 22

terraformtestinglib.terraformtestinglibexceptions (module), 23

terraformtestinglib.testing (module), 20

terraformtestinglib.testing.testing (module), 13

terraformtestinglib.utils (module), 22

terraformtestinglib.utils.errortypes (module), 20

terraformtestinglib.utils.utils (module), 21

to_list() (terraformtestinglib.testing.testing.Validator static method), 19

U

update() (terraformtestinglib.utils.utils.RecursiveDictionary method), 21

V

validate() (terraformtestinglib.linting.linting.LintingResource method), 11

validate() (terraformtestinglib.linting.linting.Rule method), 12

validate() (terraformtestinglib.linting.linting.Stack method), 12

Validator (class in terraformtestinglib.testing.testing), 18

value (terraformtestinglib.utils.errortypes.ConfigurationError attribute), 21

value (terraformtestinglib.utils.errortypes.RuleError attribute), 21

value_equals() (terraformtestinglib.testing.testing.Variable method), 20

value_exists() (terraformtestinglib.testing.testing.Variable method), 20

value_matches_regex() (terraformtestinglib.testing.testing.Variable method), 20

Variable (class in terraformtestinglib.testing.testing), 19

variable() (terraformtestinglib.testing.testing.Validator method), 19

W

warning_on_one_line() (in module terraformtestinglib.terraformtestinglib), 23