controltowerlib Documentation

Release 0.5.0

Costas Tyfoxylos

Apr 01, 2021

CONTENTS

1 controltowerlib **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 controltowerlib **11**
- 5.1 controltowerlib package . . . 11

6 Credits **23**
- 6.1 Development Lead . . . 23
- 6.2 Contributors . . . 23

7 History **25**

8 0.0.1 (18-02-2020) **27**

9 0.1.0 (12-01-2021) **29**

10 0.1.1 (19-01-2021) **31**

11 0.1.2 (19-01-2021) **33**

12 0.2.0 (28-01-2021) **35**

13 0.3.0 (26-02-2021) **37**

14 0.4.0 (11-03-2021) **39**

15 0.5.0 (01-04-2021) **41**

16 Indices and tables **43**

Python Module Index **45**

Index **47**

Contents:

CHAPTER

ONE

CONTROLTOWERLIB

A library to provide access to AWS service catalog provided by Control Tower and basic functionality with interfacing with it.

- Documentation: https://controltowerlib.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

- Lists accounts and OUs under Control Tower
- Lists OUs under Organization
- Create and delete and OU under Control Tower and register in Control Tower an OU created under Organizations
- **Create and terminate accounts under Control Tower**

  (Accounts terminated are deleted from control tower and service catalog and moved under Organizations in an OU c
  Suspended. The OU needs to exist and it should have a completely restrictive SCP applied to it)
- Some other stuff…

CHAPTER

TWO

INSTALLATION

At the command line:

```
$ pip install controltowerlib
```

Or, if you have virtualenvwrapper installed:

```
$ mkvirtualenv controltowerlib
$ pip install controltowerlib
```

Or, if you are using pipenv:

```
$ pipenv install controltowerlib
```

Or, if you are using pipx:

```
$ pipx install controltowerlib
```

CHAPTER

THREE

USAGE

To develop on controltowerlib:

```
# 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 controltowerlib in a project:

```
from controltowerlib import ControlTower
tower = ControlTower('arn:aws:iam::ACCOUNTID:role/ValidAdministrativeRole')

for account in tower.accounts:
    print(account.name)
>>> root
    Audit
    Log archive

for account in tower.accounts:
    print(account.guardrail_compliance_status)
>>> COMPLIANT
    COMPLIANT
    COMPLIANT
```

(continues on next page)

(continued from previous page)

```
for ou in tower.organizational_units:
    print(ou.name)
>>> Custom
    Core
    Root

tower.create_organizational_unit('TestOU')
>>> True

tower.delete_organizational_unit('TestOU')
>>> True
```

CHAPTER

FOUR

CONTRIBUTING

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 *controltowerlib* for local development. Using of pipenv is highly recommended.

1. Clone your fork locally:

```
$ git clone git@github.com:schubergphilis/controltowerlib.git
```

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

```
$ cd controltowerlib/
$ pipenv install --ignore-pipfile
```

3. Create a branch for local development:

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

   Now you can make your changes locally. Do your development while using the CI capabilities and making sure the code passes lint, test, build and document stages.

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

CHAPTER

FIVE

CONTROLTOWERLIB

5.1 controltowerlib package

5.1.1 Subpackages

controltowerlib.resources package

Submodules

controltowerlib.resources.configuration module

configuration module.

Import all parts from configuration here

controltowerlib.resources.resources module

resources module.

Import all parts from resources here

class `controltowerlib.resources.resources.`**`AccountFactory`**(*service_catalog_client*, *data*)

Bases: `object`

Models the account factory data of service catalog.

class `controltowerlib.resources.resources.`**`ControlTowerAccount`**(*control_tower*, *data*, *info_polling_interval=30*)

Bases: `awsauthenticationlib.awsauthenticationlib.LoggerMixin`

Models the account data.

property arn

Arn.

attach_service_control_policy(*name*)

Attaches a Service Control Policy to the account.

Parameters **name** (*str*) – The name of the SCP to attach

Returns True on success, False otherwise.

Return type result (bool)

property created_time
Created Time.

delete (*suspended_ou_name=None*)
Delete.

detach_service_control_policy (*name*)
Detaches a Service Control Policy from the account.

Parameters **name** (*str*) – The name of the SCP to detach

Returns True on success, False otherwise.

Return type result (bool)

property email
Email.

property guardrail_compliance_status
Retrieves the guardrail compliancy status for the account.

Returns COMPLIANT|NON COMPLIANT

Return type status (str)

property has_available_update
If the account is behind the landing zone version.

property id
Id.

property idempotency_token
Idempotency Token.

property landing_zone_version
Landing zone version.

property last_record_id
Last Record ID.

property name
Name.

property organizational_unit
Organizational Unit.

property owner
Owner.

property physical_id
Physical ID.

property provision_state
Provision state.

property provisioning_artifact_id
Provisioning artifact ID.

property service_catalog_id
Service Catalog ID.

property service_catalog_product_id
Service catalog product ID.

property service_catalog_status
Service catalog status.

property service_catalog_tags
Service catalog tags.

property service_catalog_type
Service catalog type.

property service_catalog_user_arn
Service catalog user arn.

property sso_user_email
SSO user email.

property sso_user_portal
SSO user portal.

property stack_arn
Stack Arn.

property status
Status.

update()
Updates the account in service catalog.

Returns True if the call succeeded False otherwise

property user_arn_session
User arn session.

class controltowerlib.resources.resources.**ControlTowerOU**(*control_tower*, *data*)
Bases: object

Model the data of a Control Tower managed OU.

property create_date
The date the ou was created in timestamp.

delete()
Deletes the ou.

Returns True on success, False otherwise.

Return type response (bool)

property id
OU ID.

property name
The name of the OU.

property parent_ou_id
The id of the parent OU.

property parent_ou_name
The name of the parent OU.

property type
The type of the OU.

class controltowerlib.resources.resources.**CoreAccount**(*control_tower*, *account_label*, *data*)
Bases: object

Models the core landing zone account data.

property core_resource_mappings
Core resource mappings.

property email
Email.

property id
Id.

property label
Account label.

property stack_set_arn
Stack set arn.

class controltowerlib.resources.resources.**GuardRail**(*control_tower*, *data*)
Bases: awsauthenticationlib.awsauthenticationlib.LoggerMixin

Models the guard rail data.

property behavior
Behavior.

property category
Category.

property compliancy_status
Compliancy status.

property description
Description.

property display_name
DisplayName.

property name
Name.

property provider
Provider.

property regional_preference
Regional preference.

property type
Type.

class controltowerlib.resources.resources.**OrganizationsOU**(*data*)
Bases: object

Model the data of an Organizations managed OU.

property arn
The arn of the OU.

property id
The id of the OU.

property name
The name of the OU.

class controltowerlib.resources.resources.**ServiceControlPolicy**(*data*)
Bases: object

Models the account factory data of service catalog.

property arn
Arn.

property aws_managed
Aws Managed.

property description
Description.

property id
Id.

property name
Name.

property type
Type.

Module contents

configuration module.

Import all parts from configuration here

5.1.2 Submodules

5.1.3 controltowerlib.controltowerlib module

Main code for controltowerlib.

class `controltowerlib.controltowerlib.`**`ControlTower`**(*arn*, *settling_time=90*)
Bases: `awsauthenticationlib.awsauthenticationlib.LoggerMixin`

Models Control Tower by wrapping around service catalog.

property accounts
The accounts under control tower.

Returns A list of account objects under control tower's control.

Return type accounts (Account)

api_content_type = 'application/x-amz-json-1.1'

api_user_agent = 'aws-sdk-js/2.528.0 promise'

property baseline_update_available
Baseline update available.

property busy
Busy.

property catastrophic_drift
List of catastrophic drift.

core_account_types = ['PRIMARY', 'LOGGING', 'SECURITY']

property core_accounts
The core accounts of the landing zone.

Returns A list of the primary, logging and security account.

Return type core_accounts (list)

create_account(*account_name*, *account_email*, *organizational_unit*, *product_name=None*, *sso_first_name=None*, *sso_last_name=None*, *sso_user_email=None*)

Creates a Control Tower managed account.

Parameters

- **account_name** (*str*) – The name of the account.
- **account_email** (*str*) – The email of the account.
- **organizational_unit** (*str*) – The organizational unit that the account should be under.
- **product_name** (*str*) – The product name, if nothing is provided it uses the account name.
- **sso_first_name** (*str*) – The first name of the SSO user, defaults to "Control"
- **sso_last_name** (*str*) – The last name of the SSO user, defaults to "Tower"
- **sso_user_email** (*str*) – The email of the sso, if nothing is provided it uses the account email.

Returns True on success, False otherwise.

Return type result (bool)

create_organizational_unit(*name*)

Creates a Control Tower managed organizational unit.

Parameters **name** (*str*) – The name of the OU to create.

Returns True if successfull, False otherwise.

Return type result (bool)

decommission()

Decommissions a landing zone.

The api call does not seem to be enough and although the resources are decomissioned like with the proper process, control tower responds with a delete failed on the api, so it seems that aws needs to perform actions on their end for the decommissioning to be successful.

Returns True if the process starts successfully, False otherwise.

Return type response (bool)

delete_organizational_unit(*name*)

Deletes a Control Tower managed organizational unit.

Parameters **name** (*str*) – The name of the OU to delete.

Returns True if successfull, False otherwise.

Return type result (bool)

deploy(*logging_account_email*, *security_account_email*, *regions=None*, *retries=10*, *wait=1*)

Deploys control tower.

Returns True on success, False on failure.

Return type bool

property deploying_messages
Deploying messages.

property drift_messages
Drift messages.

property enabled_guard_rails
Enabled guard rails.

get_account_by_arn(*arn*)
Retrieves an account by arn.

Returns An account object that matches the arn or None.

Return type account (Account)

get_account_by_id(*id_*)
Retrieves an account by id.

Returns An account object that matches the id or None.

Return type account (Account)

get_account_by_name(*name*)
Retrieves an account by name.

Returns An account object that matches the name or None.

Return type account (Account)

get_accounts_with_available_updates()
Retrieves the accounts that have available updates from control tower.

Returns A list of account objects under control tower's control with available updates.

Return type accounts (Account)

get_available_accounts()
Retrieves the available accounts from control tower.

Returns A list of available account objects under control tower's control.

Return type accounts (Account)

static get_available_regions()
The regions that control tower can be active in.

Returns A list of strings of the regions that control tower can be active in.

Return type regions (list)

get_changing_accounts()
Retrieves the under change accounts from control tower.

Returns A list of under change account objects under control tower's control.

Return type accounts (Account)

get_erroring_accounts()
Retrieves the erroring accounts from control tower.

Returns A list of erroring account objects under control tower's control.

Return type accounts (Account)

get_organizational_unit_by_id(*id_*)
Gets a Control Tower managed Organizational Unit by id.

Parameters **id** (*str*) – The id of the organizational unit to retrieve.

Returns A OU object on success, None otherwise.

Return type result (*ControlTowerOU*)

get_organizational_unit_by_name(*name*)
Gets a Control Tower managed Organizational Unit by name.

Parameters **name** (*str*) – The name of the organizational unit to retrieve.

Returns A OU object on success, None otherwise.

Return type result (*ControlTowerOU*)

get_organizations_ou_by_arn(*arn*)
Gets an Organizations managed Organizational Unit by arn.

Parameters **arn** (*str*) – The arn of the organizational unit to retrieve.

Returns A OU object on success, None otherwise.

Return type result (*OrganizationsOU*)

get_organizations_ou_by_id(*id_*)
Gets an Organizations managed Organizational Unit by id.

Parameters **id** (*str*) – The id of the organizational unit to retrieve.

Returns A OU object on success, None otherwise.

Return type result (*OrganizationsOU*)

get_organizations_ou_by_name(*name*)
Gets an Organizations managed Organizational Unit by name.

Parameters **name** (*str*) – The name of the organizational unit to retrieve.

Returns A OU object on success, None otherwise.

Return type result (*OrganizationsOU*)

get_service_control_policy_by_name(*name*)
Retrieves a service control policy by name.

Parameters **name** (*str*) – The name of the SCP to retrieve

Returns The scp if a match is found else None.

Return type scp (*ServiceControlPolicy*)

get_updated_accounts()
Retrieves the accounts that have no available updates from control tower.

Returns A list of account objects under control tower's control with no available updates.

Return type accounts (Account)

property guard_rails
Guard rails.

property guard_rails_violations
List guard rails violations.

property guardrail_update_available
Guardrail update available.

property is_deployed
The deployment status of control tower.

is_email_used(*email*)
Check email for availability to be used or if it is already in use.

property landing_zone_update_available
Landing Zone update available.

property landing_zone_version
Landing zone version.

property organizational_units
The organizational units under control tower.

Returns A list of organizational units objects under control tower's control.

Return type organizational_units (OrganizationalUnit)

property organizations_ous
The organizational units under Organizations.

Returns A list of organizational units objects under Organizations.

Return type organizational_units (*OrganizationsOU*)

property percentage_complete
Percentage complete.

property region
Region.

property region_metadata_list
Region metadata list.

register_organizations_ou(*name*)
Registers an Organizations OU under control tower.

Parameters **name** (*str*) – The name of the Organizations OU to register to Control Tower.

Returns True if successfull, False otherwise.

Return type result (bool)

property root_ou
The root ou of control tower.

Returns The root ou object.

Return type root_ou (*ControlTowerOU*)

property service_control_policies
The service control policies under organization.

Returns A list of SCPs under the organization.

Return type service_control_policies (list)

property service_landing_zone_version
Service landing zone version.

property status
Status.

supported_targets = ['listManagedOrganizationalUnits', 'manageOrganizationalUnit', 'de

update()
Updates the control tower to the latest version.

Returns True on success, False on failure.

Return type bool

property user_landing_zone_version
User landing zone version.

validate_availability()
Validation decorator.

5.1.4 controltowerlib.controltowerlibexceptions module

Custom exception code for controltowerlib.

exception controltowerlib.controltowerlibexceptions.**ControlTowerBusy**
Bases: Exception

The control tower is already executing some action.

exception controltowerlib.controltowerlibexceptions.**ControlTowerNotDeployed**
Bases: Exception

The control tower is deployed at all.

exception controltowerlib.controltowerlibexceptions.**EmailCheckFailed**
Bases: Exception

Checking of the email was not possible.

exception controltowerlib.controltowerlibexceptions.**EmailInUse**
Bases: Exception

The email privided is already in use and cannot be used to deploy an account.

exception controltowerlib.controltowerlibexceptions.**NoServiceCatalogAccess**
Bases: Exception

There is no access to service catalog.

exception controltowerlib.controltowerlibexceptions.**NoSuspendedOU**
Bases: Exception

The suspended ou has not been created.

exception controltowerlib.controltowerlibexceptions.**NonExistentSCP**
Bases: Exception

The SCP requested does not exist.

exception controltowerlib.controltowerlibexceptions.**OUCreating**
Bases: Exception

The organizational unit is still under creation and cannot be used.

exception controltowerlib.controltowerlibexceptions.**PreDeployValidationFailed**
Bases: Exception

The pre deployment validation failed.

exception controltowerlib.controltowerlibexceptions.**RoleCreationFailure**
Bases: Exception

Unable to create the required roles for the deployment of control tower, manual clean up is required.

exception controltowerlib.controltowerlibexceptions.**ServiceCallFailed**
Bases: Exception

The call to the service has failed.

exception controltowerlib.controltowerlibexceptions.**UnavailableRegion**
Bases: Exception

The region or regions provided to control tower to deploy in are not available.

exception controltowerlib.controltowerlibexceptions.**UnsupportedTarget**
Bases: Exception

The target call is not supported by the current implementation.

5.1.5 Module contents

controltowerlib module.

Import all parts from controltowerlib here

CHAPTER

SIX

CREDITS

6.1 Development Lead

- Costas Tyfoxylos <ctyfoxylos@schubergphilis.com>

6.2 Contributors

None yet. Why not be the first?

CHAPTER

SEVEN

HISTORY

CHAPTER EIGHT

0.0.1 (18-02-2020)

- First code creation

CHAPTER NINE

0.1.0 (12-01-2021)

- First public release.

CHAPTER TEN

0.1.1 (19-01-2021)

- Working around the OU pagination until the implementation fixes that properly.

CHAPTER

ELEVEN

0.1.2 (19-01-2021)

- Fixed pagination bug properly.

CHAPTER TWELVE

0.2.0 (28-01-2021)

- Implemented control tower versions and account update capabilites.

CHAPTER

THIRTEEN

0.3.0 (26-02-2021)

- Implemented control tower update, identifying the deployment status and locking when busy, along with exposing some more data like guard rails and drift messages.

CHAPTER

FOURTEEN

0.4.0 (11-03-2021)

- Made account deletion not depending on the suspended OU existance and account factory retrieval checking for CT being deployed.

CHAPTER

FIFTEEN

0.5.0 (01-04-2021)

- Implemented (not properly functioning still) decommissioning of control tower.

CHAPTER

SIXTEEN

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

C

controltowerlib, 21
controltowerlib.controltowerlib, 15
controltowerlib.controltowerlibexceptions, 20
controltowerlib.resources, 15
controltowerlib.resources.configuration, 11
controltowerlib.resources.resources, 11

INDEX

A

AccountFactory (*class in controltowerlib.resources.resources*), 11
accounts() (*controltowerlib.controltowerlib.ControlTower property*), 15
api_content_type (*controltowerlib.controltowerlib.ControlTower attribute*), 15
api_user_agent (*controltowerlib.controltowerlib.ControlTower attribute*), 15
arn() (*controltowerlib.resources.resources.ControlTowerAccount property*), 11
arn() (*controltowerlib.resources.resources.OrganizationsOU property*), 14
arn() (*controltowerlib.resources.resources.ServiceControlPolicy property*), 15
attach_service_control_policy() (*controltowerlib.resources.resources.ControlTowerAccount method*), 11
aws_managed() (*controltowerlib.resources.resources.ServiceControlPolicy property*), 15

B

baseline_update_available() (*controltowerlib.controltowerlib.ControlTower property*), 15
behavior() (*controltowerlib.resources.resources.GuardRail property*), 14
busy() (*controltowerlib.controltowerlib.ControlTower property*), 15

C

catastrophic_drift() (*controltowerlib.controltowerlib.ControlTower property*), 15
category() (*controltowerlib.resources.resources.GuardRail property*), 14
compliancy_status() (*controltowerlib.resources.resources.GuardRail property*), 14
ControlTower (*class in controltowerlib.controltowerlib*), 15
ControlTowerAccount (*class in controltowerlib.resources.resources*), 11
ControlTowerBusy, 20
controltowerlib
    module, 21
controltowerlib.controltowerlib
    module, 15
controltowerlib.controltowerlibexceptions
    module, 20
controltowerlib.resources
    module, 15
controltowerlib.resources.configuration
    module, 11
controltowerlib.resources.resources
    module, 11
ControlTowerNotDeployed, 20
ControlTowerOU (*class in controltowerlib.resources.resources*), 13
core_account_types (*controltowerlib.controltowerlib.ControlTower attribute*), 15
core_accounts() (*controltowerlib.controltowerlib.ControlTower property*), 15
core_resource_mappings() (*controltowerlib.resources.resources.CoreAccount property*), 14
CoreAccount (*class in controltowerlib.resources.resources*), 13
create_account() (*controltowerlib.controltowerlib.ControlTower method*), 16
create_date() (*controltowerlib.resources.resources.ControlTowerOU property*), 13
create_organizational_unit() (*controltowerlib.controltowerlib.ControlTower method*), 16

created_time() (*controltowerlib.resources.resources.ControlTowerAccount property*), 11

D

decommission() (*controltowerlib.controltowerlib.ControlTower method*), 16
delete() (*controltowerlib.resources.resources.ControlTowerAccount method*), 12
delete() (*controltowerlib.resources.resources.ControlTowerOU method*), 13
delete_organizational_unit() (*controltowerlib.controltowerlib.ControlTower method*), 16
deploy() (*controltowerlib.controltowerlib.ControlTower method*), 16
deploying_messages() (*controltowerlib.controltowerlib.ControlTower property*), 16
description() (*controltowerlib.resources.resources.GuardRail property*), 14
description() (*controltowerlib.resources.resources.ServiceControlPolicy property*), 15
detach_service_control_policy() (*controltowerlib.resources.resources.ControlTowerAccount method*), 12
display_name() (*controltowerlib.resources.resources.GuardRail property*), 14
drift_messages() (*controltowerlib.controltowerlib.ControlTower property*), 17

E

email() (*controltowerlib.resources.resources.ControlTowerAccount property*), 12
email() (*controltowerlib.resources.resources.CoreAccount property*), 14
EmailCheckFailed, 20
EmailInUse, 20
enabled_guard_rails() (*controltowerlib.controltowerlib.ControlTower property*), 17

G

get_account_by_arn() (*controltowerlib.controltowerlib.ControlTower method*), 17
get_account_by_id() (*controltowerlib.controltowerlib.ControlTower method*), 17
get_account_by_name() (*controltowerlib.controltowerlib.ControlTower method*), 17
get_accounts_with_available_updates() (*controltowerlib.controltowerlib.ControlTower method*), 17
get_available_accounts() (*controltowerlib.controltowerlib.ControlTower method*), 17
get_available_regions() (*controltowerlib.controltowerlib.ControlTower static method*), 17
get_changing_accounts() (*controltowerlib.controltowerlib.ControlTower method*), 17
get_erroring_accounts() (*controltowerlib.controltowerlib.ControlTower method*), 17
get_organizational_unit_by_id() (*controltowerlib.controltowerlib.ControlTower method*), 17
get_organizational_unit_by_name() (*controltowerlib.controltowerlib.ControlTower method*), 18
get_organizations_ou_by_arn() (*controltowerlib.controltowerlib.ControlTower method*), 18
get_organizations_ou_by_id() (*controltowerlib.controltowerlib.ControlTower method*), 18
get_organizations_ou_by_name() (*controltowerlib.controltowerlib.ControlTower method*), 18
get_service_control_policy_by_name() (*controltowerlib.controltowerlib.ControlTower method*), 18
get_updated_accounts() (*controltowerlib.controltowerlib.ControlTower method*), 18
guard_rails() (*controltowerlib.controltowerlib.ControlTower property*), 18
guard_rails_violations() (*controltowerlib.controltowerlib.ControlTower property*), 18
GuardRail (*class in controltowerlib.resources.resources*), 14
guardrail_compliance_status() (*controltowerlib.resources.resources.ControlTowerAccount property*), 12
guardrail_update_available() (*controltowerlib.controltowerlib.ControlTower property*), 18

H

has_available_update() (*controltowerlib.resources.resources.ControlTowerAccount property*), 12

I

id() (*controltowerlib.resources.resources.ControlTowerAccount property*), 12
id() (*controltowerlib.resources.resources.ControlTowerOU property*), 13
id() (*controltowerlib.resources.resources.CoreAccount property*), 14
id() (*controltowerlib.resources.resources.OrganizationsOU property*), 14
id() (*controltowerlib.resources.resources.ServiceControlPolicy property*), 15
idempotency_token() (*controltowerlib.resources.resources.ControlTowerAccount property*), 12
is_deployed() (*controltowerlib.controltowerlib.ControlTower property*), 18
is_email_used() (*controltowerlib.controltowerlib.ControlTower method*), 19

L

label() (*controltowerlib.resources.resources.CoreAccount property*), 14
landing_zone_update_available() (*controltowerlib.controltowerlib.ControlTower property*), 19
landing_zone_version() (*controltowerlib.controltowerlib.ControlTower property*), 19
landing_zone_version() (*controltowerlib.resources.resources.ControlTowerAccount property*), 12
last_record_id() (*controltowerlib.resources.resources.ControlTowerAccount property*), 12

M

module
- controltowerlib, 21
- controltowerlib.controltowerlib, 15
- controltowerlib.controltowerlibexceptions, 20
- controltowerlib.resources, 15
- controltowerlib.resources.configuration, 11
- controltowerlib.resources.resources, 11

N

name() (*controltowerlib.resources.resources.ControlTowerAccount property*), 12
name() (*controltowerlib.resources.resources.ControlTowerOU property*), 13
name() (*controltowerlib.resources.resources.GuardRail property*), 14
name() (*controltowerlib.resources.resources.OrganizationsOU property*), 14
name() (*controltowerlib.resources.resources.ServiceControlPolicy property*), 15
NonExistentSCP, 20
NoServiceCatalogAccess, 20
NoSuspendedOU, 20

O

organizational_unit() (*controltowerlib.resources.resources.ControlTowerAccount property*), 12
organizational_units() (*controltowerlib.controltowerlib.ControlTower property*), 19
organizations_ous() (*controltowerlib.controltowerlib.ControlTower property*), 19
OrganizationsOU (*class in controltowerlib.resources.resources*), 14
OUCreating, 20
owner() (*controltowerlib.resources.resources.ControlTowerAccount property*), 12

P

parent_ou_id() (*controltowerlib.resources.resources.ControlTowerOU property*), 13
parent_ou_name() (*controltowerlib.resources.resources.ControlTowerOU property*), 13
percentage_complete() (*controltowerlib.controltowerlib.ControlTower property*), 19
physical_id() (*controltowerlib.resources.resources.ControlTowerAccount property*), 12
PreDeployValidationFailed, 20
provider() (*controltowerlib.resources.resources.GuardRail property*), 14

provision_state() (*controltowerlib.resources.resources.ControlTowerAccount property*), 12
provisioning_artifact_id() (*controltowerlib.resources.resources.ControlTowerAccount property*), 12

R

region() (*controltowerlib.controltowerlib.ControlTower property*), 19
region_metadata_list() (*controltowerlib.controltowerlib.ControlTower property*), 19
regional_preference() (*controltowerlib.resources.resources.GuardRail property*), 14
register_organizations_ou() (*controltowerlib.controltowerlib.ControlTower method*), 19
RoleCreationFailure, 20
root_ou() (*controltowerlib.controltowerlib.ControlTower property*), 19

S

service_catalog_id() (*controltowerlib.resources.resources.ControlTowerAccount property*), 12
service_catalog_product_id() (*controltowerlib.resources.resources.ControlTowerAccount property*), 12
service_catalog_status() (*controltowerlib.resources.resources.ControlTowerAccount property*), 12
service_catalog_tags() (*controltowerlib.resources.resources.ControlTowerAccount property*), 13
service_catalog_type() (*controltowerlib.resources.resources.ControlTowerAccount property*), 13
service_catalog_user_arn() (*controltowerlib.resources.resources.ControlTowerAccount property*), 13
service_control_policies() (*controltowerlib.controltowerlib.ControlTower property*), 19
service_landing_zone_version() (*controltowerlib.controltowerlib.ControlTower property*), 19
ServiceCallFailed, 21
ServiceControlPolicy (*class in controltowerlib.resources.resources*), 14
sso_user_email() (*controltowerlib.resources.resources.ControlTowerAccount property*), 13
sso_user_portal() (*controltowerlib.resources.resources.ControlTowerAccount property*), 13
stack_arn() (*controltowerlib.resources.resources.ControlTowerAccount property*), 13
stack_set_arn() (*controltowerlib.resources.resources.CoreAccount property*), 14
status() (*controltowerlib.controltowerlib.ControlTower property*), 19
status() (*controltowerlib.resources.resources.ControlTowerAccount property*), 13
supported_targets (*controltowerlib.controltowerlib.ControlTower attribute*), 19

T

type() (*controltowerlib.resources.resources.ControlTowerOU property*), 13
type() (*controltowerlib.resources.resources.GuardRail property*), 14
type() (*controltowerlib.resources.resources.ServiceControlPolicy property*), 15

U

UnavailableRegion, 21
UnsupportedTarget, 21
update() (*controltowerlib.controltowerlib.ControlTower method*), 19
update() (*controltowerlib.resources.resources.ControlTowerAccount method*), 13
user_arn_session() (*controltowerlib.resources.resources.ControlTowerAccount property*), 13
user_landing_zone_version() (*controltowerlib.controltowerlib.ControlTower property*), 20

V

validate_availability() (*controltowerlib.controltowerlib.ControlTower method*), 20