

cion-api

1.0.0

Generated by Doxygen 1.8.14



# Contents

<b>1</b>	<b>Namespace Index</b>	<b>1</b>
1.1	Namespace List . . . . .	1
<b>2</b>	<b>Class Index</b>	<b>3</b>
2.1	Class List . . . . .	3
<b>3</b>	<b>File Index</b>	<b>5</b>
3.1	File List . . . . .	5
<b>4</b>	<b>Namespace Documentation</b>	<b>7</b>
4.1	app Namespace Reference . . . . .	7
4.1.1	Function Documentation . . . . .	7
4.1.1.1	index() . . . . .	7
4.1.2	Variable Documentation . . . . .	7
4.1.2.1	app . . . . .	8
4.1.2.2	host . . . . .	8
4.1.2.3	port . . . . .	8
4.1.2.4	prod . . . . .	8
4.1.2.5	static_path . . . . .	8
4.1.2.6	ws_route . . . . .	8
4.2	auth Namespace Reference . . . . .	8
4.2.1	Function Documentation . . . . .	9
4.2.1.1	api_auth() . . . . .	9
4.2.1.2	api_create_user() . . . . .	9
4.2.1.3	bad_creds_response() . . . . .	9

4.2.1.4	create_hash()	9
4.2.1.5	create_session()	10
4.2.1.6	db_create_user()	10
4.2.1.7	forbidden_response()	10
4.2.1.8	has_permission()	10
4.2.1.9	hash_str()	11
4.2.1.10	invalidate_sessions()	11
4.2.1.11	logout()	11
4.2.1.12	requires_auth()	11
4.2.1.13	retrieve_session()	12
4.2.1.14	validate_password()	12
4.2.1.15	verify_token()	12
4.2.1.16	watch_user()	12
4.2.2	Variable Documentation	12
4.2.2.1	sessions	12
4.3	cion_system Namespace Reference	13
4.3.1	Function Documentation	13
4.3.1.1	get_health()	13
4.4	documents Namespace Reference	13
4.4.1	Function Documentation	13
4.4.1.1	db_get_document()	13
4.4.1.2	editable_documents()	14
4.4.1.3	generate_permission_def()	14
4.4.1.4	get_document()	14
4.4.1.5	get_documents()	14
4.4.1.6	get_permission_def()	14
4.4.1.7	json()	15
4.4.1.8	lazy()	15
4.4.1.9	set_document()	15
4.4.1.10	sort_array_values()	15

4.5	environments Namespace Reference . . . . .	16
4.5.1	Function Documentation . . . . .	16
4.5.1.1	create_environment() . . . . .	16
4.5.1.2	db_create_environment() . . . . .	16
4.5.1.3	get_environments() . . . . .	16
4.5.1.4	get_webhook() . . . . .	16
4.6	permissions Namespace Reference . . . . .	16
4.7	permissions.permission Namespace Reference . . . . .	17
4.7.1	Function Documentation . . . . .	17
4.7.1.1	perm() . . . . .	17
4.8	rdb_conn Namespace Reference . . . . .	17
4.8.1	Function Documentation . . . . .	18
4.8.1.1	create_admin_user_insert() . . . . .	18
4.8.1.2	ensure_db_exists() . . . . .	18
4.8.1.3	ensure_table_exists() . . . . .	18
4.8.1.4	init() . . . . .	19
4.8.2	Variable Documentation . . . . .	19
4.8.2.1	Connection . . . . .	19
4.9	search Namespace Reference . . . . .	19
4.9.1	Function Documentation . . . . .	19
4.9.1.1	get_filter() . . . . .	19
4.9.1.2	match() . . . . .	20
4.9.1.3	parse_date_time() . . . . .	20
4.9.1.4	traverse() . . . . .	20
4.9.2	Variable Documentation . . . . .	20
4.9.2.1	filter_func . . . . .	20
4.9.2.2	filter_string . . . . .	20
4.9.2.3	rows . . . . .	20
4.10	services Namespace Reference . . . . .	21
4.10.1	Function Documentation . . . . .	21

4.10.1.1	create_service()	21
4.10.1.2	db_create_service()	21
4.10.1.3	db_delete_service()	22
4.10.1.4	db_get_running_image()	22
4.10.1.5	db_get_service_conf()	22
4.10.1.6	db_get_services()	22
4.10.1.7	db_get_unique_deployed_images()	23
4.10.1.8	db_replace_service()	23
4.10.1.9	delete_service()	23
4.10.1.10	edit_service()	23
4.10.1.11	get_running_image()	23
4.10.1.12	get_service()	24
4.10.1.13	get_services()	24
4.10.1.14	resolve_service_create()	24
4.10.1.15	resolve_service_delete()	24
4.10.1.16	task_base_image_name_filter()	25
4.10.1.17	url_safe_service_image()	25
4.10.1.18	validate_input()	25
4.11	table Namespace Reference	25
4.11.1	Function Documentation	25
4.11.1.1	table_query()	26
4.12	tasks Namespace Reference	26
4.12.1	Function Documentation	26
4.12.1.1	create_task()	26
4.12.1.2	db_create_scheduled_task()	26
4.12.1.3	db_create_task()	27
4.12.1.4	get_recent_tasks()	27
4.12.1.5	get_task()	27
4.12.1.6	get_tasks()	27
4.12.1.7	resolve_task_create()	27

4.12.1.8	<code>schedule_deploy()</code>	28
4.13	user Namespace Reference	28
4.13.1	Function Documentation	28
4.13.1.1	<code>bad_creds_response()</code>	28
4.13.1.2	<code>change_own_password()</code>	28
4.13.1.3	<code>change_password()</code>	29
4.13.1.4	<code>db_change_password()</code>	29
4.13.1.5	<code>db_delete_user()</code>	29
4.13.1.6	<code>db_get_permissions()</code>	29
4.13.1.7	<code>db_get_users()</code>	30
4.13.1.8	<code>db_set_gravatar_email()</code>	30
4.13.1.9	<code>db_set_permissions()</code>	30
4.13.1.10	<code>delete_user()</code>	30
4.13.1.11	<code>get_permissions()</code>	31
4.13.1.12	<code>get_users()</code>	31
4.13.1.13	<code>help_change_password()</code>	31
4.13.1.14	<code>set_gravatar_email()</code>	31
4.13.1.15	<code>set_permissions()</code>	31
4.14	webhooks Namespace Reference	32
4.14.1	Function Documentation	32
4.14.1.1	<code>create_webhook()</code>	32
4.14.1.2	<code>db_create_webhook()</code>	32
4.14.1.3	<code>delete_webhook()</code>	32
4.14.1.4	<code>get_webhook()</code>	32
4.14.1.5	<code>get_webhooks()</code>	33
4.15	websocket Namespace Reference	33
4.15.1	Function Documentation	33
4.15.1.1	<code>create()</code>	33

<b>5</b>	<b>Class Documentation</b>	<b>35</b>
5.1	permissions.permission.Permission Class Reference . . . . .	35
5.1.1	Detailed Description . . . . .	35
5.1.2	Constructor & Destructor Documentation . . . . .	35
5.1.2.1	__init__() . . . . .	35
5.1.3	Member Function Documentation . . . . .	35
5.1.3.1	__and__() . . . . .	36
5.1.3.2	__or__() . . . . .	36
5.1.3.3	has_permission() . . . . .	36
5.1.4	Member Data Documentation . . . . .	36
5.1.4.1	check_fn . . . . .	36
5.2	websocket.WebSocketListener Class Reference . . . . .	37
5.2.1	Constructor & Destructor Documentation . . . . .	37
5.2.1.1	__init__() . . . . .	37
5.2.2	Member Function Documentation . . . . .	37
5.2.2.1	handle_request() . . . . .	37
5.2.2.2	subscribe() . . . . .	38
5.2.2.3	unsubscribe() . . . . .	38
5.2.3	Member Data Documentation . . . . .	38
5.2.3.1	clients . . . . .	38
5.2.3.2	conn . . . . .	38
5.2.3.3	dispatch . . . . .	38
5.2.3.4	subscriptions . . . . .	38
<b>6</b>	<b>File Documentation</b>	<b>39</b>
6.1	src/app.py File Reference . . . . .	39
6.2	src/auth.py File Reference . . . . .	39
6.3	src/cion_system.py File Reference . . . . .	40
6.4	src/documents.py File Reference . . . . .	40
6.5	src/environments.py File Reference . . . . .	41
6.6	src/permissions/__init__.py File Reference . . . . .	41
6.7	src/permissions/permission.py File Reference . . . . .	41
6.8	src/rdb_conn.py File Reference . . . . .	41
6.9	src/search.py File Reference . . . . .	42
6.10	src/services.py File Reference . . . . .	42
6.11	src/table.py File Reference . . . . .	43
6.12	src/tasks.py File Reference . . . . .	43
6.13	src/user.py File Reference . . . . .	43
6.14	src/webhooks.py File Reference . . . . .	44
6.15	src/websocket.py File Reference . . . . .	44
	<b>Index</b>	<b>45</b>



# Chapter 1

## Namespace Index

### 1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

<b>app</b>	7
<b>auth</b>	8
<b>cion_system</b>	13
<b>documents</b>	13
<b>environments</b>	16
<b>permissions</b>	16
<b>permissions.permission</b>	17
<b>rdb_conn</b>	17
<b>search</b>	19
<b>services</b>	21
<b>table</b>	25
<b>tasks</b>	26
<b>user</b>	28
<b>webhooks</b>	32
<b>websocket</b>	33



## Chapter 2

# Class Index

### 2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<b>permissions.permission.Permission</b>	35
<b>websocket.WebSocketListener</b>	37



## Chapter 3

# File Index

### 3.1 File List

Here is a list of all files with brief descriptions:

src/ <b>app.py</b> . . . . .	39
src/ <b>auth.py</b> . . . . .	39
src/ <b>cion_system.py</b> . . . . .	40
src/ <b>documents.py</b> . . . . .	40
src/ <b>environments.py</b> . . . . .	41
src/ <b>rdb_conn.py</b> . . . . .	41
src/ <b>search.py</b> . . . . .	42
src/ <b>services.py</b> . . . . .	42
src/ <b>table.py</b> . . . . .	43
src/ <b>tasks.py</b> . . . . .	43
src/ <b>user.py</b> . . . . .	43
src/ <b>webhooks.py</b> . . . . .	44
src/ <b>websocket.py</b> . . . . .	44
src/permissions/ <b>__init__.py</b> . . . . .	41
src/permissions/ <b>permission.py</b> . . . . .	41



## Chapter 4

# Namespace Documentation

### 4.1 app Namespace Reference

#### Functions

- def **index** (request)

#### Variables

- **app** = web.Application()
- int **prod** = 'prod'
- **static\_path** = os.path.join(os.environ['WEB\_DIR'], 'lib')
- **ws\_route** = **websocket.create**(rdb\_conn.conn)
- **host**
- **port**

#### 4.1.1 Function Documentation

##### 4.1.1.1 index()

```
def app.index (  
    request )
```

#### 4.1.2 Variable Documentation

#### 4.1.2.1 app

```
app.app = web.Application()
```

#### 4.1.2.2 host

```
app.host
```

#### 4.1.2.3 port

```
app.port
```

#### 4.1.2.4 prod

```
int app.prod = 'prod'
```

#### 4.1.2.5 static\_path

```
app.static_path = os.path.join(os.environ['WEB_DIR'], 'lib')
```

#### 4.1.2.6 ws\_route

```
app.ws_route = websocket.create(rdb_conn.conn)
```

## 4.2 auth Namespace Reference

### Functions

- def **watch\_user** (query, token)
- def **validate\_password** (password)
- def **create\_session** (user)
- def **retrieve\_session** (request)
- def **invalidate\_sessions** (username)
- def **create\_hash**
- def **hash\_str**
- def **has\_permission** (permission\_tree, path)
- def **bad\_creds\_response** ()
- def **forbidden\_response** (error\_msg)
- def **requires\_auth** (func=None, permission\_expr=None)
- def **db\_create\_user** (username, password, permissions)
- def **api\_create\_user** (request)
- def **api\_auth** (request)
- def **verify\_token** (request)
- def **logout** (request)



## Variables

- dictionary **sessions** = {}

## 4.2.1 Function Documentation

### 4.2.1.1 `api_auth()`

```
def auth.api_auth (
    request )
```

Authenticates a user and creates a session if the username and password in the request object matches a database entry.

:param request: aiohttp request object  
:return: an aiohttp response object, 401 or 200 with token generated, username, gravatar-url and gravatar-email

### 4.2.1.2 `api_create_user()`

```
def auth.api_create_user (
    request )
```

aiohttp function to create a user in the database

:param request: aiohttp request  
:return: An aiohttp response object, 422 or 201

### 4.2.1.3 `bad_creds_response()`

```
def auth.bad_creds_response ( )
```

Creates and returns a 401 http response  
:return: The generated 401 http response

### 4.2.1.4 `create_hash()`

```
def auth.create_hash (
    to_hash )
```

#### 4.2.1.5 create\_session()

```
def auth.create_session (  
    user )
```

Creates a session for the given user and returns a generated session token.

:param user: user object of the user to create a session for  
:return: The generated session-token

#### 4.2.1.6 db\_create\_user()

```
def auth.db_create_user (  
    username,  
    password,  
    permissions )
```

Creates a user object in the database for the given username, password and permission tree.

:param username: Username for the user  
:param password: Password in plain-text for the user  
:param permissions: The permission-tree for the user.  
:return: Database response

#### 4.2.1.7 forbidden\_response()

```
def auth.forbidden_response (  
    error_msg )
```

Creates and returns a 403 forbidden response with the given error msg contained in the body.

:param error\_msg: The message to put in the response  
:return: The generated 403 http response

#### 4.2.1.8 has\_permission()

```
def auth.has_permission (  
    permission_tree,  
    path )
```

Returns True or False depending on if the given permission dict tree contains the given path.

:param permission\_tree: A dictionary tree  
:param path: A list of keys to traverse the tree with  
:return: True if the tree contains the given path list, false otherwise

#### 4.2.1.9 hash\_str()

```
def auth.hash_str (
    to_hash )
```

#### 4.2.1.10 invalidate\_sessions()

```
def auth.invalidate_sessions (
    username )
```

Invalidates all sessions for the given username

:param username: username

#### 4.2.1.11 logout()

```
def auth.logout (
    request )
```

Invalidates the session of the token given in the request.

:param request: aiohttp request object  
:return: aiohttp web response object

#### 4.2.1.12 requires\_auth()

```
def auth.requires_auth (
    func = None,
    permission_expr = None )
```

A decorator to use on aiohttp endpoints to run authentication on all requests before calling the endpoint function.

:param func: The function to wrap  
:param permission\_expr: The permission expression to check  
:return: The decorated function

#### 4.2.1.13 retrieve\_session()

```
def auth.retrieve_session (
    request )
```

Retrieves the stored session object stored on the session token contained in the given aiohttp request object.

:param request: The aiohttp request object that contains the session token  
:return: The session user object

#### 4.2.1.14 validate\_password()

```
def auth.validate_password (
    password )
```

#### 4.2.1.15 verify\_token()

```
def auth.verify_token (
    request )
```

Verifies the X-CSRF-Token contained in the request object.

:param request: aiohttp request object  
:return: 200 if valid token, 401 otherwise

#### 4.2.1.16 watch\_user()

```
def auth.watch_user (
    query,
    token )
```

### 4.2.2 Variable Documentation

#### 4.2.2.1 sessions

```
dictionary auth.sessions = {}
```

## 4.3 cion\_system Namespace Reference

### Functions

- def **get\_health** (request)

#### 4.3.1 Function Documentation

##### 4.3.1.1 get\_health()

```
def cion_system.get_health (
    request )

aiohttp endpoint to act as a healthcheck

:param request: aiohttp request object
:return: an aiohttp response object with http status code **200**.
```

## 4.4 documents Namespace Reference

### Functions

- def **lazy** (fn)
- def **editable\_documents** ()
- def **sort\_array\_values** (d)
- def **db\_get\_document** (doc\_name, kwargs)
- def **generate\_permission\_def** (swarms)
- def **get\_document** (request)
- def **set\_document** (request)
- def **get\_documents** (request)
- def **get\_permission\_def** (request)
- def **json** (data, status=200, kwargs)

#### 4.4.1 Function Documentation

##### 4.4.1.1 db\_get\_document()

```
def documents.db_get_document (
    doc_name,
    kwargs )

Gets and returns the document by the given name from the database.

:param doc_name: Name of the document to fetch
:return: The fetched document
```

#### 4.4.1.2 `editable_documents()`

```
def documents.editable_documents ( )
```

#### 4.4.1.3 `generate_permission_def()`

```
def documents.generate_permission_def (
    swarms )
```

#### 4.4.1.4 `get_document()`

```
def documents.get_document (
    request )
```

Aiohttp endpoint to fetch a document from the database  
:param request: aiohttp request object  
:return: the fetched document in an aiohttp request object

#### 4.4.1.5 `get_documents()`

```
def documents.get_documents (
    request )
```

Aiohttp endpoint to fetch all documents from the database who have  
``plaintext-editable`` set to true.

:param request: aiohttp request object  
:return: aiohttp response object

#### 4.4.1.6 `get_permission_def()`

```
def documents.get_permission_def (
    request )
```

Aiohttp endpoint to fetch the definition for the permission tree.

:param request: aiohttp request  
:return: aiohttp response with permission definition tree in body

#### 4.4.1.7 json()

```
def documents.json (
    data,
    status = 200,
    kwargs )
```

Creates a json webresponse

:param status: HTTP status code of response  
:param data: json data to respond with  
:param kwargs: arguments to json.dumps  
:return: Web response

#### 4.4.1.8 lazy()

```
def documents.lazy (
    fn )
```

Decorator that ensures the function is only called once, the result is cached and reused on successive calls.

:param fn: Function to decorate  
:return: The lazy function

#### 4.4.1.9 set\_document()

```
def documents.set_document (
    request )
```

Replaces a document with the given in the database with the given document body.

:param request: aiohttp request object  
:return: aiohttp response object with a **\*\*200\*\*** http status code.

#### 4.4.1.10 sort\_array\_values()

```
def documents.sort_array_values (
    d )
```

Sorts a dict's values

:param d: dictionary, can be a tree  
:return: The sorted dictionary

## 4.5 environments Namespace Reference

### Functions

- def **db\_create\_environment** (mode, name, tag\_match, tls=None)
- def **create\_environment** (request)
- def **get\_environments** (request)
- def **get\_webhook** (request)

### 4.5.1 Function Documentation

#### 4.5.1.1 create\_environment()

```
def environments.create_environment (  
    request )
```

#### 4.5.1.2 db\_create\_environment()

```
def environments.db_create_environment (  
    mode,  
    name,  
    tag_match,  
    tls = None )
```

#### 4.5.1.3 get\_environments()

```
def environments.get_environments (  
    request )
```

#### 4.5.1.4 get\_webhook()

```
def environments.get_webhook (  
    request )
```

## 4.6 permissions Namespace Reference

### Namespaces

- **permission**



## 4.7 permissions.permission Namespace Reference

### Classes

- class **Permission**

### Functions

- def **perm** (path, resolve\_placeholders=None)

#### 4.7.1 Function Documentation

##### 4.7.1.1 perm()

```
def permissions.permission.perm (
    path,
    resolve_placeholders = None )
```

Creates and returns a permission object.

Placeholders are values that begin with the character `*$*`.

```resolve_placeholders``` has ot be a function that takes an aiohttp request as a parameter and returns a dictionary containing the values for the placeholders defined in the path.

So the permission-path `*$env.user*` would need a ```resolve_placeholders``` function that returns a dictionary with the key `*env*` having a value.

```
:param path: A period-separated string representing a path
:param resolve_placeholders: A function resolve placeholders
:return: The generated permission object
```

## 4.8 rdb\_conn Namespace Reference

### Functions

- def **init** ()
- def **ensure\_table\_exists** (table\_name, primary\_key='id', func=None, indices=None)
- def **ensure\_db\_exists** (db\_name)
- def **create\_admin\_user\_insert** ()

### Variables

- **Connection**

## 4.8.1 Function Documentation

### 4.8.1.1 `create_admin_user_insert()`

```
def rdb_conn.create_admin_user_insert ( )
```

Creates the admin user table row

:return: table entry representing the admin user

### 4.8.1.2 `ensure_db_exists()`

```
def rdb_conn.ensure_db_exists (
    db_name )
```

Creates a database by the given name if it does not exist.

:param db\_name: name of the database  
:return: database response

### 4.8.1.3 `ensure_table_exists()`

```
def rdb_conn.ensure_table_exists (
    table_name,
    primary_key = 'id',
    func = None,
    indices = None )
```

Creates a table in the database if it does not exist.

:param table\_name: name of the table  
:param primary\_key: field to use as primary key, default \*id\*  
:param func: rethinkdb function to run on the database after the table has  
been created, if it did not previously exist  
:param indices: a list of strings; indexes to create on the table after  
it's creation  
:return: the database response

#### 4.8.1.4 init()

```
def rdb_conn.init ( )
```

Initializes the database connection. And sets up database, table and default data if they do not exist

:return: connection object

### 4.8.2 Variable Documentation

#### 4.8.2.1 Connection

```
rdb_conn.Connection
```

## 4.9 search Namespace Reference

### Functions

- def **match** (value, expr)
- def **parse\_date\_time** (val)
- def **traverse** (row, group, name)
- def **get\_filter** (search\_string)

### Variables

- string **filter\_string** = "event:(asdf) AND status:done"
- list **rows**
- def **filter\_func** = **get\_filter**( **filter\_string**)

#### 4.9.1 Function Documentation

##### 4.9.1.1 get\_filter()

```
def search.get_filter (
    search_string )
```

#### 4.9.1.2 match()

```
def search.match (
    value,
    expr )
```

#### 4.9.1.3 parse\_date\_time()

```
def search.parse_date_time (
    val )
```

#### 4.9.1.4 traverse()

```
def search.traverse (
    row,
    group,
    name )
```

### 4.9.2 Variable Documentation

#### 4.9.2.1 filter\_func

```
def search.filter_func = get_filter( filter_string)
```

#### 4.9.2.2 filter\_string

```
string search.filter_string = "event:(asdf) AND status:done"
```

#### 4.9.2.3 rows

```
list search.rows
```

## 4.10 services Namespace Reference

### Functions

- def **validate\_input** (string)
- def **url\_safe\_service\_image** (service\_name, image\_name)
- def **task\_base\_image\_name\_filter** (glob, image\_base\_name)
- def **db\_get\_service\_conf** (service\_name)
- def **db\_create\_service** (service\_name, environments, image\_name)
- def **db\_replace\_service** (service\_name, environments, image\_name)
- def **db\_get\_running\_image** (service\_name)
- def **db\_get\_unique\_deployed\_images** (service\_name)
- def **db\_get\_services** ()
- def **db\_delete\_service** (service\_name)
- def **get\_services** (request)
- def **get\_running\_image** (request)
- def **get\_service** (request)
- def **resolve\_service\_create** (request)
- def **create\_service** (request)
- def **edit\_service** (request)
- def **resolve\_service\_delete** (request)
- def **delete\_service** (request)

### 4.10.1 Function Documentation

#### 4.10.1.1 create\_service()

```
def services.create_service (  
    request )
```

aiohttp endpoint to create a service configuration

#### 4.10.1.2 db\_create\_service()

```
def services.db_create_service (  
    service_name,  
    environments,  
    image_name )
```

Creates a service configuration in the database

```
:param service_name: name of the service  
:param environments: environments for the service configuration  
:param image_name: image name for the service configuration  
:return:
```

#### 4.10.1.3 db\_delete\_service()

```
def services.db_delete_service (
    service_name )
```

Deletes a service configuration from the database

:param service\_name: name of service to delete service configuration for  
:return: database result

#### 4.10.1.4 db\_get\_running\_image()

```
def services.db_get_running_image (
    service_name )
```

Gets the last updated image for a given service name.

:param service\_name: name of the service to get image for  
:return: running image

#### 4.10.1.5 db\_get\_service\_conf()

```
def services.db_get_service_conf (
    service_name )
```

Fetches and returns configuration for the given service name

:param service\_name: name of the service to get configuration for  
:return: service configuration dictionary

#### 4.10.1.6 db\_get\_services()

```
def services.db_get_services ( )
```

Fetches all service configurations

:return: service configurations

#### 4.10.1.7 db\_get\_unique\_deployed\_images()

```
def services.db_get_unique_deployed_images (
    service_name )
```

#### 4.10.1.8 db\_replace\_service()

```
def services.db_replace_service (
    service_name,
    environments,
    image_name )
```

Updates a service configuration

```
:param service_name: service name of the configuration to update
:param environments: environments to update
:param image_name: image name to update
:return:
```

#### 4.10.1.9 delete\_service()

```
def services.delete_service (
    request )
```

aiohttp endpoint to delete a service configuration

#### 4.10.1.10 edit\_service()

```
def services.edit_service (
    request )
```

aiohttp endpoint to edit a service configuration

#### 4.10.1.11 get\_running\_image()

```
def services.get_running_image (
    request )
```

```
aiohttp endpoint to get the running image of a configured service
:param request: aiohttp request object
:return: aiohttp response with image name
```

#### 4.10.1.12 `get_service()`

```
def services.get_service (
    request )
```

aiohttp endpoint to fetch a service configuration

:return: service configuration for the service name contained in the request

#### 4.10.1.13 `get_services()`

```
def services.get_services (
    request )
```

aiohttp endpoint to fetch all service configuration

:param request: aiohttp request object  
:return: service configurations

#### 4.10.1.14 `resolve_service_create()`

```
def services.resolve_service_create (
    request )
```

Permission placeholder resolver for ``create\_service``

:param request: aiohttp request object  
:return: a dictionary containing environments from the request object

#### 4.10.1.15 `resolve_service_delete()`

```
def services.resolve_service_delete (
    request )
```

Permission placeholder resolver for the service delete endpoint

:param request: aiohttp request object  
:return: dictionary containing placeholder values from the request object



#### 4.10.1.16 task\_base\_image\_name\_filter()

```
def services.task_base_image_name_filter (
    glob,
    image_base_name )
```

```
:param glob:
:param image_base_name:
:return:
```

#### 4.10.1.17 url\_safe\_service\_image()

```
def services.url_safe_service_image (
    service_name,
    image_name )
```

Validates the given strings and returns an aiohttp error response if they do not validate.

```
:param service_name: name of the service
:param image_name: name of the image
:return: aiohttp response if given strings do not validate
```

#### 4.10.1.18 validate\_input()

```
def services.validate_input (
    string )
```

Validates the given string

```
:param string: string to validate
:return: the regex match search
```

## 4.11 table Namespace Reference

### Functions

- def **table\_query** (request, table\_name)

#### 4.11.1 Function Documentation

#### 4.11.1.1 `table_query()`

```
def table.table_query (
    request,
    table_name )
```

## 4.12 tasks Namespace Reference

### Functions

- def **db\_create\_task** (image, environment, service\_name)
- def **db\_create\_scheduled\_task** (schedule\_at, event, image, environment, service\_name)
- def **resolve\_task\_create** (request)
- def **create\_task** (request)
- def **schedule\_deploy** (request)
- def **get\_recent\_tasks** (request)
- def **get\_task** (request)
- def **get\_tasks** (request)

#### 4.12.1 Function Documentation

##### 4.12.1.1 `create_task()`

```
def tasks.create_task (
    request )
```

aihttp endpoint to create a task in the database

##### 4.12.1.2 `db_create_scheduled_task()`

```
def tasks.db_create_scheduled_task (
    schedule_at,
    event,
    image,
    environment,
    service_name )
```

Creates an entry in the `delayed_tasks` table

```
:param schedule_at: date for when the task is to be scheduled for
:param event: event of the task
:param image: the image-field of the task
:param environment: what environment the task refers to
:param service_name: service-field of the task
:return:
```

#### 4.12.1.3 db\_create\_task()

```
def tasks.db_create_task (
    image,
    environment,
    service_name )
```

Creates a task in the database

```
:param image: image-name field
:param environment: environment field
:param service_name: service-name field
:return: database result
```

#### 4.12.1.4 get\_recent\_tasks()

```
def tasks.get_recent_tasks (
    request )
```

#### 4.12.1.5 get\_task()

```
def tasks.get_task (
    request )
```

#### 4.12.1.6 get\_tasks()

```
def tasks.get_tasks (
    request )
```

Gets tasks from the database using the following query params:

- pageStart: starting page
- pageLength: length of page
- sortIndex: index to sort by
- searchTerm: lucene search term to filter the query by

#### 4.12.1.7 resolve\_task\_create()

```
def tasks.resolve_task_create (
    request )
```

Permission placeholder resolver function for the task create endpoint

```
:param request: aiohttp request object
:return: dictionary containing values for the placeholders in the
        permission path
```

#### 4.12.1.8 schedule\_deploy()

```
def tasks.schedule_deploy (
    request )
```

aihttp endpoint to schedule a deployment task in the database

### 4.13 user Namespace Reference

#### Functions

- def **db\_set\_gravatar\_email** (username, gravatar\_email)
- def **db\_get\_users** ()
- def **db\_change\_password** (username, password)
- def **db\_delete\_user** (username)
- def **db\_get\_permissions** (username)
- def **db\_set\_permissions** (username, permissions)
- def **bad\_creds\_response** ()
- def **get\_permissions** (request)
- def **set\_permissions** (request)
- def **set\_gravatar\_email** (request)
- def **get\_users** (request)
- def **help\_change\_password** (request\_body, username)
- def **change\_password** (request)
- def **change\_own\_password** (request)
- def **delete\_user** (request)

#### 4.13.1 Function Documentation

##### 4.13.1.1 bad\_creds\_response()

```
def user.bad_creds_response ( )
```

Generates an aihttp response with http status code 401

:return: the generated response object

##### 4.13.1.2 change\_own\_password()

```
def user.change_own_password (
    request )
```

Updates the password for the specified user.

Username comes from the request url.

#### 4.13.1.3 change\_password()

```
def user.change_password (
    request )
```

Updates the password for the specified user.

Username comes from the request url.

#### 4.13.1.4 db\_change\_password()

```
def user.db_change_password (
    username,
    password )
```

Sets the password for the given username to the given value

```
:param username: username
:param password: plain-text password to generate hash-information for
:return: database result
```

#### 4.13.1.5 db\_delete\_user()

```
def user.db_delete_user (
    username )
```

Deletes a user from the database

```
:param username: username for the user to delete
:return: database result
```

#### 4.13.1.6 db\_get\_permissions()

```
def user.db_get_permissions (
    username )
```

Fetches the permission tree for the given username

```
:param username: username to fetch permissions for
:return: database result
```

#### 4.13.1.7 db\_get\_users()

```
def user.db_get_users ( )
```

Gets all users from the dictionary

#### 4.13.1.8 db\_set\_gravatar\_email()

```
def user.db_set_gravatar_email (
    username,
    gravatar_email )
```

Sets the gravatar field for the given user to the given value in the database

#### 4.13.1.9 db\_set\_permissions()

```
def user.db_set_permissions (
    username,
    permissions )
```

Sets the permission tree for the given username to the given permission tree.

```
:param username: username for the user to update
:param permissions: new value for the permission-tree
:return: database result
```

#### 4.13.1.10 delete\_user()

```
def user.delete_user (
    request )
```

aihttp endpoint to delete a user from the database.

Username comes from a path parameter in the request url.

#### 4.13.1.11 `get_permissions()`

```
def user.get_permissions (
    request )
```

aihttp endpoint to fetch a user's permission tree

The username comes from a path parameter in the request url

#### 4.13.1.12 `get_users()`

```
def user.get_users (
    request )
```

aihttp endpoint to fetch all users from the database

#### 4.13.1.13 `help_change_password()`

```
def user.help_change_password (
    request_body,
    username )
```

#### 4.13.1.14 `set_gravatar_email()`

```
def user.set_gravatar_email (
    request )
```

aihttp endpoint to update the gravatar email field in the database for the currently logged in user.

Gravatar value comes from the body

#### 4.13.1.15 `set_permissions()`

```
def user.set_permissions (
    request )
```

Updates the permission-tree for a user.

Username comes from a path-parameter. Permission tree comes from the request body.

## 4.14 webhooks Namespace Reference

### Functions

- def **db\_create\_webhook** (url, event, match\_on, headers, data)
- def **create\_webhook** (request)
- def **get\_webhooks** (request)
- def **get\_webhook** (request)
- def **delete\_webhook** (request)

### 4.14.1 Function Documentation

#### 4.14.1.1 create\_webhook()

```
def webhooks.create_webhook (  
    request )
```

#### 4.14.1.2 db\_create\_webhook()

```
def webhooks.db_create_webhook (  
    url,  
    event,  
    match_on,  
    headers,  
    data )
```

#### 4.14.1.3 delete\_webhook()

```
def webhooks.delete_webhook (  
    request )
```

#### 4.14.1.4 get\_webhook()

```
def webhooks.get_webhook (  
    request )
```



4.14.1.5 `get_webhooks()`

```
def webhooks.get_webhooks (
    request )
```

## 4.15 websocket Namespace Reference

### Classes

- class **WebSocketListener**

### Functions

- def **create** (conn)

#### 4.15.1 Function Documentation

4.15.1.1 `create()`

```
def websocket.create (
    conn )
```



## Chapter 5

# Class Documentation

### 5.1 permissions.permission.Permission Class Reference

#### Public Member Functions

- def **\_\_init\_\_** (self, **check\_fn**)
- def **has\_permission** (self, args, kwargs)
- def **\_\_and\_\_** (self, other)
- def **\_\_or\_\_** (self, other)

#### Public Attributes

- **check\_fn**

#### 5.1.1 Detailed Description

Represents on permission path

#### 5.1.2 Constructor & Destructor Documentation

##### 5.1.2.1 \_\_init\_\_()

```
def permissions.permission.Permission.__init__ (
    self,
    check_fn )
```

#### 5.1.3 Member Function Documentation

#### 5.1.3.1 `__and__()`

```
def permissions.permission.Permission.__and__ (
    self,
    other )
```

Creates and returns a new permission object which runs check on self and given other-permissions object and returns true if self check **\*\*and\*\*** other check returns true.

:param other: Permission object to compare  
:return: A new permission object with a check function that returns true if self or other check returns true

#### 5.1.3.2 `__or__()`

```
def permissions.permission.Permission.__or__ (
    self,
    other )
```

Creates and returns a new permission object which runs check on self and given other-permissions object and returns true if self check **\*\*or\*\*** other check returns true.

:param other: Permission object to compare  
:return: A new permission object with a check function that returns true if self or other check returns true

#### 5.1.3.3 `has_permission()`

```
def permissions.permission.Permission.has_permission (
    self,
    args,
    kwargs )
```

Runs check function  
:return: result of check function

### 5.1.4 Member Data Documentation

#### 5.1.4.1 `check_fn`

permissions.permission.Permission.check\_fn

The documentation for this class was generated from the following file:

- src/permissions/ **permission.py**

## 5.2 websocket.WebSocketListener Class Reference

### Public Member Functions

- `def __init__(self, conn)`
- `def handle_request(self, request)`
- `def subscribe(self, ws, message)`
- `def unsubscribe(self, ws, message)`

### Public Attributes

- `clients`
- `subscriptions`
- `conn`

### Static Public Attributes

- dictionary `dispatch`

### 5.2.1 Constructor & Destructor Documentation

#### 5.2.1.1 \_\_init\_\_()

```
def websocket.WebSocketListener.__init__ (
    self,
    conn )
```

### 5.2.2 Member Function Documentation

#### 5.2.2.1 handle\_request()

```
def websocket.WebSocketListener.handle_request (
    self,
    request )
```

#### 5.2.2.2 subscribe()

```
def websocket.WebSocketListener.subscribe (
    self,
    ws,
    message )
```

#### 5.2.2.3 unsubscribe()

```
def websocket.WebSocketListener.unsubscribe (
    self,
    ws,
    message )
```

### 5.2.3 Member Data Documentation

#### 5.2.3.1 clients

websocket.WebSocketListener.clients

#### 5.2.3.2 conn

websocket.WebSocketListener.conn

#### 5.2.3.3 dispatch

dictionary websocket.WebSocketListener.dispatch [static]

##### Initial value:

```
= {
    "subscribe": subscribe,
    "unsubscribe": unsubscribe,
}
```

#### 5.2.3.4 subscriptions

websocket.WebSocketListener.subscriptions

The documentation for this class was generated from the following file:

- src/ **websocket.py**

## Chapter 6

# File Documentation

### 6.1 src/app.py File Reference

#### Namespaces

- **app**

#### Functions

- **def app.index** (request)

#### Variables

- **app.app** = web.Application()
- **int app.prod** = 'prod'
- **app.static\_path** = os.path.join(os.environ['WEB\_DIR'], 'lib')
- **app.ws\_route** = **websocket.create**(rdb\_conn.conn)
- **app.host**
- **app.port**

### 6.2 src/auth.py File Reference

#### Namespaces

- **auth**

## Functions

- def **auth.watch\_user** (query, token)
- def **auth.validate\_password** (password)
- def **auth.create\_session** (user)
- def **auth.retrieve\_session** (request)
- def **auth.invalidate\_sessions** (username)
- def **auth.create\_hash**
- def **auth.hash\_str**
- def **auth.has\_permission** (permission\_tree, path)
- def **auth.bad\_creds\_response** ()
- def **auth.forbidden\_response** (error\_msg)
- def **auth.requires\_auth** (func=None, permission\_expr=None)
- def **auth.db\_create\_user** (username, password, permissions)
- def **auth.api\_create\_user** (request)
- def **auth.api\_auth** (request)
- def **auth.verify\_token** (request)
- def **auth.logout** (request)

## Variables

- dictionary **auth.sessions** = {}

## 6.3 src/cion\_system.py File Reference

### Namespaces

- **cion\_system**

### Functions

- def **cion\_system.get\_health** (request)

## 6.4 src/documents.py File Reference

### Namespaces

- **documents**

### Functions

- def **documents.lazy** (fn)
- def **documents.editable\_documents** ()
- def **documents.sort\_array\_values** (d)
- def **documents.db\_get\_document** (doc\_name, kwargs)
- def **documents.generate\_permission\_def** (swarms)
- def **documents.get\_document** (request)
- def **documents.set\_document** (request)
- def **documents.get\_documents** (request)
- def **documents.get\_permission\_def** (request)
- def **documents.json** (data, status=200, kwargs)



## 6.5 src/environments.py File Reference

### Namespaces

- **environments**

### Functions

- def **environments.db\_create\_environment** (mode, name, tag\_match, tls=None)
- def **environments.create\_environment** (request)
- def **environments.get\_environments** (request)
- def **environments.get\_webhook** (request)

## 6.6 src/permissions/\_\_init\_\_.py File Reference

### Namespaces

- **permissions**

## 6.7 src/permissions/permission.py File Reference

### Classes

- class **permissions.permission.Permission**

### Namespaces

- **permissions.permission**

### Functions

- def **permissions.permission.perm** (path, resolve\_placeholders=None)

## 6.8 src/rdb\_conn.py File Reference

### Namespaces

- **rdb\_conn**

### Functions

- def **rdb\_conn.init** ()
- def **rdb\_conn.ensure\_table\_exists** (table\_name, primary\_key='id', func=None, indices=None)
- def **rdb\_conn.ensure\_db\_exists** (db\_name)
- def **rdb\_conn.create\_admin\_user\_insert** ()

## Variables

- `rdb_conn.Connection`

## 6.9 `src/search.py` File Reference

### Namespaces

- `search`

### Functions

- `def search.match (value, expr)`
- `def search.parse_date_time (val)`
- `def search.traverse (row, group, name)`
- `def search.get_filter (search_string)`

### Variables

- `string search.filter_string = "event:(asdf) AND status:done"`
- `list search.rows`
- `def search.filter_func = get_filter(filter_string)`

## 6.10 `src/services.py` File Reference

### Namespaces

- `services`

### Functions

- `def services.validate_input (string)`
- `def services.url_safe_service_image (service_name, image_name)`
- `def services.task_base_image_name_filter (glob, image_base_name)`
- `def services.db_get_service_conf (service_name)`
- `def services.db_create_service (service_name, environments, image_name)`
- `def services.db_replace_service (service_name, environments, image_name)`
- `def services.db_get_running_image (service_name)`
- `def services.db_get_unique_deployed_images (service_name)`
- `def services.db_get_services ()`
- `def services.db_delete_service (service_name)`
- `def services.get_services (request)`
- `def services.get_running_image (request)`
- `def services.get_service (request)`
- `def services.resolve_service_create (request)`
- `def services.create_service (request)`
- `def services.edit_service (request)`
- `def services.resolve_service_delete (request)`
- `def services.delete_service (request)`

## 6.11 src/table.py File Reference

### Namespaces

- **table**

### Functions

- def **table.table\_query** (request, table\_name)

## 6.12 src/tasks.py File Reference

### Namespaces

- **tasks**

### Functions

- def **tasks.db\_create\_task** (image, environment, service\_name)
- def **tasks.db\_create\_scheduled\_task** (schedule\_at, event, image, environment, service\_name)
- def **tasks.resolve\_task\_create** (request)
- def **tasks.create\_task** (request)
- def **tasks.schedule\_deploy** (request)
- def **tasks.get\_recent\_tasks** (request)
- def **tasks.get\_task** (request)
- def **tasks.get\_tasks** (request)

## 6.13 src/user.py File Reference

### Namespaces

- **user**

### Functions

- def **user.db\_set\_gravatar\_email** (username, gravatar\_email)
- def **user.db\_get\_users** ()
- def **user.db\_change\_password** (username, password)
- def **user.db\_delete\_user** (username)
- def **user.db\_get\_permissions** (username)
- def **user.db\_set\_permissions** (username, permissions)
- def **user.bad\_creds\_response** ()
- def **user.get\_permissions** (request)
- def **user.set\_permissions** (request)
- def **user.set\_gravatar\_email** (request)
- def **user.get\_users** (request)
- def **user.help\_change\_password** (request\_body, username)
- def **user.change\_password** (request)
- def **user.change\_own\_password** (request)
- def **user.delete\_user** (request)

## 6.14 src/webhooks.py File Reference

### Namespaces

- **webhooks**

### Functions

- def **webhooks.db\_create\_webhook** (url, event, match\_on, headers, data)
- def **webhooks.create\_webhook** (request)
- def **webhooks.get\_webhooks** (request)
- def **webhooks.get\_webhook** (request)
- def **webhooks.delete\_webhook** (request)

## 6.15 src/websocket.py File Reference

### Classes

- class **websocket.WebSocketListener**

### Namespaces

- **websocket**

### Functions

- def **websocket.create** (conn)

# Index

- `__and__`
    - `permissions::permission::Permission`, 35
  - `__init__`
    - `permissions::permission::Permission`, 35
    - `websocket::WebSocketListener`, 37
  - `__or__`
    - `permissions::permission::Permission`, 36
- `api_auth`
  - `auth`, 9
- `api_create_user`
  - `auth`, 9
- `app`, 7
  - `app`, 7
  - `host`, 8
  - `index`, 7
  - `port`, 8
  - `prod`, 8
  - `static_path`, 8
  - `ws_route`, 8
- `auth`, 8
  - `api_auth`, 9
  - `api_create_user`, 9
  - `bad_creds_response`, 9
  - `create_hash`, 9
  - `create_session`, 9
  - `db_create_user`, 10
  - `forbidden_response`, 10
  - `has_permission`, 10
  - `hash_str`, 10
  - `invalidate_sessions`, 11
  - `logout`, 11
  - `requires_auth`, 11
  - `retrieve_session`, 11
  - `sessions`, 12
  - `validate_password`, 12
  - `verify_token`, 12
  - `watch_user`, 12
- `bad_creds_response`
  - `auth`, 9
  - `user`, 28
- `change_own_password`
  - `user`, 28
- `change_password`
  - `user`, 28
- `check_fn`
  - `permissions::permission::Permission`, 36
- `cion_system`, 13
  - `get_health`, 13
- `clients`
  - `websocket::WebSocketListener`, 38
- `conn`
  - `websocket::WebSocketListener`, 38
- `Connection`
  - `rdb_conn`, 19
- `create`
  - `websocket`, 33
- `create_admin_user_insert`
  - `rdb_conn`, 18
- `create_environment`
  - `environments`, 16
- `create_hash`
  - `auth`, 9
- `create_service`
  - `services`, 21
- `create_session`
  - `auth`, 9
- `create_task`
  - `tasks`, 26
- `create_webhook`
  - `webhooks`, 32
- `db_change_password`
  - `user`, 29
- `db_create_environment`
  - `environments`, 16
- `db_create_scheduled_task`
  - `tasks`, 26
- `db_create_service`
  - `services`, 21
- `db_create_task`
  - `tasks`, 26
- `db_create_user`
  - `auth`, 10
- `db_create_webhook`
  - `webhooks`, 32
- `db_delete_service`
  - `services`, 21
- `db_delete_user`
  - `user`, 29
- `db_get_document`
  - `documents`, 13
- `db_get_permissions`
  - `user`, 29
- `db_get_running_image`
  - `services`, 22
- `db_get_service_conf`
  - `services`, 22

- db\_get\_services
  - services, 22
- db\_get\_unique\_deployed\_images
  - services, 22
- db\_get\_users
  - user, 29
- db\_replace\_service
  - services, 23
- db\_set\_gravatar\_email
  - user, 30
- db\_set\_permissions
  - user, 30
- delete\_service
  - services, 23
- delete\_user
  - user, 30
- delete\_webhook
  - webhooks, 32
- dispatch
  - websocket::WebSocketListener, 38
- documents, 13
  - db\_get\_document, 13
  - editable\_documents, 13
  - generate\_permission\_def, 14
  - get\_document, 14
  - get\_documents, 14
  - get\_permission\_def, 14
  - json, 14
  - lazy, 15
  - set\_document, 15
  - sort\_array\_values, 15
- edit\_service
  - services, 23
- editable\_documents
  - documents, 13
- ensure\_db\_exists
  - rdb\_conn, 18
- ensure\_table\_exists
  - rdb\_conn, 18
- environments, 16
  - create\_environment, 16
  - db\_create\_environment, 16
  - get\_environments, 16
  - get\_webhook, 16
- filter\_func
  - search, 20
- filter\_string
  - search, 20
- forbidden\_response
  - auth, 10
- generate\_permission\_def
  - documents, 14
- get\_document
  - documents, 14
- get\_documents
  - documents, 14
- get\_environments
  - environments, 16
- get\_filter
  - search, 19
- get\_health
  - cion\_system, 13
- get\_permission\_def
  - documents, 14
- get\_permissions
  - user, 30
- get\_recent\_tasks
  - tasks, 27
- get\_running\_image
  - services, 23
- get\_service
  - services, 23
- get\_services
  - services, 24
- get\_task
  - tasks, 27
- get\_tasks
  - tasks, 27
- get\_users
  - user, 31
- get\_webhook
  - environments, 16
  - webhooks, 32
- get\_webhooks
  - webhooks, 32
- handle\_request
  - websocket::WebSocketListener, 37
- has\_permission
  - auth, 10
  - permissions::permission::Permission, 36
- hash\_str
  - auth, 10
- help\_change\_password
  - user, 31
- host
  - app, 8
- index
  - app, 7
- init
  - rdb\_conn, 18
- invalidate\_sessions
  - auth, 11
- json
  - documents, 14
- lazy
  - documents, 15
- logout
  - auth, 11
- match
  - search, 19

- parse\_date\_time
  - search, 20
- perm
  - permissions::permission, 17
- permissions, 16
- permissions.permission, 17
- permissions.permission.Permission, 35
- permissions::permission
  - perm, 17
- permissions::permission::Permission
  - \_\_and\_\_, 35
  - \_\_init\_\_, 35
  - \_\_or\_\_, 36
  - check\_fn, 36
  - has\_permission, 36
- port
  - app, 8
- prod
  - app, 8
- rdb\_conn, 17
  - Connection, 19
  - create\_admin\_user\_insert, 18
  - ensure\_db\_exists, 18
  - ensure\_table\_exists, 18
  - init, 18
- requires\_auth
  - auth, 11
- resolve\_service\_create
  - services, 24
- resolve\_service\_delete
  - services, 24
- resolve\_task\_create
  - tasks, 27
- retrieve\_session
  - auth, 11
- rows
  - search, 20
- schedule\_deploy
  - tasks, 27
- search, 19
  - filter\_func, 20
  - filter\_string, 20
  - get\_filter, 19
  - match, 19
  - parse\_date\_time, 20
  - rows, 20
  - traverse, 20
- services, 21
  - create\_service, 21
  - db\_create\_service, 21
  - db\_delete\_service, 21
  - db\_get\_running\_image, 22
  - db\_get\_service\_conf, 22
  - db\_get\_services, 22
  - db\_get\_unique\_deployed\_images, 22
  - db\_replace\_service, 23
  - delete\_service, 23
  - edit\_service, 23
  - get\_running\_image, 23
  - get\_service, 23
  - get\_services, 24
  - resolve\_service\_create, 24
  - resolve\_service\_delete, 24
  - task\_base\_image\_name\_filter, 24
  - url\_safe\_service\_image, 25
  - validate\_input, 25
- sessions
  - auth, 12
- set\_document
  - documents, 15
- set\_gravatar\_email
  - user, 31
- set\_permissions
  - user, 31
- sort\_array\_values
  - documents, 15
- src/app.py, 39
- src/auth.py, 39
- src/cion\_system.py, 40
- src/documents.py, 40
- src/environments.py, 41
- src/permissions/\_\_init\_\_.py, 41
- src/permissions/permission.py, 41
- src/rdb\_conn.py, 41
- src/search.py, 42
- src/services.py, 42
- src/table.py, 43
- src/tasks.py, 43
- src/user.py, 43
- src/webhooks.py, 44
- src/websocket.py, 44
- static\_path
  - app, 8
- subscribe
  - websocket::WebSocketListener, 37
- subscriptions
  - websocket::WebSocketListener, 38
- table, 25
  - table\_query, 25
- table\_query
  - table, 25
- task\_base\_image\_name\_filter
  - services, 24
- tasks, 26
  - create\_task, 26
  - db\_create\_scheduled\_task, 26
  - db\_create\_task, 26
  - get\_recent\_tasks, 27
  - get\_task, 27
  - get\_tasks, 27
  - resolve\_task\_create, 27
  - schedule\_deploy, 27
- traverse
  - search, 20

- unsubscribe
  - websocket::WebSocketListener, 38
- url\_safe\_service\_image
  - services, 25
- user, 28
  - bad\_creds\_response, 28
  - change\_own\_password, 28
  - change\_password, 28
  - db\_change\_password, 29
  - db\_delete\_user, 29
  - db\_get\_permissions, 29
  - db\_get\_users, 29
  - db\_set\_gravatar\_email, 30
  - db\_set\_permissions, 30
  - delete\_user, 30
  - get\_permissions, 30
  - get\_users, 31
  - help\_change\_password, 31
  - set\_gravatar\_email, 31
  - set\_permissions, 31
- validate\_input
  - services, 25
- validate\_password
  - auth, 12
- verify\_token
  - auth, 12
- watch\_user
  - auth, 12
- webhooks, 32
  - create\_webhook, 32
  - db\_create\_webhook, 32
  - delete\_webhook, 32
  - get\_webhook, 32
  - get\_webhooks, 32
- websocket, 33
  - create, 33
- websocket.WebSocketListener, 37
- websocket::WebSocketListener
  - \_\_init\_\_, 37
  - clients, 38
  - conn, 38
  - dispatch, 38
  - handle\_request, 37
  - subscribe, 37
  - subscriptions, 38
  - unsubscribe, 38
- ws\_route
  - app, 8