
HomeControl

Jul 13, 2020

Contents:

1	dependencies module	1
1.1	Action Engine	1
1.2	Configuration Manager	1
1.3	Data Types	2
1.4	Entity Types	3
1.5	Event Engine	3
1.6	Item Manager	4
1.7	JSON Response	6
1.8	JSON	6
1.9	Module Manager	7
1.10	State Engine	7
1.11	Throttle Function	7
1.12	Tick Engine	8
1.13	Validators	8
1.14	YAML Loader	8
2	homecontrol module	11
3	API Module	13
4	Custom Scripts	15
5	PiGPIO Adapter	17
5.1	Installation	17
	Python Module Index	19
	HTTP Routing Table	21
	Index	23

1.1 Action Engine

ActionEngine for HomeControl

```
class homecontrol.dependencies.action_engine.ActionEngine (item: homecontrol.dependencies.entity_types.Item, core)
```

Bases: object

Holds available actions for an item

```
coroutine execute (name: str, *args, **kwargs) → bool
```

Executes an action, optionally with parameters

```
homecontrol.dependencies.action_engine.action (arg: Union[Callable, str]) → Callable
```

Decorator to mark a coroutine as an action

```
>>> @action
>>> async def foo(): ...
```

```
>>> @action(name)
>>> async def foo(): ...
```

1.2 Configuration Manager

config_manager module

```
class homecontrol.dependencies.config_manager.ConfigManager (cfg: dict, cfg_path: str)
```

Bases: object

Manages the configuration with configuration domains

coroutine approve_domain_config (*domain: str, config: dict, initial: bool = True*) → dict

Returns an approved and validated version of config for a domain

get (*key, default=None*)

getter for self.cfg

coroutine register_domain (*domain: str, handler: object = None, schema: voluptuous.schema_builder.Schema = None, allow_reload: bool = False, default: dict = None*) → object

Registers a configuration domain

Objects can register themselves to their own configuration domain and subscribe to changes in the configuration

Parameters

- **domain** (*str*) – The configuration domain (A top-level key in config.yaml)
- **handler** (*object*) – The object subscribing to this domain. Methods it can implement are: - `approve_configuration(domain, config)` -> bool: - `apply_new_configuration(domain, config)` -> None: If not specified then your configuration domain will not be reloadable

allow_reload (bool): Allow reloading this configuration domain

schema (voluptuous.Schema): Schema to validate the configuration and to fill in defaults

default (dict): A default configuration

Returns A validated and approved version of the configuration

coroutine reload_config () → None

Reloads the configuration and updates where it can

1.3 Data Types

data types for HomeControl that can be translated to JSON for API usage

class homecontrol.dependencies.data_types.**Color** (*h: int, s: int, l: int*)

Bases: object

Representation for a color

dump () -> (<class 'int'>, <class 'int'>, <class 'int'>)

Dumps the Color into a JSON serialisable format

staticmethod from_data (*hsl: tuple*)

Constructor from the data received through the API or configuration

staticmethod from_hsl (*hsl: tuple*)

HSL constructor

staticmethod from_rgb (*rgb: tuple*)

RGB constructor

h

Hue

l

Lightness

rgb
RGB

s
Saturation

```
class homecontrol.dependencies.data_types.DateTime
    Bases: datetime.datetime

    date time format

    dump ()
        Dump to JSON serialisable data

    staticmethod from_data (data)
        Construct from JSON serialisable data
```

1.4 Entity Types

Module containing the entity types Every new Item or Module will get one of these classes as a base class

```
class homecontrol.dependencies.entity_types.Item
    Bases: object

    A dummy Item

    config_schema = <Schema(<class 'object'>, extra=PREVENT_EXTRA, required=False) object>

    coroutine constructor (identifier: str, name: str, cfg: dict, state_defaults: dict, core: Core,
                           unique_identifier: str = None) → Item
        Constructs an item

    coroutine init () → None
        Default init method

    status = 'offline'

    coroutine stop () → None
        Default stop method

    update_status (status: homecontrol.const.ItemStatus) → None
        Updates the item status and broadcasts an event

class homecontrol.dependencies.entity_types.Module
    Bases: object

    A dummy Module

    coroutine init () → None
        Default init method

    coroutine stop () → None
        Default stop method

homecontrol.dependencies.entity_types.ModuleDef
    alias of homecontrol.dependencies.entity_types.Module
```

1.5 Event Engine

EventEngine for HomeControl

```
class homecontrol.dependencies.event_engine.Event (event_type: str, data: dict =
                                                    None, timestamp: datetime.datetime
                                                    = None, kwargs: dict = None)

    Bases: object
    Representation for an Event
    data
    event_type
    kwargs
    timestamp

class homecontrol.dependencies.event_engine.EventEngine (core)
    Bases: object
    Dispatcher for events

    broadcast (event_type: str, data: dict = None, **kwargs) → List[_asyncio.Future]
        Broadcast an event and return the futures
        Every listener is a coroutine that will simply receive event and kwargs
        Example: >>> async def on_event(event: Event, ...): >>> return

    broadcast_threaded (event_type: str, data: dict = None, **kwargs) → List[_asyncio.Task]
        Same as broadcast BUT - It returns Futures and not Tasks - It uses threads

    staticmethod create_event (event_type: str, data: dict = None, **kwargs) → homecon-
                                trol.dependencies.event_engine.Event
        Creates an Event to be broadcasted

    coroutine gather (event_type: str, data: dict = None, timeout: Union[float, int, None] = None,
                     **kwargs) → List[Any]
        Broadcast an event and return the results

    get_event_handlers (event: homecontrol.dependencies.event_engine.Event) → List[T]
        Returns a list of handlers for an Event

    register (event: str) → Callable
        Decorator to register event handlers

    remove_handler (event: str, handler: Callable) → None
        Removes an event handler
```

1.6 Item Manager

ItemManager for HomeControl

```
class homecontrol.dependencies.item_manager.ItemManager (core)
    Bases: object
    ItemManager manages all your stateful items

    coroutine add_from_module (mod_obj: homecontrol.dependencies.entity_types.Module) →
                                None
        Adds the item specifications of a module to the dict of available ones
        mod_obj: homecontrol.entity_types.Module
```

```

coroutine create_from_storage_entry (storage_entry: homecontrol.dependencies.item_manager.StorageEntry)
    → homecontrol.dependencies.entity_types.Item

    Creates an Item from a storage entry

coroutine create_item (identifier: str, item_type: str, cfg: dict = None, state_defaults: dict =
    None, name: str = None, unique_identifier: str = None) → homecontrol.dependencies.entity_types.Item

    Creates a HomeControl item

get_by_unique_identifier (unique_identifier: str) → homecontrol.dependencies.entity_types.Item

    Returns an item by its unique identifier

get_item (identifier: str) → homecontrol.dependencies.entity_types.Item

    Returns an item by identifier or unique_identifier

get_storage_entry (unique_identifier: str) → homecontrol.dependencies.item_manager.StorageEntry

    Returns the StorageEntry for a unique_identifier

coroutine init () → None

    Initialise the items from configuration

coroutine init_item (item: homecontrol.dependencies.entity_types.Item) → None

    Initialises an item

for ... in iter_items_by_id (iterable) → [<class 'homecontrol.dependencies.entity_types.Item'>]

    Translates item identifiers into item instances

load_yaml_config () → None

    Loads the YAML configuration

coroutine register_entry (storage_entry: homecontrol.dependencies.item_manager.StorageEntry,
    override: bool = False) → Optional[homecontrol.dependencies.entity_types.Item]

    Registers a storage entry

coroutine remove_item (identifier: str) → None

    Removes a HomeControl item

    identifier: str The item's identifier

coroutine stop () → None

    Stops every item

coroutine stop_item (item: homecontrol.dependencies.entity_types.Item, status: homecontrol.const.ItemStatus =
    <ItemStatus.STOPPED: 'stopped'>) → None

    Stops an item

update_storage_entry (entry: homecontrol.dependencies.item_manager.StorageEntry) → None

    Updates a config storage entry

class homecontrol.dependencies.item_manager.StorageEntry (unique_identifier: str,
    type: str, provider: str,
    enabled: bool = True,
    identifier: str = None,
    state_defaults: dict = NOTHING, cfg: dict = NOTHING, name: str =
    None, hidden: bool = False)

    Bases: object

```

The storage representation of an item

cfg
enabled
hidden
identifier
name
provider
state_defaults
type
unique_identifier

`homecontrol.dependencies.item_manager.yaml_entry_to_storage_entry` (*yaml_entry: dict*) → `homecontrol.dependencies.item_manager.StorageEntry`

Converts a yaml entry to a JSON entry

1.7 JSON Response

JSONResponse module

```
class homecontrol.dependencies.json_response.JSONResponse(data: Any = None,  
                                                         error: str = None,  
                                                         status_code: int = 200,  
                                                         core=None, headers:  
                                                         dict = None)
```

Bases: `aiohttp.web_response.Response`

A HTTP response for JSON data

1.8 JSON

JSON Encoder and Decoder

```
class homecontrol.dependencies.json.JSONEncoder(core: Core, *args, **kwargs)  
    Bases: json.encoder.JSONEncoder
```

Custom JSONEncoder that also parses HomeControl types

```
default (o)  
    Encode custom types
```

```
homecontrol.dependencies.json.dump (obj, fp, *, indent=None, sort_keys=False, core: Core =  
                                     None, **kw)
```

Dumps an object into a Writer with support for HomeControl's data types

```
homecontrol.dependencies.json.dumps (obj, *, indent=None, sort_keys=False, core: Core =  
                                       None, **kw)
```

Dumps an object into a JSON string with support for HomeControl's data types

1.9 Module Manager

ModuleManager module

```
class homecontrol.dependencies.module_manager.ModuleAccessor (module_manager:
                                                                homecon-
                                                                trol.dependencies.module_manager.Module)

    Bases: object

    Wrapper for ModuleManager.loaded_modules

class homecontrol.dependencies.module_manager.ModuleFolder (name: str)
    Bases: object

    module folder representation to create a dummy package in sys.modules

class homecontrol.dependencies.module_manager.ModuleManager (core: Core)
    Bases: object

    Manages your modules

    coroutine init () → None
        Initialise the modules

    coroutine load_file_module (mod_path: str, name: str) -> (<class 'homecon-
                                                                trol.dependencies.entity_types.Module'>, <class 'Exception'>)
        Loads a module from a file and initialises it

        Returns a Module object

    coroutine load_folder (path: str) → [<class 'object'>]
        Load every module in a folder

    coroutine load_folder_module (path: str, name: str) -> (<class 'homecon-
                                                                trol.dependencies.entity_types.Module'>, <class 'Exception'>)
        Loads a module from a folder and initialises it

        It also takes care of pip requirements

        Returns a Module object

    resource_path (module: homecontrol.dependencies.entity_types.Module, path: str = "") → str
        Returns the path for a module's resource folder Note that only folder modules can have a resource path

    coroutine stop () → None
        Unloads all modules to prepare for a shutdown
```

1.10 State Engine

1.11 Throttle Function

throttle_function

```
class homecontrol.dependencies.throttle_function.throttle (s: float = 1)
    Bases: object

    Throttles a function so it only gets called at a fixed frequency
```

1.12 Tick Engine

1.13 Validators

Config validators

```
class homecontrol.dependencies.validators.IsItem(core: Core, msg: str = None)
    Bases: object
    A validator to get an item
```

1.14 YAML Loader

Provides a YAML loader

```
class homecontrol.dependencies.yaml_loader.Constructor
    Bases: yaml.constructor.SafeConstructor
    Constructor for yaml

    env_var_constructor (node: yaml.nodes.Node) → str
        Embeds an environment variable !env_var <name> [default]

    format_string_constructor (node: yaml.nodes.Node = None) → str
        Renders a format string .. rubric:: Example
        !format { template: "Hello {who}", who: You }

    include_dir_file_mapped_constructor (node: yaml.nodes.Node = None) → dict
        !include_dir_file_mapped <folder>
        Loads multiple files from a folder and maps their contents to their filenames

    include_file_constructor (node: yaml.nodes.Node = None) → object
        !include <path> ~/ for paths relative to your home directory / for absolute paths anything else for paths
        relative to your config folder

    include_merge_constructor (node: yaml.nodes.Node = None) -> (<class 'list'>, <class
        'dict'>)
        !include <file|folder> ...
        Merges file or folder contents
        This constructor only works if all the files' contents are of same type and if this type is either list or dict.

    listdir_constructor (node: yaml.nodes.Node) → list
        !listdir <path>
        Returns the contents of a directory

    path_constructor (node: yaml.nodes.Node) → str
        !path <path> ~/ for paths relative to your home directory / for absolute paths anything else for paths relative
        to your config folder

class homecontrol.dependencies.yaml_loader.YAMLLoader(stream, cfg_folder: str =
    None)
    Bases: yaml.reader.Reader, yaml.scanner.Scanner, yaml.parser.Parser, yaml.
    composer.Composer, homecontrol.dependencies.yaml\_loader.Constructor, yaml.
    resolver.Resolver
```

Loads YAML with custom constructors

```
classmethod load (data, cfg_folder: str = None)  
    Loads data
```


CHAPTER 2

homecontrol module

The core instance for HomeControl

```
class homecontrol.core.Core (cfg: dict, cfg_file: str, loop: Optional[asyncio.events.AbstractEventLoop] = None, start_args: Optional[argparse.Namespace] = None)
```

Bases: object

Represents the root object for HomeControl

coroutine block_until_stop () → int

Blocking method to keep HomeControl running until Core.block_future is done

Also triggers the stop coroutine when block_future has a result

coroutine bootstrap () → None

Startup coroutine for Core

restart () → None

Restarts HomeControl

shutdown () → None

Shuts HomeControl down

coroutine stop () → None

Stops HomeControl

Constants used by HomeControl

```
class homecontrol.const.ItemStatus
```

Bases: enum.Enum

Every status an item can have

OFFLINE = 'offline'

ONLINE = 'online'

STOPPED = 'stopped'

WAITING_FOR_DEPENDENCY = 'waiting-for-dependency'

Exceptions

exception `homecontrol.exceptions.ConfigDomainAlreadyRegistered`

Bases: `homecontrol.exceptions.HomeControlException`

The configuration domain is already registered

exception `homecontrol.exceptions.ConfigurationNotApproved`

Bases: `homecontrol.exceptions.HomeControlException`

Configuration has not been approved

exception `homecontrol.exceptions.HomeControlException`

Bases: `BaseException`

The base exception for HomeControl

exception `homecontrol.exceptions.ItemNotFoundException`

Bases: `homecontrol.exceptions.HomeControlException`

Item not found

exception `homecontrol.exceptions.ItemNotOnlineError`

Bases: `homecontrol.exceptions.HomeControlException`

Item is not online

exception `homecontrol.exceptions.ItemTypeNotExistsError`

Bases: `homecontrol.exceptions.HomeControlException`

Item type does not exist

exception `homecontrol.exceptions.ModuleNotFoundException`

Bases: `homecontrol.exceptions.HomeControlException`

Module not found

exception `homecontrol.exceptions.PipInstallError`

Bases: `homecontrol.exceptions.HomeControlException`

A pip install failed

CHAPTER 3

API Module

The HTTP API for HomeControl Currently it supports following routes:

GET /api/ping
Ping the API

Example request:

```
GET /api/ping HTTP/ HTTP/1.1
Host: homecontrol.local:8080
Accept: application/json
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{
  "data": "PONG",
  "status_code": 200,
  "success": true
}
```

Response Headers

- **Content-Type** – application/json

Status Codes

- **200 OK** – HomeControl is online

CHAPTER 4

Custom Scripts

This module offers a very simple way to execute custom code in HomeControl Please note that this method is quite dangerous as these scripts could do anything and if you're running HomeControl as root they will even have root privileges

Provides an interface with a Raspberry Pi's GPIO ports using pigpio.

5.1 Installation

You can find an installation guide on the [official website](#).

On Raspbian you only need to run

```
sudo apt update
sudo apt install pigpio
```

The simplest way to automatically run pigpio on boot would be:

```
sudo bash -c "echo /usr/bin/pigpiod >> /etc/rc.local"
```


h

- homecontrol.const, [11](#)
- homecontrol.core, [11](#)
- homecontrol.dependencies.action_engine,
[1](#)
- homecontrol.dependencies.config_manager,
[1](#)
- homecontrol.dependencies.data_types, [2](#)
- homecontrol.dependencies.entity_types,
[3](#)
- homecontrol.dependencies.event_engine,
[3](#)
- homecontrol.dependencies.item_manager,
[4](#)
- homecontrol.dependencies.json, [6](#)
- homecontrol.dependencies.json_response,
[6](#)
- homecontrol.dependencies.module_manager,
[7](#)
- homecontrol.dependencies.throttle_function,
[7](#)
- homecontrol.dependencies.validators, [8](#)
- homecontrol.dependencies.yaml_loader, [8](#)
- homecontrol.exceptions, [12](#)

HTTP Routing Table

/api

GET /api/ping, 13

A

action() (in module *homecontrol.dependencies.action_engine*), 1
 ActionEngine (class in *homecontrol.dependencies.action_engine*), 1
 add_from_module() (*homecontrol.dependencies.item_manager.ItemManager* method), 4
 approve_domain_config() (*homecontrol.dependencies.config_manager.ConfigManager* method), 1

B

block_until_stop() (*homecontrol.core.Core* method), 11
 bootstrap() (*homecontrol.core.Core* method), 11
 broadcast() (*homecontrol.dependencies.event_engine.EventEngine* method), 4
 broadcast_threaded() (*homecontrol.dependencies.event_engine.EventEngine* method), 4

C

cfg (*homecontrol.dependencies.item_manager.StorageEntry* attribute), 6
 Color (class in *homecontrol.dependencies.data_types*), 2
 config_schema (*homecontrol.dependencies.entity_types.Item* attribute), 3
 ConfigDomainAlreadyRegistered, 12
 ConfigManager (class in *homecontrol.dependencies.config_manager*), 1
 ConfigurationNotApproved, 12
 Constructor (class in *homecontrol.dependencies.yaml_loader*), 8
 constructor() (*homecontrol.dependencies.entity_types.Item* method),

3

Core (class in *homecontrol.core*), 11
 create_event() (*homecontrol.dependencies.event_engine.EventEngine* method), 4
 create_from_storage_entry() (*homecontrol.dependencies.item_manager.ItemManager* method), 4
 create_item() (*homecontrol.dependencies.item_manager.ItemManager* method), 5

D

data (*homecontrol.dependencies.event_engine.Event* attribute), 4
 DateTime (class in *homecontrol.dependencies.data_types*), 3
 default() (*homecontrol.dependencies.json.JSONEncoder* method), 6
 dump() (*homecontrol.dependencies.data_types.Color* method), 2
 dump() (*homecontrol.dependencies.data_types.DateTime* method), 3
 dump() (in module *homecontrol.dependencies.json*), 6
 dumps() (in module *homecontrol.dependencies.json*), 6

E

enabled (*homecontrol.dependencies.item_manager.StorageEntry* attribute), 6
 env_var_constructor() (*homecontrol.dependencies.yaml_loader.Constructor* method), 8
 Event (class in *homecontrol.dependencies.event_engine*), 3
 event_type (*homecontrol.dependencies.event_engine.Event* attribute), 4
 EventEngine (class in *homecontrol.dependencies.event_engine*), 4

`execute()` (*homecontrol.dependencies.action_engine.ActionEngine method*), 1

F

`format_string_constructor()` (*homecontrol.dependencies.yaml_loader.Constructor method*), 8

`from_data()` (*homecontrol.dependencies.data_types.Color method*), 2

`from_data()` (*homecontrol.dependencies.data_types.DateTime method*), 3

`from_hsl()` (*homecontrol.dependencies.data_types.Color method*), 2

`from_rgb()` (*homecontrol.dependencies.data_types.Color method*), 2

G

`gather()` (*homecontrol.dependencies.event_engine.EventEngine method*), 4

`get()` (*homecontrol.dependencies.config_manager.ConfigManager method*), 2

`get_by_unique_identifier()` (*homecontrol.dependencies.item_manager.ItemManager method*), 5

`get_event_handlers()` (*homecontrol.dependencies.event_engine.EventEngine method*), 4

`get_item()` (*homecontrol.dependencies.item_manager.ItemManager method*), 5

`get_storage_entry()` (*homecontrol.dependencies.item_manager.ItemManager method*), 5

H

`h` (*homecontrol.dependencies.data_types.Color attribute*), 2

`hidden` (*homecontrol.dependencies.item_manager.StorageEntry attribute*), 6

`homecontrol.const` (*module*), 11

`homecontrol.core` (*module*), 11

`homecontrol.dependencies.action_engine` (*module*), 1

`homecontrol.dependencies.config_manager` (*module*), 1

`homecontrol.dependencies.data_types` (*module*), 2

`homecontrol.dependencies.entity_types` (*module*), 3

`homecontrol.dependencies.event_engine` (*module*), 3

`homecontrol.dependencies.item_manager` (*module*), 4

`homecontrol.dependencies.json` (*module*), 6

`homecontrol.dependencies.json_response` (*module*), 6

`homecontrol.dependencies.module_manager` (*module*), 7

`homecontrol.dependencies.throttle_function` (*module*), 7

`homecontrol.dependencies.validators` (*module*), 8

`homecontrol.dependencies.yaml_loader` (*module*), 8

`homecontrol.exceptions` (*module*), 12

`HomeControlException`, 12

I

`identifier` (*homecontrol.dependencies.item_manager.StorageEntry attribute*), 6

`include_dir_file_mapped_constructor()` (*homecontrol.dependencies.yaml_loader.Constructor method*), 8

`include_file_constructor()` (*homecontrol.dependencies.yaml_loader.Constructor method*), 8

`include_merge_constructor()` (*homecontrol.dependencies.yaml_loader.Constructor method*), 8

`init()` (*homecontrol.dependencies.entity_types.Item method*), 3

`init()` (*homecontrol.dependencies.entity_types.Module method*), 3

`init()` (*homecontrol.dependencies.item_manager.ItemManager method*), 5

`init()` (*homecontrol.dependencies.module_manager.ModuleManager method*), 7

`init_item()` (*homecontrol.dependencies.item_manager.ItemManager method*), 5

`IsItem` (*class in homecontrol.dependencies.validators*), 8

`Item` (*class in homecontrol.dependencies.entity_types*), 3

`ItemManager` (*class in homecontrol.dependencies.item_manager*), 4

`ItemNotFoundException`, 12

`ItemNotOnlineError`, 12

`ItemStatus` (*class in homecontrol.const*), 11

`ItemTypeNotExistsError`, 12

- `iter_items_by_id()` (*homecontrol.dependencies.item_manager.ItemManager* method), 5
- ## J
- `JSONEncoder` (class in *homecontrol.dependencies.json*), 6
- `JSONResponse` (class in *homecontrol.dependencies.json_response*), 6
- ## K
- `kwargs` (*homecontrol.dependencies.event_engine.Event* attribute), 4
- ## L
- `l` (*homecontrol.dependencies.data_types.Color* attribute), 2
- `listdir_constructor()` (*homecontrol.dependencies.yaml_loader.Constructor* method), 8
- `load()` (*homecontrol.dependencies.yaml_loader.YAMLLoader* method), 9
- `load_file_module()` (*homecontrol.dependencies.module_manager.ModuleManager* method), 7
- `load_folder()` (*homecontrol.dependencies.module_manager.ModuleManager* method), 7
- `load_folder_module()` (*homecontrol.dependencies.module_manager.ModuleManager* method), 7
- `load_yaml_config()` (*homecontrol.dependencies.item_manager.ItemManager* method), 5
- ## M
- `Module` (class in *homecontrol.dependencies.entity_types*), 3
- `ModuleAccessor` (class in *homecontrol.dependencies.module_manager*), 7
- `ModuleDef` (in module *homecontrol.dependencies.entity_types*), 3
- `ModuleFolder` (class in *homecontrol.dependencies.module_manager*), 7
- `ModuleManager` (class in *homecontrol.dependencies.module_manager*), 7
- `ModuleNotFoundException`, 12
- ## N
- `name` (*homecontrol.dependencies.item_manager.StorageEntry* attribute), 6
- ## O
- `OFFLINE` (*homecontrol.const.ItemStatus* attribute), 11
- ## P
- `path_constructor()` (*homecontrol.dependencies.yaml_loader.Constructor* method), 8
- `PipInstallError`, 12
- `provider` (*homecontrol.dependencies.item_manager.StorageEntry* attribute), 6
- ## R
- `register()` (*homecontrol.dependencies.event_engine.EventEngine* method), 4
- `register_domain()` (*homecontrol.dependencies.config_manager.ConfigManager* method), 2
- `register_entry()` (*homecontrol.dependencies.item_manager.ItemManager* method), 5
- `reload_config()` (*homecontrol.dependencies.config_manager.ConfigManager* method), 2
- `remove_handler()` (*homecontrol.dependencies.event_engine.EventEngine* method), 4
- `remove_item()` (*homecontrol.dependencies.item_manager.ItemManager* method), 5
- `resource_path()` (*homecontrol.dependencies.module_manager.ModuleManager* method), 7
- `restart()` (*homecontrol.core.Core* method), 11
- `rgb` (*homecontrol.dependencies.data_types.Color* attribute), 2
- ## S
- `s` (*homecontrol.dependencies.data_types.Color* attribute), 3
- `shutdown()` (*homecontrol.core.Core* method), 11
- `state_defaults` (*homecontrol.dependencies.item_manager.StorageEntry* attribute), 6
- `status` (*homecontrol.dependencies.entity_types.Item* attribute), 3
- `stop()` (*homecontrol.core.Core* method), 11
- `stop()` (*homecontrol.dependencies.entity_types.Item* method), 3
- `stop()` (*homecontrol.dependencies.entity_types.Module* method), 3
- `stop()` (*homecontrol.dependencies.item_manager.ItemManager* method), 5
- ## ONLINE
- `ONLINE` (*homecontrol.const.ItemStatus* attribute), 11

`stop()` (*homecontrol.dependencies.module_manager.ModuleManager*
method), 7

`stop_item()` (*homecon-*
trol.dependencies.item_manager.ItemManager
method), 5

`STOPPED` (*homecontrol.const.ItemStatus* attribute), 11

`StorageEntry` (class in *homecon-*
trol.dependencies.item_manager), 5

T

`throttle` (class in *homecon-*
trol.dependencies.throttle_function), 7

`timestamp` (*homecon-*
trol.dependencies.event_engine.Event at-
tribute), 4

`type` (*homecontrol.dependencies.item_manager.StorageEntry*
attribute), 6

U

`unique_identifier` (*homecon-*
trol.dependencies.item_manager.StorageEntry
attribute), 6

`update_status()` (*homecon-*
trol.dependencies.entity_types.Item method),
3

`update_storage_entry()` (*homecon-*
trol.dependencies.item_manager.ItemManager
method), 5

W

`WAITING_FOR_DEPENDENCY` (*homecon-*
trol.const.ItemStatus attribute), 11

Y

`yaml_entry_to_storage_entry()` (in module
homecontrol.dependencies.item_manager), 6

`YAMLLoader` (class in *homecon-*
trol.dependencies.yaml_loader), 8