
RAUC hawkBit Client Documentation

Release 0.2.0

Bastian Stender, Enrico Joerns

Mar 30, 2021

Contents

1	Contributing	3
1.1	Workflow	3
1.2	Code	3
1.3	Documentation	4
1.4	Developer’s Certificate of Origin	4
2	Changes	5
2.1	Release 0.2.0 (released Feb 20, 2020)	5
2.2	Release 0.1.0 (released Nov 20, 2017)	5
3	Modules	7
3.1	rauc_hawkbit package	7
4	Indices and Tables	13
	Python Module Index	15
	Index	17

The RAUC hawkBit client is a simple python-based application and library that runs on your target and operates as an interface between the [RAUC D-Bus API](#) and the [hawkBit DDI API](#).

Thank you for thinking about contributing to rauc-hawkbit! Some different backgrounds and use-cases are essential for making rauc-hawkbit work well for all users.

The following should help you with submitting your changes, but don't let these guidelines keep you from opening a pull request. If in doubt, we'd prefer to see the code earlier as a work-in-progress PR and help you with the submission process.

1.1 Workflow

- Changes should be submitted via a [GitHub pull request](#).
- Try to limit each commit to a single conceptual change.
- Add a signed-off-by line to your commits according to the *Developer's Certificate of Origin* (see below).
- Check that the tests still work before submitting the pull request. Also check the CI's feedback on the pull request after submission.
- When adding new features also add the corresponding documentation and test code.
- If your change affects backward compatibility, describe the necessary changes in the commit message and update the examples where needed.

1.2 Code

- Follow the [PEP 8](#) style.
- Use [isort](#) to sort the import statements.

1.3 Documentation

- Use [semantic linefeeds](#) in .rst files.

1.4 Developer's Certificate of Origin

RAUC hawkBit Client uses the [Developer's Certificate of Origin 1.1](#) with the same [process](#) as used for the Linux kernel:

Developer's Certificate of Origin 1.1

By making a contribution to this project, I certify that:

- (a) The contribution was created in whole or in part by me and I have the right to submit it under the open source license indicated in the file; or
- (b) The contribution is based upon previous work that, to the best of my knowledge, is covered under an appropriate open source license and I have the right under that license to submit that work with modifications, whether created in whole or in part by me, under the same open source license (unless I am permitted to submit under a different license), as indicated in the file; or
- (c) The contribution was provided directly to me by some other person who certified (a), (b) or (c) and I have not modified it.
- (d) I understand and agree that this project and the contribution are public and that a record of the contribution (including all personal information I submit with it, including my sign-off) is maintained indefinitely and may be redistributed consistent with this project or the open source license(s) involved.

Then you just add a line (using `git commit -s`) saying:

Signed-off-by: Random J Developer <random@developer.example.org>

using your real name (sorry, no pseudonyms or anonymous contributions).

2.1 Release 0.2.0 (released Feb 20, 2020)

- Fix `api_path` handling to be compatible with more recent hawkBit versions (failed with ‘500: Server Error’ before)
- Update to aiohttp 3.3.2 to improve timeout handling (by Livio Bieri)
- Significant speed improvements by reading maximum of available data instead of using fixed chunk sizes (by Livio Bieri)
- Prefer https [download] over http [download-http] (by Livio Bieri)
- minor cleanups and documentation fixes

2.2 Release 0.1.0 (released Nov 20, 2017)

This is the initial release of the RAUC hawkBit Client.

3.1 rauc_hawkbit package

3.1.1 Subpackages

rauc_hawkbit.ddi package

Submodules

rauc_hawkbit.ddi.cancel_action module

class rauc_hawkbit.ddi.cancel_action.**Action**(*ddi, action_id*)

Bases: object

Represents `/tenant/controller/v1/{targetid}/cancelAction/{actionId}` in HawkBit's DDI API.

feedback(*status_execution, status_result, status_details=()*)

See [#_post_tenant_controller_v1_targetid_cancelaction_actionid_feedback](http://sp.apps.bosch-iot-cloud.com/documentation/rest-api/rootcontroller-api-guide.html#_post_tenant_controller_v1_targetid_cancelaction_actionid_feedback) # noqa

class rauc_hawkbit.ddi.cancel_action.**CancelAction**(*ddi*)

Bases: object

Represents `/tenant/controller/v1/{targetid}/cancelAction` in HawkBit's DDI API.

class rauc_hawkbit.ddi.cancel_action.**CancelStatusExecution**

Bases: enum.Enum

An enumeration.

canceled = 3

closed = 1

proceeding = 2

```

rejected = 5
resumed = 6
scheduled = 4

```

```

rauc_hawkbit.ddi.cancel_action.CancelStatusResult
    alias of rauc_hawkbit.ddi.cancel_action.CancelStatusResultFinished

```

rauc_hawkbit.ddi.client module

```

exception rauc_hawkbit.ddi.client.APIError
    Bases: Exception

```

```

class rauc_hawkbit.ddi.client.ConfigStatusExecution
    Bases: enum.Enum

```

An enumeration.

```

canceled = 3
closed = 1
proceeding = 2
rejected = 5
resumed = 6
scheduled = 4

```

```

rauc_hawkbit.ddi.client.ConfigStatusResult
    alias of rauc_hawkbit.ddi.client.ConfigStatusResultFinished

```

```

class rauc_hawkbit.ddi.client.DDIClient (session, host, ssl, auth_token, tenant_id, controller_id, timeout=10)

```

Bases: object

Base Direct Device Integration API client providing GET, POST and PUT helpers as well as access to next level API resources.

```

build_api_url (api_path)
    Build the actual API URL.

```

Parameters `api_path` (*str*) – REST API path

Returns Expanded API URL with protocol (http/https) and host prepended

```

cancelAction

```

```

check_http_status (resp)
    Log API error message.

```

```

configData (status_execution, status_result, action_id="", status_details=(), **kwdata)
    Provide meta information during device registration at the update server.

```

See [#_put_tenant_controller_v1_targetid_configdata](http://sp.apps.bosch-iot-cloud.com/documentation/rest-api/rootcontroller-api-guide.html#_put_tenant_controller_v1_targetid_configdata) # noqa

Parameters

- **status_execution** (*ConfigStatusExecution*) – status of the action execution
- **status_result** (*status_result*) – result of the action execution

Keyword Arguments

- **action_id** (*str*) – Id of the action, not mandator for configData
- **status_details** ((*tuple*, *list*)) – List of details to provide
- **other** – passed as custom configuration data (key/value)

deploymentBase

error_responses = {400: 'Bad Request - e.g. invalid parameters', 401: 'The request r

get_binary (*url*, *dl_location*, *mime*='application/octet-stream', *timeout*=3600)

Actual download method with checksum checking.

Parameters

- **url** (*str*) – URL of item to download
- **dl_location** (*str*) – storage path for downloaded artifact

Keyword Arguments

- **mime** – mimetype of content to retrieve (default: 'application/octet-stream')
- **timeout** – download timeout (default: 3600)

Returns MD5 hash of downloaded content

get_binary_resource (*api_path*, *dl_location*, *mime*='application/octet-stream', *timeout*=3600,
***kwargs*)

Helper method for binary HTTP GET API requests.

Triggers download of the retrieved content to *dl_location*.

Parameters

- **api_path** (*str*) – REST API path
- **dl_location** (*str*) – storage path for downloaded artifact

Keyword Arguments

- **mime** – mimetype of content to retrieve (default: 'application/octet-stream')
- **kwargs** – Other keyword args used for replacing items in the API path

Returns MD5 hash of downloaded content

get_resource (*api_path*, *query_params*={}, ***kwargs*)

Helper method for HTTP GET API requests.

Parameters **api_path** (*str*) – REST API path

Keyword Arguments

- **query_params** – Query parameters to add to the API URL
- **kwargs** – Other keyword args used for replacing items in the API path

Returns Response JSON data

post_resource (*api_path*, *data*, ***kwargs*)

Helper method for HTTP POST API requests.

Parameters

- **api_path** (*str*) – REST API path
- **data** – JSON data for POST request

Keyword Arguments **kwargs** – keyword args used for replacing items in the API path

put_resource (*api_path*, *data*, ***kwargs*)
Helper method for HTTP PUT API requests.

Parameters

- **api_path** (*str*) – REST API path
- **data** – JSON data for POST request

Keyword Arguments **kwargs** – keyword args used for replacing items in the API path

softwaremodules

rauc_hawkbit.ddi.deployment_base module

class rauc_hawkbit.ddi.deployment_base.**DeploymentBase** (*ddi*)
Bases: object

Represents `/{{tenant}}/controller/v1/{{targetid}}/deploymentBase` in HawkBit's DDI API.

class rauc_hawkbit.ddi.deployment_base.**DeploymentBaseAction** (*ddi*, *action_id*)
Bases: object

Represents `/{{tenant}}/controller/v1/{{targetid}}/deploymentBase/{{actionId}}` in HawkBit's DDI API. See http://sp.apps.bosch-iot-cloud.com/documentation/rest-api/rootcontroller-api-guide.html#_get_tenant_controller_v1_targetid_deploymentbase_actionid # noqa

feedback (*status_execution*, *status_result*, *status_details=()*, ***kwstatus_result_progress*)
See http://sp.apps.bosch-iot-cloud.com/documentation/rest-api/rootcontroller-api-guide.html#_post_tenant_controller_v1_targetid_deploymentbase_actionid_feedback # noqa

class rauc_hawkbit.ddi.deployment_base.**DeploymentStatusExecution**
Bases: enum.Enum

An enumeration.

canceled = 3

closed = 1

proceeding = 2

rejected = 5

resumed = 6

scheduled = 4

rauc_hawkbit.ddi.deployment_base.**DeploymentStatusResult**
alias of rauc_hawkbit.ddi.deployment_base.**DeploymentStatusResultFinished**

class rauc_hawkbit.ddi.deployment_base.**DeploymentUpdate**
Bases: enum.Enum

An enumeration.

attempt = 2

forced = 3

skip = 1

rauc_hawkbit.ddi.softwaremodules module

class rauc_hawkbit.ddi.softwaremodules.**Artifacts** (*ddi, software_module_id*)

Bases: object

Represents `/{{tenant}}/controller/v1/{{targetid}}/softwaremodules/{{softwareModuleId}}/artifacts` # noqa in HawkBit's DDI API.

class rauc_hawkbit.ddi.softwaremodules.**FileName** (*ddi, software_module_id, file_name*)

Bases: object

Represents `/{{tenant}}/controller/v1/{{targetid}}/softwaremodules/{{softwareModuleId}}/artifacts/{{fileName}}` # noqa in HawkBit's DDI API.

MD5SUM (*md5_dl_location*)

See http://sp.apps.bosch-iot-cloud.com/documentation/rest-api/rootcontroller-api-guide.html#_get_tenant_controller_v1_targetid_softwaremodules_softwaremoduleid_artifacts_filename_md5sum # noqa

class rauc_hawkbit.ddi.softwaremodules.**SoftwareModule** (*ddi, software_module_id*)

Bases: object

Represents `/{{tenant}}/controller/v1/{{targetid}}/softwaremodules/{{softwareModuleId}}` # noqa in HawkBit's DDI API.

artifacts

class rauc_hawkbit.ddi.softwaremodules.**SoftwareModules** (*ddi*)

Bases: object

Represents `/{{tenant}}/controller/v1/{{targetid}}/softwaremodules` in HawkBit's DDI API.

3.1.2 Submodules

3.1.3 rauc_hawkbit.dbus_client module

3.1.4 rauc_hawkbit.rauc_dbus_ddi_client module

CHAPTER 4

Indices and Tables

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

r

- `rauc_hawkbit`, [7](#)
- `rauc_hawkbit.ddi`, [7](#)
- `rauc_hawkbit.ddi.cancel_action`, [7](#)
- `rauc_hawkbit.ddi.client`, [8](#)
- `rauc_hawkbit.ddi.deployment_base`, [10](#)
- `rauc_hawkbit.ddi.softwaremodules`, [11](#)

A

Action (class in `rauc_hawkbite.ddi.cancel_action`), 7

APIError, 8

Artifacts (class in `rauc_hawkbite.ddi.softwaremodules`), 11

artifacts (`rauc_hawkbite.ddi.softwaremodules.SoftwareModule` attribute), 11

attempt (`rauc_hawkbite.ddi.deployment_base.DeploymentUpdate` attribute), 10

B

build_api_url() (`rauc_hawkbite.ddi.client.DDIClient` method), 8

C

CancelAction (class in `rauc_hawkbite.ddi.cancel_action`), 7

cancelAction (`rauc_hawkbite.ddi.client.DDIClient` attribute), 8

canceled (`rauc_hawkbite.ddi.cancel_action.CancelStatusExecution` attribute), 7

canceled (`rauc_hawkbite.ddi.client.ConfigStatusExecution` attribute), 8

canceled (`rauc_hawkbite.ddi.deployment_base.DeploymentStatusExecution` attribute), 10

CancelStatusExecution (class in `rauc_hawkbite.ddi.cancel_action`), 7

CancelStatusResult (in module `rauc_hawkbite.ddi.cancel_action`), 8

check_http_status() (`rauc_hawkbite.ddi.client.DDIClient` method), 8

closed (`rauc_hawkbite.ddi.cancel_action.CancelStatusExecution` attribute), 7

closed (`rauc_hawkbite.ddi.client.ConfigStatusExecution` attribute), 8

closed (`rauc_hawkbite.ddi.deployment_base.DeploymentStatusExecution` attribute), 10

configData() (`rauc_hawkbite.ddi.client.DDIClient` method), 8

ConfigStatusExecution (class in `rauc_hawkbite.ddi.client`), 8

ConfigStatusResult (in module `rauc_hawkbite.ddi.client`), 8

D

DDIClient (class in `rauc_hawkbite.ddi.client`), 8

DeploymentBase (class in `rauc_hawkbite.ddi.deployment_base`), 10

deploymentBase (`rauc_hawkbite.ddi.client.DDIClient` attribute), 9

DeploymentBaseAction (class in `rauc_hawkbite.ddi.deployment_base`), 10

DeploymentStatusExecution (class in `rauc_hawkbite.ddi.deployment_base`), 10

DeploymentStatusResult (in module `rauc_hawkbite.ddi.deployment_base`), 10

DeploymentUpdate (class in `rauc_hawkbite.ddi.deployment_base`), 10

E

error_responses (`rauc_hawkbite.ddi.client.DDIClient` attribute), 9

F

feedback() (`rauc_hawkbite.ddi.cancel_action.Action` method), 7

feedback() (`rauc_hawkbite.ddi.deployment_base.DeploymentBaseAction` method), 10

FileName (class in `rauc_hawkbite.ddi.softwaremodules`), 11

failed (`rauc_hawkbite.ddi.deployment_base.DeploymentUpdate` attribute), 10

G

get_binary_resource() (`rauc_hawkbite.ddi.client.DDIClient` method), 9

get_binary_resource() (`rauc_hawkbite.ddi.client.DDIClient` method), 9

`get_resource()` (*rauc_hawkbit.ddi.client.DDIClient* SoftwareModules (class in *method*), 9 *rauc_hawkbit.ddi.softwaremodules*), 11
`softwaremodules` (*rauc_hawkbit.ddi.client.DDIClient* attribute), 10

M

`MD5SUM()` (*rauc_hawkbit.ddi.softwaremodules.FileName* method), 11

P

`post_resource()` (*rauc_hawkbit.ddi.client.DDIClient* method), 9
`proceeding` (*rauc_hawkbit.ddi.cancel_action.CancelStatusExecution* attribute), 7
`proceeding` (*rauc_hawkbit.ddi.client.ConfigStatusExecution* attribute), 8
`proceeding` (*rauc_hawkbit.ddi.deployment_base.DeploymentStatusExecution* attribute), 10
`put_resource()` (*rauc_hawkbit.ddi.client.DDIClient* method), 9
Python Enhancement Proposals
PEP 8, 3

R

`rauc_hawkbit` (module), 7
`rauc_hawkbit.ddi` (module), 7
`rauc_hawkbit.ddi.cancel_action` (module), 7
`rauc_hawkbit.ddi.client` (module), 8
`rauc_hawkbit.ddi.deployment_base` (module), 10
`rauc_hawkbit.ddi.softwaremodules` (module), 11
`rejected` (*rauc_hawkbit.ddi.cancel_action.CancelStatusExecution* attribute), 7
`rejected` (*rauc_hawkbit.ddi.client.ConfigStatusExecution* attribute), 8
`rejected` (*rauc_hawkbit.ddi.deployment_base.DeploymentStatusExecution* attribute), 10
`resumed` (*rauc_hawkbit.ddi.cancel_action.CancelStatusExecution* attribute), 8
`resumed` (*rauc_hawkbit.ddi.client.ConfigStatusExecution* attribute), 8
`resumed` (*rauc_hawkbit.ddi.deployment_base.DeploymentStatusExecution* attribute), 10

S

`scheduled` (*rauc_hawkbit.ddi.cancel_action.CancelStatusExecution* attribute), 8
`scheduled` (*rauc_hawkbit.ddi.client.ConfigStatusExecution* attribute), 8
`scheduled` (*rauc_hawkbit.ddi.deployment_base.DeploymentStatusExecution* attribute), 10
`skip` (*rauc_hawkbit.ddi.deployment_base.DeploymentUpdate* attribute), 10
SoftwareModule (class in *rauc_hawkbit.ddi.softwaremodules*), 11