
payload Documentation

Release

Paul Belanger

January 31, 2015

Contents

1	API Complete Reference	3
1.1	Payload API Reference	3
2	Developers Docs	9
2.1	Developer Guide	9
3	Man pages	13
3.1	payload-api	13
3.2	payload-manage	14
4	Modules	15
4.1	payload	15
5	Indices and tables	31
	HTTP Routing Table	33
	Python Module Index	35

Contents:

API Complete Reference

1.1 Payload API Reference

1.1.1 API Operations

Agents API

List agents

GET /v1/agents

Parameters

Response

```
[
  {
    "uuid": "5f91c19d-84f0-4122-bc7f-dcdda6704953",
    "created_at": "2011-04-22T13:33:48Z",
    "project_id": "793491dd5fa8477eb2d6a820193a183b",
    "user_id": "02d99a62af974b26b510c3564ba84644",
    "updated_at": "2011-04-22T13:33:48Z",
  }
]
```

Get a single agent

GET /v1/agents/:uuid

Parameters

Response

```
{
  "uuid": "5f91c19d-84f0-4122-bc7f-dcdda6704953",
  "created_at": "2011-04-22T13:33:48Z",
  "project_id": "793491dd5fa8477eb2d6a820193a183b",
  "user_id": "02d99a62af974b26b510c3564ba84644",
  "updated_at": "2011-04-22T13:33:48Z",
}
```

Create an agent

POST /v1/agents

Input

Response

Edit an agent

PUT /v1/agents/:uuid

Parameters

Input

Response

Delete an agent

DELETE /v1/agents/:uuid

Parameters

Response

Queues API

Queue Members API

List queue members

GET /v1/queues/:queue_id/members

Parameters

Response

```
[
  {
    "created_at": "2011-04-22T13:33:48Z",
    "number": "1001@example.org",
    "paused_at": None,
    "queue_id": "cc096e0b-0c96-4b8b-b812-ef456f361ee3",
    "uuid": "b1775ea3f7fb451a97ef669d1ccf8e64",
  }
]
```

Get queue member

```
GET /v1/queues/:queue_id/members/:agent_id
```

Parameters**Response**

```
{
  "created_at": "2011-04-22T13:33:48Z",
  "number": "1001@example.org",
  "paused_at": None,
  "queue_id": "cc096e0b-0c96-4b8b-b812-ef456f361ee3",
  "uuid": "b1775ea3f7fb451a97ef669d1ccf8e64",
}
```

Queue Stats API**List Stats from queue**

```
GET /v1/queues/:queue_id/stats
```

Parameters**Response**

```
[
  {
    "callers": "32",
  }
]
```

List queues

```
GET /v1/queues
```

Parameters

Response

```
[
  {
    "uuid": "8090f8b0-4115-11e3-aa6e-0800200c9a66",
    "created_at": "2011-04-22T13:33:48Z",
    "description": "24/7 Technical support",
    "disabled": False,
    "name": "support",
    "project_id": "793491dd5fa8477eb2d6a820193a183b",
    "user_id": "02d99a62af974b26b510c3564ba84644",
    "updated_at": "2011-04-22T13:33:48Z",
  }
]
```

Get a single queue

```
GET /v1/queues/:uuid
```

Parameters

Response

```
{
  "uuid": "8090f8b0-4115-11e3-aa6e-0800200c9a66",
  "created_at": "2011-04-22T13:33:48Z",
  "description": "24/7 Technical support",
  "disabled": False,
  "name": "support",
  "project_id": "793491dd5fa8477eb2d6a820193a183b",
  "user_id": "02d99a62af974b26b510c3564ba84644",
  "updated_at": "2011-04-22T13:33:48Z",
}
```

Create a queue

```
POST /v1/queues
```

Input

Response

Edit a queue

```
PUT /v1/queues/:uuid
```

Parameters

Input

Response

Delete a queue

```
DELETE /v1/queues/:uuid
```

Parameters

Response

Developers Docs

2.1 Developer Guide

2.1.1 Programming HowTos and Tutorials

Setting Up a Development Environment

Note: This document originates from the OpenStack Nova project. Since we also use the same toolset for testing, we can also use some of their documentaion. We have obviously made changes that only affect our project but we credit the OpenStack project for the original ¹.

This page describes how to setup a working Python development environment that can be used in developing payload on Ubuntu. These instructions assume you're already familiar with git.

Following these instructions will allow you to run the payload unit tests.

Virtual environments

payload development uses a set of shell scripts from OpenStack's DevStack. Virtual enviroments with venv are also available with the source code.

Linux Systems

Note: This section is tested for payload on Ubuntu (12.04-64) distribution. Feel free to add notes and change according to your experiences or operating system.

Install the prerequisite packages.

On Ubuntu Precise (12.04):

¹ See <http://docs.openstack.org/developer/nova/devref/development.environment.html>

```
sudo apt-get install python-dev python-pip git-core redis-server
```

Getting the code

Grab the code from GitHub:

```
git clone https://github.com/kickstandproject/payload.git
cd payload
```

Installing and using the virtualenv

To install the virtual environment you simply run the following:

```
python tools/install_venv.py
```

This will install all of the Python packages listed in the `requirements.txt` file into your virtualenv. There will also be some additional packages (pip, distribute, greenlet) that are installed by the `tools/install_venv.py` file into the virtualenv.

If all goes well, you should get a message something like this:

```
payload development environment setup is complete.
```

```
payload development uses virtualenv to track and manage Python dependencies
while in development and testing.
```

To activate the payload virtualenv for the extent of your current shell session you can run:

```
$ source .venv/bin/activate
```

Or, if you prefer, you can run commands in the virtualenv on a case by case basis by running:

```
$ tools/with_venv.sh <your command>
```

Also, make test will automatically use the virtualenv.

Running unit tests

The unit tests will run by default inside tox env in the `.tox` directory. Run the unit tests by doing:

```
tox
```

See [Unit Tests](#) for more details.

Contributing Your Work

Once your work is complete you may wish to contribute it to the project. Refer to [HowToContribute](#) for information. Nova uses the Gerrit code review system. For information on how to submit your branch to Gerrit, see [GerritWorkflow](#).

Unit Tests

Note: This document originates from the OpenStack Nova project. Since we also use the same toolset for testing, we can also use some of their documentation. We have obviously made changes that only affect our project but we credit the OpenStack project for the original ².

Payload contains a suite of unit tests, in the `payload/tests` directory.

Any proposed code change will be automatically rejected by the Kickstand Project Jenkins server if the change causes unit test failures.

Running the tests

Run the unit tests by doing:

```
tox
```

This script is a wrapper around the `nose` testrunner and the `pep8` checker.

² See http://docs.openstack.org/developer/nova/devref/unit_tests.html

Man pages

3.1 payload-api

3.1.1 Server for the payload APIs

Author paul.belanger@polybeacon.com

Date January 31, 2015

Copyright PolyBeacon, Inc

Version

Manual section 1

SYNOPSIS

payload-api [options]

DESCRIPTION

payload-api is a server daemon that serves the payload APIs

OPTIONS

General options

FILES

- /etc/payload/payload.conf

SEE ALSO

- [payload](#)

BUGS

- payload is sourced in Github so you can view current bugs at [payload](#)

3.2 payload-manage

3.2.1 Control and manage payload

Author paul.belanger@polybeacon.com

Date January 31, 2015

Copyright PolyBeacon, Inc

Version

Manual section 1

SYNOPSIS

```
payload-manage <category> <action> [<args>]
```

DESCRIPTION

payload-manage is a CLI tool to control payload

OPTIONS

General options

payload DB

```
payload-manage db sync
```

Sync the database up to the most recent version.

```
payload-manage db version
```

Print the current database version.

FILES

- /etc/payload/payload.conf

SEE ALSO

- [payload](#)

BUGS

- payload is sourced in Github so you can view current bugs at [payload](#)

Modules

4.1 payload

4.1.1 payload Package

payload Package

test Module

```
class payload.test.Database (db_session, db_migrate, sql_connection, sqlite_db, sqlite_clean_db)
```

Bases: fixtures.fixture.Fixture

setUp ()

```
class payload.test.Redis
```

Bases: fixtures.fixture.Fixture

setUp ()

```
class payload.test.TestCase (*args, **kwargs)
```

Bases: testtools.testcase.TestCase

Test case base class for all unit tests.

config (**kw)

Override some configuration values.

The keyword arguments are the names of configuration options to override and their values.

If a group argument is supplied, the overrides are applied to the specified configuration option group.

All overrides are automatically cleared at the end of the current test by the fixtures cleanup process.

path_get (project_file=None)

Get the absolute path to a file. Used for testing the API.

Parameters **project_file** – File whose path to return. Default: None.

Returns path to the specified file, or path to project root.

setUp ()

Run before each method to initialize test environment.

version Module

Subpackages

api Package

api Package

app Module

class `payload.api.app.VersionSelectorApplication`

Bases: `object`

`payload.api.app.app_factory(global_config, **local_conf)`

`payload.api.app.build_server()`

`payload.api.app.get_handler_cls()`

`payload.api.app.get_pecan_config()`

`payload.api.app.get_server_cls(host)`

Return an appropriate WSGI server class base on provided host

Parameters `host` – The listen host for the ceilometer API server.

`payload.api.app.load_app()`

`payload.api.app.setup_app(pecan_config=None)`

config Module

hooks Module

class `payload.api.hooks.CacheHook`

Bases: `pecan.hooks.PecanHook`

before (*state*)

class `payload.api.hooks.DBHook`

Bases: `pecan.hooks.PecanHook`

before (*state*)

middleware Module Middleware to replace the plain text message body of an error response with one formatted so the client can parse it.

Based on `pecan.middleware.errordocument`

class `payload.api.middleware.ParsableErrorMiddleware(app)`

Bases: `object`

Replace error body with something the client can parse.

Subpackages

controllers Package

controllers Package

root Module

```
class payload.api.controllers.root.RootController
```

Bases: object

v1 = <payload.api.controllers.v1.controller.Controller object at 0x7ffe2e2da690>

Subpackages**v1 Package****v1 Package****agent Module**

```
class payload.api.controllers.v1.agent.Agent (**kwargs)
```

Bases: payload.api.controllers.v1.base.APIBase

API representation of an agent.

created_at

Complex type attribute definition.

Example:

```
class MyComplexType(wsme.types.Base):
    optionalvalue = int
    mandatoryvalue = wsattr(int, mandatory=True)
    named_value = wsattr(int, name='named.value')
```

After inspection, the non-wsattr attributes will be replaced, and the above class will be equivalent to:

```
class MyComplexType(wsme.types.Base):
    optionalvalue = wsattr(int)
    mandatoryvalue = wsattr(int, mandatory=True)
```

project_id

Complex type attribute definition.

Example:

```
class MyComplexType(wsme.types.Base):
    optionalvalue = int
    mandatoryvalue = wsattr(int, mandatory=True)
    named_value = wsattr(int, name='named.value')
```

After inspection, the non-wsattr attributes will be replaced, and the above class will be equivalent to:

```
class MyComplexType(wsme.types.Base):
    optionalvalue = wsattr(int)
    mandatoryvalue = wsattr(int, mandatory=True)
```

updated_at

Complex type attribute definition.

Example:

```
class MyComplexType(wsme.types.Base):
    optionalvalue = int
    mandatoryvalue = wsattr(int, mandatory=True)
    named_value = wsattr(int, name='named.value')
```

After inspection, the non-wsattr attributes will be replaced, and the above class will be equivalent to:

```
class MyComplexType(wsme.types.Base):
    optionalvalue = wsattr(int)
    mandatoryvalue = wsattr(int, mandatory=True)
```

user_id

Complex type attribute definition.

Example:

```
class MyComplexType(wsme.types.Base):
    optionalvalue = int
    mandatoryvalue = wsattr(int, mandatory=True)
    named_value = wsattr(int, name='named.value')
```

After inspection, the non-wsattr attributes will be replaced, and the above class will be equivalent to:

```
class MyComplexType(wsme.types.Base):
    optionalvalue = wsattr(int)
    mandatoryvalue = wsattr(int, mandatory=True)
```

uuid

Complex type attribute definition.

Example:

```
class MyComplexType(wsme.types.Base):
    optionalvalue = int
    mandatoryvalue = wsattr(int, mandatory=True)
    named_value = wsattr(int, name='named.value')
```

After inspection, the non-wsattr attributes will be replaced, and the above class will be equivalent to:

```
class MyComplexType(wsme.types.Base):
    optionalvalue = wsattr(int)
    mandatoryvalue = wsattr(int, mandatory=True)
```

```
class payload.api.controllers.v1.agent.AgentsController
```

Bases: `pecan.rest.RestController`

REST Controller for Agents.

delete (*args, **kwargs)
Delete an agent.

get_all (*args, **kwargs)
Retrieve a list of agents.

get_one (*args, **kwargs)
Retrieve information about the given agent.

post (*args, **kwargs)
Create a new agent.

put (*args, **kwargs)
Update an existing agent.

base Module

```
class payload.api.controllers.v1.base.APIBase (**kw)
    Bases: wsme.types.Base
```

```
as_dict()
```

```
created_at
```

Complex type attribute definition.

Example:

```
class MyComplexType (wsme.types.Base):
    optionalvalue = int
    mandatoryvalue = wsattr(int, mandatory=True)
    named_value = wsattr(int, name='named.value')
```

After inspection, the non-wsattr attributes will be replaced, and the above class will be equivalent to:

```
class MyComplexType (wsme.types.Base):
    optionalvalue = wsattr(int)
    mandatoryvalue = wsattr(int, mandatory=True)
```

```
updated_at
```

Complex type attribute definition.

Example:

```
class MyComplexType (wsme.types.Base):
    optionalvalue = int
    mandatoryvalue = wsattr(int, mandatory=True)
    named_value = wsattr(int, name='named.value')
```

After inspection, the non-wsattr attributes will be replaced, and the above class will be equivalent to:

```
class MyComplexType (wsme.types.Base):
    optionalvalue = wsattr(int)
    mandatoryvalue = wsattr(int, mandatory=True)
```

controller Module

```
class payload.api.controllers.v1.controller.Controller
    Bases: object
```

```
agents = <payload.api.controllers.v1.agent.AgentsController object at 0x7ffe2e2da4d0>
```

```
queues = <payload.api.controllers.v1.queue.QueuesController object at 0x7ffe2e2da610>
```

```
schemas = <payload.api.controllers.v1.schema.SchemasController object at 0x7ffe2e2da650>
```

schema Module

```
class payload.api.controllers.v1.schema.SchemasController
    Bases: pecan.rest.RestController
```

REST Controller for Schemas.

```
agent (*args, **kwargs)
```

Retrieve schema for an agent.

```
queue (*args, **kwargs)
```

Retrieve schema for a queue.

queue_caller (*args, **kwargs)
Retrieve schema for a queue caller.

Subpackages

queue Package

queue Package

class payload.api.controllers.v1.queue.**Queue** (**kwargs)
Bases: payload.api.controllers.v1.base.APIBase

API representation of a queue.

created_at
Complex type attribute definition.

Example:

```
class MyComplexType(wsme.types.Base):
    optionalvalue = int
    mandatoryvalue = wsattr(int, mandatory=True)
    named_value = wsattr(int, name='named.value')
```

After inspection, the non-wsattr attributes will be replaced, and the above class will be equivalent to:

```
class MyComplexType(wsme.types.Base):
    optionalvalue = wsattr(int)
    mandatoryvalue = wsattr(int, mandatory=True)
```

description
Complex type attribute definition.

Example:

```
class MyComplexType(wsme.types.Base):
    optionalvalue = int
    mandatoryvalue = wsattr(int, mandatory=True)
    named_value = wsattr(int, name='named.value')
```

After inspection, the non-wsattr attributes will be replaced, and the above class will be equivalent to:

```
class MyComplexType(wsme.types.Base):
    optionalvalue = wsattr(int)
    mandatoryvalue = wsattr(int, mandatory=True)
```

disabled
Complex type attribute definition.

Example:

```
class MyComplexType(wsme.types.Base):
    optionalvalue = int
    mandatoryvalue = wsattr(int, mandatory=True)
    named_value = wsattr(int, name='named.value')
```

After inspection, the non-wsattr attributes will be replaced, and the above class will be equivalent to:

```
class MyComplexType(wsme.types.Base):
    optionalvalue = wsattr(int)
    mandatoryvalue = wsattr(int, mandatory=True)
```

name

Complex type attribute definition.

Example:

```
class MyComplexType(wsme.types.Base):
    optionalvalue = int
    mandatoryvalue = wsattr(int, mandatory=True)
    named_value = wsattr(int, name='named.value')
```

After inspection, the non-wsattr attributes will be replaced, and the above class will be equivalent to:

```
class MyComplexType(wsme.types.Base):
    optionalvalue = wsattr(int)
    mandatoryvalue = wsattr(int, mandatory=True)
```

project_id

Complex type attribute definition.

Example:

```
class MyComplexType(wsme.types.Base):
    optionalvalue = int
    mandatoryvalue = wsattr(int, mandatory=True)
    named_value = wsattr(int, name='named.value')
```

After inspection, the non-wsattr attributes will be replaced, and the above class will be equivalent to:

```
class MyComplexType(wsme.types.Base):
    optionalvalue = wsattr(int)
    mandatoryvalue = wsattr(int, mandatory=True)
```

updated_at

Complex type attribute definition.

Example:

```
class MyComplexType(wsme.types.Base):
    optionalvalue = int
    mandatoryvalue = wsattr(int, mandatory=True)
    named_value = wsattr(int, name='named.value')
```

After inspection, the non-wsattr attributes will be replaced, and the above class will be equivalent to:

```
class MyComplexType(wsme.types.Base):
    optionalvalue = wsattr(int)
    mandatoryvalue = wsattr(int, mandatory=True)
```

user_id

Complex type attribute definition.

Example:

```
class MyComplexType(wsme.types.Base):
    optionalvalue = int
    mandatoryvalue = wsattr(int, mandatory=True)
    named_value = wsattr(int, name='named.value')
```

After inspection, the non-wsattr attributes will be replaced, and the above class will be equivalent to:

```
class MyComplexType(wsme.types.Base):
    optionalvalue = wsattr(int)
    mandatoryvalue = wsattr(int, mandatory=True)
```

uuid

Complex type attribute definition.

Example:

```
class MyComplexType(wsme.types.Base):
    optionalvalue = int
    mandatoryvalue = wsattr(int, mandatory=True)
    named_value = wsattr(int, name='named.value')
```

After inspection, the non-wsattr attributes will be replaced, and the above class will be equivalent to:

```
class MyComplexType(wsme.types.Base):
    optionalvalue = wsattr(int)
    mandatoryvalue = wsattr(int, mandatory=True)
```

class payload.api.controllers.v1.queue.QueuesController

Bases: pecan.rest.RestController

REST Controller for queues.

callers = <payload.api.controllers.v1.queue.caller.QueueCallersController object at 0x7ffe2e352f90>

delete (*args, **kwargs)
Delete a queue.

get_all (*args, **kwargs)
Retrieve a list of queues.

get_one (*args, **kwargs)
Retrieve information about the given queue.

members = <payload.api.controllers.v1.queue.member.QueueMembersController object at 0x7ffe2e3528d0>

post (*args, **kwargs)
Create a new queue.

put (*args, **kwargs)

member Module

class payload.api.controllers.v1.queue.member.QueueMember(**kwargs)

Bases: object

API representation of a queue member.

created_at

Complex type attribute definition.

Example:

```
class MyComplexType(wsme.types.Base):
    optionalvalue = int
    mandatoryvalue = wsattr(int, mandatory=True)
    named_value = wsattr(int, name='named.value')
```

After inspection, the non-wsattr attributes will be replaced, and the above class will be equivalent to:

```
class MyComplexType(wsme.types.Base):
    optionalvalue = wsattr(int)
    mandatoryvalue = wsattr(int, mandatory=True)
```

number

Complex type attribute definition.

Example:

```
class MyComplexType(wsme.types.Base):
    optionalvalue = int
    mandatoryvalue = wsattr(int, mandatory=True)
    named_value = wsattr(int, name='named.value')
```

After inspection, the non-wsattr attributes will be replaced, and the above class will be equivalent to:

```
class MyComplexType(wsme.types.Base):
    optionalvalue = wsattr(int)
    mandatoryvalue = wsattr(int, mandatory=True)
```

paused

Complex type attribute definition.

Example:

```
class MyComplexType(wsme.types.Base):
    optionalvalue = int
    mandatoryvalue = wsattr(int, mandatory=True)
    named_value = wsattr(int, name='named.value')
```

After inspection, the non-wsattr attributes will be replaced, and the above class will be equivalent to:

```
class MyComplexType(wsme.types.Base):
    optionalvalue = wsattr(int)
    mandatoryvalue = wsattr(int, mandatory=True)
```

paused_at

Complex type attribute definition.

Example:

```
class MyComplexType(wsme.types.Base):
    optionalvalue = int
    mandatoryvalue = wsattr(int, mandatory=True)
    named_value = wsattr(int, name='named.value')
```

After inspection, the non-wsattr attributes will be replaced, and the above class will be equivalent to:

```
class MyComplexType(wsme.types.Base):
    optionalvalue = wsattr(int)
    mandatoryvalue = wsattr(int, mandatory=True)
```

queue_id

Complex type attribute definition.

Example:

```
class MyComplexType(wsme.types.Base):
    optionalvalue = int
    mandatoryvalue = wsattr(int, mandatory=True)
    named_value = wsattr(int, name='named.value')
```

After inspection, the non-wsattr attributes will be replaced, and the above class will be equivalent to:

```
class MyComplexType(wsme.types.Base):
    optionalvalue = wsattr(int)
    mandatoryvalue = wsattr(int, mandatory=True)
```

status

Complex type attribute definition.

Example:

```
class MyComplexType(wsme.types.Base):
    optionalvalue = int
    mandatoryvalue = wsattr(int, mandatory=True)
    named_value = wsattr(int, name='named.value')
```

After inspection, the non-wsattr attributes will be replaced, and the above class will be equivalent to:

```
class MyComplexType(wsme.types.Base):
    optionalvalue = wsattr(int)
    mandatoryvalue = wsattr(int, mandatory=True)
```

status_at

Complex type attribute definition.

Example:

```
class MyComplexType(wsme.types.Base):
    optionalvalue = int
    mandatoryvalue = wsattr(int, mandatory=True)
    named_value = wsattr(int, name='named.value')
```

After inspection, the non-wsattr attributes will be replaced, and the above class will be equivalent to:

```
class MyComplexType(wsme.types.Base):
    optionalvalue = wsattr(int)
    mandatoryvalue = wsattr(int, mandatory=True)
```

uuid

Complex type attribute definition.

Example:

```
class MyComplexType(wsme.types.Base):
    optionalvalue = int
    mandatoryvalue = wsattr(int, mandatory=True)
    named_value = wsattr(int, name='named.value')
```

After inspection, the non-wsattr attributes will be replaced, and the above class will be equivalent to:

```
class MyComplexType(wsme.types.Base):
    optionalvalue = wsattr(int)
    mandatoryvalue = wsattr(int, mandatory=True)
```

```
class payload.api.controllers.v1.queue.member.QueueMembersController
    Bases: pecan.rest.RestController
```

REST Controller for queue members.

get_all (*args, **kwargs)

Retrieve a list of queue members.

get_one (*args, **kwargs)

Retrieve information about the given queue member.

caller Module

class payload.api.controllers.v1.queue.caller.**QueueCaller** (**kwargs)

Bases: object

API representation of a queue caller.

created_at

Complex type attribute definition.

Example:

```
class MyComplexType(wsme.types.Base):
    optionalvalue = int
    mandatoryvalue = wsattr(int, mandatory=True)
    named_value = wsattr(int, name='named.value')
```

After inspection, the non-wsattr attributes will be replaced, and the above class will be equivalent to:

```
class MyComplexType(wsme.types.Base):
    optionalvalue = wsattr(int)
    mandatoryvalue = wsattr(int, mandatory=True)
```

member_uuid

Complex type attribute definition.

Example:

```
class MyComplexType(wsme.types.Base):
    optionalvalue = int
    mandatoryvalue = wsattr(int, mandatory=True)
    named_value = wsattr(int, name='named.value')
```

After inspection, the non-wsattr attributes will be replaced, and the above class will be equivalent to:

```
class MyComplexType(wsme.types.Base):
    optionalvalue = wsattr(int)
    mandatoryvalue = wsattr(int, mandatory=True)
```

name

Complex type attribute definition.

Example:

```
class MyComplexType(wsme.types.Base):
    optionalvalue = int
    mandatoryvalue = wsattr(int, mandatory=True)
    named_value = wsattr(int, name='named.value')
```

After inspection, the non-wsattr attributes will be replaced, and the above class will be equivalent to:

```
class MyComplexType(wsme.types.Base):
    optionalvalue = wsattr(int)
    mandatoryvalue = wsattr(int, mandatory=True)
```

number

Complex type attribute definition.

Example:

```
class MyComplexType(wsme.types.Base):
    optionalvalue = int
    mandatoryvalue = wsattr(int, mandatory=True)
    named_value = wsattr(int, name='named.value')
```

After inspection, the non-wsattr attributes will be replaced, and the above class will be equivalent to:

```
class MyComplexType(wsme.types.Base):
    optionalvalue = wsattr(int)
    mandatoryvalue = wsattr(int, mandatory=True)
```

position

Complex type attribute definition.

Example:

```
class MyComplexType(wsme.types.Base):
    optionalvalue = int
    mandatoryvalue = wsattr(int, mandatory=True)
    named_value = wsattr(int, name='named.value')
```

After inspection, the non-wsattr attributes will be replaced, and the above class will be equivalent to:

```
class MyComplexType(wsme.types.Base):
    optionalvalue = wsattr(int)
    mandatoryvalue = wsattr(int, mandatory=True)
```

queue_id

Complex type attribute definition.

Example:

```
class MyComplexType(wsme.types.Base):
    optionalvalue = int
    mandatoryvalue = wsattr(int, mandatory=True)
    named_value = wsattr(int, name='named.value')
```

After inspection, the non-wsattr attributes will be replaced, and the above class will be equivalent to:

```
class MyComplexType(wsme.types.Base):
    optionalvalue = wsattr(int)
    mandatoryvalue = wsattr(int, mandatory=True)
```

status

Complex type attribute definition.

Example:

```
class MyComplexType(wsme.types.Base):
    optionalvalue = int
    mandatoryvalue = wsattr(int, mandatory=True)
    named_value = wsattr(int, name='named.value')
```

After inspection, the non-wsattr attributes will be replaced, and the above class will be equivalent to:

```
class MyComplexType(wsme.types.Base):
    optionalvalue = wsattr(int)
    mandatoryvalue = wsattr(int, mandatory=True)
```

status_at

Complex type attribute definition.

Example:

```
class MyComplexType(wsme.types.Base):
    optionalvalue = int
    mandatoryvalue = wsattr(int, mandatory=True)
    named_value = wsattr(int, name='named.value')
```

After inspection, the non-wsattr attributes will be replaced, and the above class will be equivalent to:

```
class MyComplexType(wsme.types.Base):
    optionalvalue = wsattr(int)
    mandatoryvalue = wsattr(int, mandatory=True)
```

uuid

Complex type attribute definition.

Example:

```
class MyComplexType(wsme.types.Base):
    optionalvalue = int
    mandatoryvalue = wsattr(int, mandatory=True)
    named_value = wsattr(int, name='named.value')
```

After inspection, the non-wsattr attributes will be replaced, and the above class will be equivalent to:

```
class MyComplexType(wsme.types.Base):
    optionalvalue = wsattr(int)
    mandatoryvalue = wsattr(int, mandatory=True)
```

```
class payload.api.controllers.v1.queue.caller.QueueCallersController
    Bases: pecan.rest.RestController
```

REST Controller for queue callers.

get_all (*args, **kwargs)

List callers from the specified queue.

GET /queues/:queue_uuid/callers

Example request:

```
GET /queues/cc096e0b-0c96-4b8b-b812-ef456f361ee3/callers
```

Example response:

```
[
  {
    "created_at": "2014-12-12T02:05:14Z",
    "name": "Paul Belanger",
    "number": "6135551234",
    "position": 0,
    "queue_id": "cc096e0b-0c96-4b8b-b812-ef456f361ee3",
    "uuid": "e5814fee-6e8a-4771-8edd-ea413eff57f1",
  },
  {
    "created_at": "2014-12-12T02:07:05Z",
    "name": "Leif Madsen",
    "number": "9055555678",
    "position": 1,
    "queue_id": "cc096e0b-0c96-4b8b-b812-ef456f361ee3",
    "uuid": "4b4fa110-be14-45b7-a998-2219ab8bee6f",
  }
]
```

```
}
]
```

get_one (*args, **kwargs)

Get a single caller from the specified queue.

GET /queues/:queue_uuid/callers/:caller_uuid

Example request:

```
GET /queues/cc096e0b-0c96-4b8b-b812-ef456f361ee3/callers/e5814fee-6e8a-4771-8edd-ea413ef
```

Example response:

```
{
  "created_at": "2014-12-12T02:05:14Z",
  "name": "Paul Belanger",
  "number": "6135551234",
  "position": 0,
  "queue_id": "cc096e0b-0c96-4b8b-b812-ef456f361ee3",
  "uuid": "e5814fee-6e8a-4771-8edd-ea413eff57f1",
}
```

cmd Package

cmd Package

api Module Payload Service API

```
payload.cmd.api.main()
```

manage Module CLI interface for payload management.

```
payload.cmd.manage.add_command_parsers(subparsers)
```

```
payload.cmd.manage.do_db_sync()
```

Place a database under migration control and upgrade, creating first if necessary.

```
payload.cmd.manage.do_db_version()
```

Print database's current migration level.

```
payload.cmd.manage.main()
```

common Package

common Package

exception Module Payload base exception handling.

```
exception payload.common.exception.AgentNotFound(message=None, **kwargs)
```

Bases: `payload.common.exception.NotFound`

message = 'Agent %(uuid)s could not be found'

```
exception payload.common.exception.NotFound(message=None, **kwargs)
```

Bases: `payload.common.exception.PayloadException`

code = 404

message = 'Resource could not be found'

exception `payload.common.exception.PayloadException` (*message=None, **kwargs*)

Bases: `exceptions.Exception`

Base Payload Exception

To correctly use this class, inherit from it and define a 'message' property. That message will get printf'd with the keyword arguments provided to the constructor.

code = 500

format_message ()

headers = {}

message = 'An unknown exception occurred.'

safe = False

exception `payload.common.exception.QueueCallerNotFound` (*message=None, **kwargs*)

Bases: `payload.common.exception.NotFound`

message = 'Queue caller %(uuid)s could not be found'

exception `payload.common.exception.QueueMemberDuplicated` (*message=None, **kwargs*)

Bases: `payload.common.exception.PayloadException`

message = 'Queue member %(uuid)s already exists.'

exception `payload.common.exception.QueueMemberNotFound` (*message=None, **kwargs*)

Bases: `payload.common.exception.NotFound`

message = 'Queue member %(uuid)s could not be found'

exception `payload.common.exception.QueueNotFound` (*message=None, **kwargs*)

Bases: `payload.common.exception.NotFound`

message = 'Queue %(uuid)s could not be found'

paths Module

`payload.common.paths.basedir_def (*args)`

Return an uninterpolated path relative to \$pybasedir.

`payload.common.paths.basedir_rel (*args)`

Return a path relative to \$pybasedir.

`payload.common.paths.bindir_def (*args)`

Return an uninterpolated path relative to \$bindir.

`payload.common.paths.bindir_rel (*args)`

Return a path relative to \$bindir.

`payload.common.paths.state_path_def (*args)`

Return an uninterpolated path relative to \$state_path.

`payload.common.paths.state_path_rel (*args)`

Return a path relative to \$state_path.

schema Module

```
class payload.common.schema.Schema (name, properties={})
    Bases: object

    raw ()
```

db Package

db Package

api Module

```
payload.db.api.create_agent (user_id, project_id)
    Create a new agent.

payload.db.api.create_queue (name, user_id, project_id, description='', disabled=False)
    Create a new queue.

payload.db.api.create_queue_member (agent_uuid, queue_uuid)
    Create a new queue member.

payload.db.api.delete_agent (uuid)
    Delete an agent.

payload.db.api.delete_queue (uuid)
    Delete a queue.

payload.db.api.delete_queue_member (agent_uuid, queue_uuid)
    Delete a queue member.

payload.db.api.get_agent (uuid)
    Retrieve information about the given agent.

payload.db.api.get_queue (uuid)
    Retrieve information about the given queue.

payload.db.api.get_queue_member (agent_uuid, queue_uuid)
    Retrieve information about the given queue member.

payload.db.api.list_agents ()
    Retrieve a list of agents.

payload.db.api.list_queue_members (uuid)
    Retrieve a list of queue members.

payload.db.api.list_queues ()
    Retrieve a list of queues.
```

migration Module

```
payload.db.migration.db_sync (version=None)
    Migrate the database to version or the most recent version.

payload.db.migration.db_version ()
    Display the current database version.
```

Indices and tables

- *genindex*
- *modindex*
- *search*

HTTP Routing Table

/queues

GET /queues/:queue_uuid/callers, [27](#)

GET /queues/:queue_uuid/callers/:caller_uuid,
[28](#)

Python Module Index

p

- `payload.__init__`, 15
- `payload.api`, 16
- `payload.api.app`, 16
- `payload.api.config`, 16
- `payload.api.controllers`, 16
- `payload.api.controllers.root`, 17
- `payload.api.controllers.v1`, 17
- `payload.api.controllers.v1.agent`, 17
- `payload.api.controllers.v1.base`, 19
- `payload.api.controllers.v1.controller`, 19
- `payload.api.controllers.v1.queue`, 20
- `payload.api.controllers.v1.queue.caller`, 25
- `payload.api.controllers.v1.queue.member`, 22
- `payload.api.controllers.v1.schema`, 19
- `payload.api.hooks`, 16
- `payload.api.middleware`, 16
- `payload.cmd`, 28
- `payload.cmd.api`, 28
- `payload.cmd.manage`, 28
- `payload.common`, 28
- `payload.common.exception`, 28
- `payload.common.paths`, 29
- `payload.common.schema`, 30
- `payload.db`, 30
- `payload.db.api`, 30
- `payload.db.migration`, 30
- `payload.test`, 15
- `payload.version`, 16

Index

A

`add_command_parsers()` (in module `payload.cmd.manage`), 28
`Agent` (class in `payload.api.controllers.v1.agent`), 17
`agent()` (`payload.api.controllers.v1.schema.SchemasController` method), 19
`AgentNotFound`, 28
`agents` (`payload.api.controllers.v1.controller.Controller` attribute), 19
`AgentsController` (class in `payload.api.controllers.v1.agent`), 18
`APIBase` (class in `payload.api.controllers.v1.base`), 19
`app_factory()` (in module `payload.api.app`), 16
`as_dict()` (`payload.api.controllers.v1.base.APIBase` method), 19

B

`basedir_def()` (in module `payload.common.paths`), 29
`basedir_rel()` (in module `payload.common.paths`), 29
`before()` (`payload.api.hooks.CacheHook` method), 16
`before()` (`payload.api.hooks.DBHook` method), 16
`bindir_def()` (in module `payload.common.paths`), 29
`bindir_rel()` (in module `payload.common.paths`), 29
`build_server()` (in module `payload.api.app`), 16

C

`CacheHook` (class in `payload.api.hooks`), 16
`callers` (`payload.api.controllers.v1.queue.QueuesController` attribute), 22
`code` (`payload.common.exception.NotFound` attribute), 28
`code` (`payload.common.exception.PayloadException` attribute), 29
`config()` (`payload.test.TestCase` method), 15
`Controller` (class in `payload.api.controllers.v1.controller`), 19
`create_agent()` (in module `payload.db.api`), 30

`create_queue()` (in module `payload.db.api`), 30
`create_queue_member()` (in module `payload.db.api`), 30
`created_at` (`payload.api.controllers.v1.agent.Agent` attribute), 17
`created_at` (`payload.api.controllers.v1.base.APIBase` attribute), 19
`created_at` (`payload.api.controllers.v1.queue.caller.QueueCaller` attribute), 25
`created_at` (`payload.api.controllers.v1.queue.member.QueueMember` attribute), 22
`created_at` (`payload.api.controllers.v1.queue.Queue` attribute), 20

D

`Database` (class in `payload.test`), 15
`db_sync()` (in module `payload.db.migration`), 30
`db_version()` (in module `payload.db.migration`), 30
`DBHook` (class in `payload.api.hooks`), 16
`delete()` (`payload.api.controllers.v1.agent.AgentsController` method), 18
`delete()` (`payload.api.controllers.v1.queue.QueuesController` method), 22
`delete_agent()` (in module `payload.db.api`), 30
`delete_queue()` (in module `payload.db.api`), 30
`delete_queue_member()` (in module `payload.db.api`), 30
`description` (`payload.api.controllers.v1.queue.Queue` attribute), 20
`disabled` (`payload.api.controllers.v1.queue.Queue` attribute), 20
`do_db_sync()` (in module `payload.cmd.manage`), 28
`do_db_version()` (in module `payload.cmd.manage`), 28

F

`format_message()` (`payload.common.exception.PayloadException` method), 29

G

get_agent() (in module payload.db.api), 30
 get_all() (payload.api.controllers.v1.agent.AgentsController method), 18
 get_all() (payload.api.controllers.v1.queue.caller.QueueCallerController method), 27
 get_all() (payload.api.controllers.v1.queue.member.QueueMemberController method), 24
 get_all() (payload.api.controllers.v1.queue.QueuesController method), 22
 get_handler_cls() (in module payload.api.app), 16
 get_one() (payload.api.controllers.v1.agent.AgentsController method), 18
 get_one() (payload.api.controllers.v1.queue.caller.QueueCallerController method), 28
 get_one() (payload.api.controllers.v1.queue.member.QueueMemberController method), 24
 get_one() (payload.api.controllers.v1.queue.QueuesController method), 22
 get_pecan_config() (in module payload.api.app), 16
 get_queue() (in module payload.db.api), 30
 get_queue_member() (in module payload.db.api), 30
 get_server_cls() (in module payload.api.app), 16

H

headers (payload.common.exception.PayloadException attribute), 29

L

list_agents() (in module payload.db.api), 30
 list_queue_members() (in module payload.db.api), 30
 list_queues() (in module payload.db.api), 30
 load_app() (in module payload.api.app), 16

M

main() (in module payload.cmd.api), 28
 main() (in module payload.cmd.manage), 28
 member_uuid (payload.api.controllers.v1.queue.caller.QueueCaller attribute), 25
 members (payload.api.controllers.v1.queue.QueuesController attribute), 22
 message (payload.common.exception.AgentNotFound attribute), 28
 message (payload.common.exception.NotFound attribute), 29
 message (payload.common.exception.PayloadException attribute), 29
 message (payload.common.exception.QueueCallerNotFound attribute), 29
 message (payload.common.exception.QueueMemberDuplicated attribute), 29
 message (payload.common.exception.QueueMemberNotFound attribute), 29

message (payload.common.exception.QueueNotFound attribute), 29

N

not_found (payload.api.controllers.v1.queue.caller.QueueCaller attribute), 25
 not_found (payload.api.controllers.v1.queue.QueueMember attribute), 21
 NotFound, 28
 number (payload.api.controllers.v1.queue.caller.QueueCaller attribute), 25
 number (payload.api.controllers.v1.queue.member.QueueMember attribute), 23

P

payload.common.exception (class in payload.common.exception), 16
 path_get() (payload.test.TestCase method), 15
 paused (payload.api.controllers.v1.queue.member.QueueMember attribute), 23
 paused_at (payload.api.controllers.v1.queue.member.QueueMember attribute), 23
 payload.__init__ (module), 15
 payload.api (module), 16
 payload.api.app (module), 16
 payload.api.config (module), 16
 payload.api.controllers (module), 16
 payload.api.controllers.root (module), 17
 payload.api.controllers.v1 (module), 17
 payload.api.controllers.v1.agent (module), 17
 payload.api.controllers.v1.base (module), 19
 payload.api.controllers.v1.controller (module), 19
 payload.api.controllers.v1.queue (module), 20
 payload.api.controllers.v1.queue.caller (module), 25
 payload.api.controllers.v1.queue.member (module), 22
 payload.api.controllers.v1.schema (module), 19
 payload.api.hooks (module), 16
 payload.api.middleware (module), 16
 payload.cmd (module), 28
 payload.cmd.api (module), 28
 payload.cmd.manage (module), 28
 payload.common (module), 28
 payload.common.exception (module), 28
 payload.common.paths (module), 29
 payload.common.schema (module), 30
 payload.db (module), 30
 payload.db.api (module), 30
 payload.db.migration (module), 30
 payload.test (module), 15
 payload.version (module), 16
 PayloadException, 29
 position (payload.api.controllers.v1.queue.caller.QueueCaller attribute), 26

post() (payload.api.controllers.v1.agent.AgentsController method), 18
 post() (payload.api.controllers.v1.queue.QueuesController method), 22
 project_id (payload.api.controllers.v1.agent.Agent attribute), 17
 project_id (payload.api.controllers.v1.queue.Queue attribute), 21
 put() (payload.api.controllers.v1.agent.AgentsController method), 18
 put() (payload.api.controllers.v1.queue.QueuesController method), 22

Q

Queue (class in payload.api.controllers.v1.queue), 20
 queue() (payload.api.controllers.v1.schema.SchemasController method), 19
 queue_caller() (payload.api.controllers.v1.schema.SchemasController method), 19
 queue_id (payload.api.controllers.v1.queue.caller.QueueCaller attribute), 26
 queue_id (payload.api.controllers.v1.queue.member.QueueMember attribute), 23
 QueueCaller (class in payload.api.controllers.v1.queue.caller), 25
 QueueCallerNotFound, 29
 QueueCallersController (class in payload.api.controllers.v1.queue.caller), 27
 QueueMember (class in payload.api.controllers.v1.queue.member), 22
 QueueMemberDuplicated, 29
 QueueMemberNotFound, 29
 QueueMembersController (class in payload.api.controllers.v1.queue.member), 24
 QueueNotFound, 29
 queues (payload.api.controllers.v1.controller.Controller attribute), 19
 QueuesController (class in payload.api.controllers.v1.queue), 22

R

raw() (payload.common.schema.Schema method), 30
 Redis (class in payload.test), 15
 RootController (class in payload.api.controllers.root), 17

S

safe (payload.common.exception.PayloadException attribute), 29
 Schema (class in payload.common.schema), 30
 schemas (payload.api.controllers.v1.controller.Controller attribute), 19
 SchemasController (class in payload.api.controllers.v1.schema), 19
 setUp() (payload.test.Database method), 15

setUp() (payload.test.Redis method), 15
 setUp() (payload.test.TestCase method), 15
 setup_app() (in module payload.api.app), 16
 state_path_def() (in module payload.common.paths), 29
 state_path_rel() (in module payload.common.paths), 29
 status (payload.api.controllers.v1.queue.caller.QueueCaller attribute), 26
 status (payload.api.controllers.v1.queue.member.QueueMember attribute), 24
 status_at (payload.api.controllers.v1.queue.caller.QueueCaller attribute), 26
 status_at (payload.api.controllers.v1.queue.member.QueueMember attribute), 24

T

TestCase (class in payload.test), 15

U

updated_at (payload.api.controllers.v1.agent.Agent attribute), 17
 updated_at (payload.api.controllers.v1.base.APIBase attribute), 19
 updated_at (payload.api.controllers.v1.queue.Queue attribute), 21
 user_id (payload.api.controllers.v1.agent.Agent attribute), 18
 user_id (payload.api.controllers.v1.queue.Queue attribute), 21
 uuid (payload.api.controllers.v1.agent.Agent attribute), 18
 uuid (payload.api.controllers.v1.queue.caller.QueueCaller attribute), 27
 uuid (payload.api.controllers.v1.queue.member.QueueMember attribute), 24
 uuid (payload.api.controllers.v1.queue.Queue attribute), 22

V

v1 (payload.api.controllers.root.RootController attribute), 17
 VersionSelectorApplication (class in payload.api.app), 16