SlaveAPI Documentation

Release 1

Ben Hearsum

June 26, 2015

Contents

Contents:

CHAPTER 1

API

1.1 Endpoints

1.1.1 /results

class slaveapi.web.results.**Results**
Provides results from previously requested actions.

get()
Returns all results from all previously requested actions in JSON format.

Returns: See *slaveapi.actions.results.dictify_results()* for details on the format of the returned data.

1.1.2 /slaves/:slave/actions/aws_create_instance

class slaveapi.web.slave.**AWSCreateInstance**(**args*, ***kwargs*)
Request an AWS slave to be instantiated. See *slaveapi.web.action_base.ActionView* for details on GET and POST methods. See :py:func:‘slaveapi.actions.aws_create_instance.aws_create_instance for details on what options are supported

1.1.3 /slaves/:slave/actions/buildslave_last_activity

class slaveapi.web.slave.**GetLastActivity**(**args*, ***kwargs*)
Request the last activity age (in seconds). See *slaveapi.web.action_base.ActionView* for details on GET and POST methods. See *slaveapi.actions.buildslave_last_activity.buildslave_last_activi* for details on how LastActivity is retrieved.

1.1.4 /slaves/:slave/actions/buildslave_uptime

class slaveapi.web.slave.**GetUptime**(**args*, ***kwargs*)
Request the build slave uptime (in seconds). See *slaveapi.web.action_base.ActionView* for details on GET and POST methods. See *slaveapi.actions.buildslave_uptime.buildslave_uptime()* for details on how Uptime is retrieved.

1.1.5 /slaves/:slave/actions/disable

class slaveapi.web.slave.**Disable**(**args*, ***kwargs*)
Request a slave to be disabled. See *slaveapi.web.action_base.ActionView* for details on GET and POST methods. See *slaveapi.actions.disable.disable()* for details on what options are supported

1.1.6 /slaves/:slave/actions/reboot

class slaveapi.web.slave.**Reboot**(**args*, ***kwargs*)
Request a reboot of a slave or get status on a previously requested reboot. See *slaveapi.web.action_base.ActionView* for details on GET and POST methods. See *slaveapi.actions.reboot.reboot()* for details on how reboots are performed.

1.1.7 /slaves/:slave/actions/shutdown_buildslave

class slaveapi.web.slave.**ShutdownBuildslave**(**args*, ***kwargs*)
Request a shutdown of a buildslave or get status on a previously requested shutdown. See *slaveapi.web.action_base.ActionView* for details on GET and POST methods. See *slaveapi.actions.shutdown_buildslave.shutdown_buildslave()* for details on how buildslave shutdowns are performed.

1.1.8 /slaves/:slave/actions/start

class slaveapi.web.slave.**AWSStartInstance**(**args*, ***kwargs*)
Start aws instance if it exists. *slaveapi.web.action_base.ActionView* for details on GET and POST methods. See *slaveapi.actions.aws_start_instance.aws_start_instance()* for details on how LastActivity is retrieved.

1.1.9 /slaves/:slave/actions/stop

class slaveapi.web.slave.**AWSStopInstance**(**args*, ***kwargs*)
Stop aws instance if it exists. *slaveapi.web.action_base.ActionView* for details on GET and POST methods. See *slaveapi.actions.aws_stop_instance.aws_stop_instance()* for details on how LastActivity is retrieved.

1.1.10 /slaves/:slave/actions/terminate

class slaveapi.web.slave.**AWSTerminateInstance**(**args*, ***kwargs*)
Terminate aws instance if it exists. *slaveapi.web.action_base.ActionView* for details on GET and POST methods. See *slaveapi.actions.aws_terminate_instance.aws_terminate_instance()* for details on how LastActivity is retrieved.

1.2 Helpers

1.2.1 ActionView

class slaveapi.web.action_base.**ActionView**

Abstract base class for views that expose actions. Subclasses must set "action", which should be a callable that accepts a slave name as its first argument. Subclasses should set this in their __init__ rather than as a class attribute, otherwise Python will turn it into a class method and pass along "self" to the action – which actions don't generally expect. If the action requires extra arguments (eg, arguments sent through POST data), the subclass should override the "post" method and pass them to it.

get(*slave*)

Retrieve results from an action.

URL Args: slave (str): The slave to retrieve results for.

Query Args: requestid (int): If specified, returns only the results for this specific previous action.. If not passed, results from all previous actions of this type are returned.

Returns: The status of the request specified or the status of all previous actions of this type. See *slaveapi.actions.results.ActionResult.to_dict()* for details on what status looks like.

post(*slave*, **action_args*, ***action_kwargs*)

Request an action of a slave.

URL Args: slave: The slave to perform the action against.

Query Args: waittime: How long to wait (in seconds) for the action to complete before returning a requestid to the user and continuing the work in the background.

Returns: The status of the action, after waiting *waittime* for it to complete. See *slaveapi.actions.results.ActionResult.to_dict()* for details on what status looks like.

CHAPTER 2

Actions

2.1 Buildslave Last Activity

class slaveapi.actions.buildslave_last_activity.**buildslave_last_activity**
Get the build slave state, last activity time, and uptime. Returns a dictionary of the form: {

'last_state': # unknown, booting, stopped, ready, running_command 'last_activity_seconds': # last activity age according to twistd.log, in seconds. 'uptime': uptime # machine uptime, in seconds.

}

2.2 Buildslave Uptime

class slaveapi.actions.buildslave_uptime.**buildslave_uptime**
Attempts to retrieve the build slave uptime (time since last reboot). This is done with the "uptime" command on Unix-based OS's, and by running "net statistics server" on windows.

2.3 Create AWS Instance

class slaveapi.actions.aws_create_instance.**aws_create_instance**
Attempts to create an aws instance for a given owner

Parameters

- **email** (*str*) – the full ldap username of owner requesting instance
- **bug** (*str*) – the number of the bug that needs the instance
- **instance_type** (*str*) – accepted values are 'build' and 'test'
- **disambig** (*int*) – A numerical value to help uniquely identify this machine (useful when a user already has an instance but needs a new one of the same type)

Return type tuple

2.4 Disable

class slaveapi.actions.disable.**disable**

Attempts to disable the named slave from buildbot.

Details of what was attempted and the result are reported into the slave's problem tracking bug at the end.

Disabling is done in a series of steps outlined below:

1. Disable In Slavealloc: unchecks enabled box for slave in slavealloc
2. **Stop Buildbot Process: stops buildbot process by either a calling a** graceful_shutdown or a more aggressive forceful reboot.
3. **Update Problem Tracking Bug: reopen problem tracking bug and leave** comment if a 'comment' field was passed

Parameters

- **name** (*str*) – hostname of slave
- **reason** (*str*) – reason we wish to disable the slave
- **force** (*bool*) – force a reboot immediately instead of graceful_shutdown

Return type tuple

2.5 Reboot

class slaveapi.actions.reboot.**reboot**

Attempts to reboot the named slave a series of ways, escalating from peacefully to mercilessly. Details of what was attempted and the result are reported into the slave's problem tracking bug at the end. Reboots are attempted through the following means (from most peaceful to least merciful):

•SSH: Logs into the machine via SSH and reboots it with an appropriate command.

•Mozpool: Connects to Mozpool and issues a reboot command. If the slave has no mozpool interface, this is skipped.

•IPMI: Uses the slave's IPMI interface to initiate a hard reboot. If the slave has no IPMI interface, this is skipped.

•PDU: Powercycles the slave by turning off the power, and then turning it back on.

•Bugzilla: Requests that IT reboot the slave by updating or creating the appropriate bugs.

2.6 Shutdown Buildslave

class slaveapi.actions.shutdown_buildslave.**shutdown_buildslave**

Attempts to gracefully shut down the buildslave process on the named slave. In order to support Windows, this must be done by contacting the Buildbot Master the slave talks to, and requesting the shut down there. (Slave-side graceful shutdown doesn't work on Windows.) Once initiated, the shutdown is confirmed by watching the slave's twistd.log file.

2.7 Start AWS Instance

class slaveapi.actions.aws_start_instance.**aws_start_instance**
Attempts to start an aws instance

Parameters **name** – hostname of slave

Return type tuple of status, msg

2.8 Stop AWS Instance

class slaveapi.actions.aws_stop_instance.**aws_stop_instance**
Attempts to stop an aws instance

Parameters **name** – hostname of slave

Return type tuple of status, msg

2.9 Terminate AWS Instance

class slaveapi.actions.aws_terminate_instance.**aws_terminate_instance**
Attempts to terminate an aws instance

Parameters **name** – hostname of slave

Return type tuple of status, msg

CHAPTER 3

Results

3.1 ActionResult

class slaveapi.actions.results.**ActionResult**(*slave*, *action*, *state=0*, *request_timestamp=0*, *start_timestamp=0*, *finish_timestamp=0*)

Contains basic information about the result of a specific Action.

to_dict(*include_requestid=False*)

Returns the state and text of this ActionResult in a dict. If include_requestid is True, "requestid" will also be present. Example:

```
{
    "state": 2,
    "text": "Great success!",
    "request_timestamp": 1392414314,
    "start_timestamp": 1392414315,
    "finish_timestamp": 1392414316,
    "requestid": "234567832"
}
```

3.2 dictify_results

results.**dictify_results**(*results*)

Returns a dict of ActionResults broken down by slave, action, and requestid. Specific results are processed by slaveapi.actions.results.ActionResults.to_dict(). Example:

```
{
    "linux-ix-slave04": {
        "reboot": {
            "1235543252": {
                "state": 2,
                "text": "Great success!",
                "request_timestamp": 1392414314,
                "start_timestamp": 1392414315,
                "finish_timestamp": 1392414316
            }
        }
    },
    "w64-ix-slave05": {
        "reboot": {
```

```
                "5748263211": {
                    "state": 3,
                    "text": "Failure :(",
                    "request_timestamp": 1392414317,
                    "start_timestamp": 1392414318,
                    "finish_timestamp": 1392414319
                }
            }
        }
    }
```

CHAPTER 4

Indices and tables

- genindex
- modindex
- search

A

B

D

G

P

R

S

T