Cisco Software Manager Server

API Guide

This document describes the programmatic interfaces, RESTful APIs, which are supported by Cisco Software Manager Server (CSM Server).

Overview

CSM Server supports a set of finite RESTful APIs. The first step to use these APIs is to acquire an access token using a valid CSM Server username and password. The access token has a time limit of an hour. Once the time expires, the external application must acquire a new access token. Below are examples of how an access token can be acquired.

Example of using curl:
```
curl -u <username>:<password> http://localhost:5000/api/v1/token
```

Example of using Python requests (http://docs.python-requests.org/en/master/)

```
import requests
requests.get("http://localhost:5000/api/v1/token", auth=(<username>,
<password>))
```

Sample JSON Response:
```
{
  "token": "eyJhbGciOiJIUzI1NiIsImV4cCI6MTQzMzE4ODcyNiwiaWF0IjoxNDMzMTg4M"
}
```

Once an access token is acquired, it will be used as a username for the RESTful APIs. The password should use the string "unused". See the Python Example in section Create Hosts.

The APIs in this document use JSON format as input and output. Some implementations require Content-Type: application/json be set in the HTTP header to work properly.

Table of Contents

RESTful APIs

In the Sample Requests in each section, the URL contains the string "localhost:5000". When building your URLs, "localhost" should be replaced with the name or IP address of the server running CSM Server.

1.1 Host APIs

1.1.1 Create Hosts

Creates hosts on CSM Server. Multiple hosts can be created with one request. If an existing host is specified, it will cause an update in the CSM database.

Request Parameters:

Parameter	Required	Type	Length	Description
hostname	Yes	string	50	The hostname of the managed host to be created.
region	Yes	string	100	The region the managed host belongs to. The region must exist in the CSM database.
roles	No	string	100	The roles of the managed host. If multiple roles are specified, they must be comma delimited.
connection_type	Yes	string	N/A	The connection method to the managed host. It must be either "telnet" or "ssh".
ts_or_ip	Yes	string	100	The Terminal Server IP or management IP to the managed host. If multiple IPs (active and standby) are specified, they must be comma delimited.
username	No	string	50	The username to the managed host.
password	No	string	50	The password to the managed host.
port_number	No	string	50	The port number to connect to the managed host. If multiple ports (active and standby) are specified, they must be comma delimited.
jump_host	No	string	100	The jump host name for CSM to log into in order to connect to the managed host. The jump host must exist in the CSM database.

Sample Request:

POST:
http://localhost:5000/api/v1/hosts

Single Host Example: **Multiple Host Example:**

BODY: *BODY:*
[{   *[{*

```
    "hostname": "My Host 1",                    "hostname": "My Host 1",
    "region": "SJ Labs",                        "region": "SJ Labs",
    "roles": "PE",                              "roles": "PE",
    "connection_type": "telnet",                "connection_type": "telnet",
    "ts_or_ip": "172.28.98.2",                  "ts_or_ip": "172.28.98.2",
    "username": "cisco",                        "username": "cisco",
    "password": "cisco"                         "password": "cisco"
}]                                          },
                                            {
                                                "hostname": "My Host 2",
                                                "region": "SJ Labs",
                                                "roles": "PE",
                                                "connection_type": "telnet",
                                                "ts_or_ip": "172.28.98.3",
                                                "username": "cisco",
                                                "password": "cisco"
                                            }
                                        ]
```

Python Example:

```python
import requests
payload = [
    {"hostname": "My Host 1", "region": "SJ Labs", "roles": "PE",
     "connection_type": "telnet", "ts_or_ip": "172.28.98.2",
     "username": "cisco", "password": "cisco"},

    {"hostname": "My Host 2", "region": "SJ Labs", "roles": "PE",
     "connection_type": "telnet", "ts_or_ip": "172.28.98.2",
     "username": "cisco", "password": "cisco"},
]
response = requests.post("http://localhost:5000/api/v1/hosts",
                         auth=(token, "unused"), json=payload)
```

Sample Response:

```
{
  "api_response": {
    "host_list": [ {"status": "SUCCESS", "hostname": "My Host 1"},
                   {"status": "SUCCESS", "hostname": "My Host 2"} ]
  }
}
```

Possible Error Codes:

HTTP Code	Possible Scenario
400	Bad Request; check that POST or GET is correct.
	Bad input parameters—check that valid input was given.
401	Invalid credentials or expired token.
	User does not have the required permissions.

1.1.2 Get Hosts

Returns managed hosts specified by their region and family.

Request Parameters:

Parameter	Required	Type	Length	Description
family	No	string	20	The family the managed hosts belong to. (e.g. ASR9K, CRS, NCS1K, NCS5K, NCS5500, NCS6K)
region	No	string	100	The region the managed hosts belong to.
page	No	int		The page number. Each page returns 1000 entries.

Sample Request:

```
GET:
http://localhost:5000/api/v1/hosts
http://localhost:5000/api/v1/hosts?region=SJ Labs
http://localhost:5000/api/v1/hosts?region=SJ Labs&page=2
http://localhost:5000/api/v1/hosts?region=SJ%20Labs&family=ASR9K
```

Sample Response:

```
{
  "api_response": {
    "host_list": [
      {
        "connection_type": "telnet",
        "family": "ASR9K",
        "hardware": "ASR-9006",
        "os_type": "XR",
        "ts_or_ip": "10.122.54.101",
        "host_username": "cisco",
        "hostname": "R1",
        "jump_host": "",
        "port_number": "",
        "region": "RTP-SVS",
        "roles": "P",
        "software_platform": "ASR9K",
        "software_version": "5.3.0"
      },
      {
        "connection_type": "telnet",
        "family": "ASR9K",
        "hardware": "ASR-9904",
        "os_type": "eXR",
        "ts_or_ip": "10.48.32.235",
        "host_username": "iox",
        "hostname": "R2",
        "jump_host": "",
        "port_number": "",
```

```
      "region": "SJ Labs",
      "roles": "",
      "software_platform": "ASR9K-64",
      "software_version": "6.1.0.06I"
    },
  ]
 },
 "current_page": 1,
 "total_pages": 1
}
```

Possible Error Codes:

HTTP Code	Possible Scenario
400	Bad Request; check that POST or GET is correct.
	Unknown page number; there are fewer results than would require the number of pages input.
	Unknown region; the region provided does not exist in the database. Check that the name was input correctly.
401	Invalid credentials or expired token.
	User does not have the required permissions.

1.1.3 Delete Host

Deletes a managed host.

Request Parameters:

Parameter	Required	Type	Length	Description
hostname	Yes	string	50	The hostname of the managed host to be deleted.

Sample Request:

```
DELETE:
http://localhost:5000/api/v1/hosts/<hostname>/delete
```

Sample Response:

```
{
  "api_response": {
    "status": "SUCCESS",
    "hostname": "My Host"
  }
}
```

Possible Error Codes:

HTTP Code	Possible Scenario
400	Server encountered an error.
401	Invalid credentials or expired token.
	User does not have the required permissions.

1.2 CCO APIs

1.2.1 Get CCO Catalog

Returns the platforms and releases that are displayed under the CCO menu. The platform and release can be used to retrieve software information in other CCO related APIs.

Request Parameters:

None

Sample Request:

```
GET:
http://localhost:5000/api/v1/cco/catalog
```

Sample Response:

```
{
  "api_response": {
    "asr9k_px": [
      "6.0.0",
      "5.3.3",
      "5.3.2",
      "5.3.1"
    ],
    "crs_px": [
      "5.3.3",
      "5.3.2"
    ],
    "ncs6k": [
      "5.2.5",
      "5.2.4",
      "5.2.3"
    ],
    "ncs6k_sysadmin": [
      "5.2.5",
      "5.2.4",
      "5.2.3"
    ]
  }
}
```

Possible Error Codes:

HTTP Code	Possible Scenario
401	Invalid credentials or expired token.

1.2.2 Get CCO Software

Returns all software information for a particular release and platform since a particular date (i.e. CCO posted date).

Request Parameters:

Parameter	Required	Type	Length	Description
platform	Yes	string	N/A	The platform that is supported by the CCO software catalog, as shown in the get catalog response.
release	Yes	string	N/A	The release that is supported by the CCO software catalog, as shown in the get catalog response.
date	No	string	N/A	The date the software (SMU/Service Pack) was posted. The date should be in mm-dd-yyyy format.
optimal	No	string	N/A	By default, only optimal software is returned. If this value is "false", it will return all posted software.

Sample Request:

```
GET:
http://localhost:5000/api/v1/cco/software?platform=asr9k_px&release=5.3.3
http://localhost:5000/api/v1/cco/software?platform=asr9k_px&release=5.3.3&dat
e=12-20-2015
http://localhost:5000/api/v1/cco/software?platform=asr9k_px&release=5.3.3&dat
e=12-20-2015&optimal=false
```

Sample Response:

```
{
  "api_response": {
    "software_list": [
      {
        "impact": "Needs Reboot",
        "package_bundles": "asr9k-mini-px",
        "compressed_image_size": "4261807",
        "posted_date": "04/19/2016 17:14:11 PDT",
        "composite_DDTS": "",
        "functional_areas": "QOS",
        "superseded_by": "",
        "supersedes": "",
        "name": "asr9k-px-5.3.3.CSCux31992",
        "prerequisites": "",
        "ddts": "CSCux31992",
```

```
      "status": "Posted",
      "uncompressed_image_size": "18851717",
      "type": "Recommended",
      "id": "AA11103",
      "description": "nV Edge IRL flap by Bay0 MPA reload, even if we
change timeout of UDLD"
    },
    {
      "impact": "Hitless",
      "package_bundles": "asr9k-mpls-px",
      "compressed_image_size": "216967",
      "posted_date": "08/08/2016 12:57:37 PDT",
      "composite_DDTS": "",
      "functional_areas": "SNMP",
      "superseded_by": "",
      "supersedes": "",
      "name": "asr9k-px-5.3.3.CSCva29114",
      "prerequisites": "",
      "ddts": "CSCva29114",
      "status": "Posted",
      "uncompressed_image_size": "633293",
      "type": "Optional",
      "id": "AA11833",
      "description": "SNMP mplsL3VpnVrfRteInetCidrProto Error: OID not
increasing"
    },
    ...
   ]
  }
}
```

Possible Error Codes:

HTTP Code	Possible Scenario
401	Invalid credentials or expired token.

1.2.3 Get CCO Software Entry

Returns information related to a specific software item by its name or ID (e.g. a SMU/Sevice Pack/Release Software)

Request Parameters:

Parameter	Required	Type	Length	Description
name_or_id	Yes	string	N/A	The software name or ID as defined (e.g. asr9k-p-4.2.3.CSCut30136 or AA09694).
platform	Yes	string	N/A	The platform that is supported by the CCO software catalog, as shown in the get catalog response.

release	Yes	string	N/A	The release that is supported by the CCO software catalog, as shown in the get catalog response.

Sample Request:

```
GET:
http://localhost:5000/api/v1/cco/software/<name_or_id>?platform=asr9k_px&rele
ase=5.3.3
```

Sample Response:

```
{
  "api_response": {
    "composite_DDTS": "",
    "compressed_image_size": "113616329",
    "ddts": "CSCuz05961",
    "description": "Link Flaps : Adaptive FEC control algorithm not working
in 4x10GE",
    "functional_areas": "ETHER",
    "id": "AA11308",
    "impact": "Needs Reboot",
    "name": "asr9k-px-5.3.3.CSCuz05961",
    "package_bundles": "asr9k-mini-px",
    "posted_date": "04/27/2016 21:39:38 PDT",
    "prerequisites": "asr9k-px-5.3.3.CSCux24553",
    "prerequisites_smu_ids": "AA11262",
    "status": "Posted",
    "superseded_by": "",
    "superseded_by_smu_ids": "",
    "supersedes": "asr9k-px-5.3.3.CSCtz68435,asr9k-px-5.3.3.CSCux32820,asr9k-
px-5.3.3.CSCuv63743,asr9k-px-5.3.3.CSCuy75598,asr9k-px-
5.3.3.CSCuy47708,asr9k-px-5.3.3.CSCuy71556,asr9k-px-5.3.3.CSCux20499,asr9k-
px-5.3.3.CSCuy32183,asr9k-px-5.3.3.CSCux85576",
    "supersedes_smu_ids":
"AA11011,AA11051,AA11161,AA11159,AA11117,AA11223,AA10928,AA11036,AA10944",
    "type": "Optional",
    "uncompressed_image_size": "181961047"
  }
}
```

Possible Error Codes:

HTTP Code	Possible Scenario
401	Invalid credentials or expired token.
404	No results for given input.

1.3 Schedule an Installation

Multiple install operations can be specified through one API request. The following pre-defined strings can be used for an install_action: "Pre-Upgrade", "Add", "Activate", "Post-Upgrade", "Commit", "Remove", and "Deactivate". When multiple install actions are specified for the same host, CSM will enforce an implicit dependency in the order shown below. For example, if both Add and Post-Upgrade are scheduled, Add will be a dependency for Post-Upgrade. Until Add is successfully executed, Post-Upgrade will not proceed. Only one install of each install_action can be submitted in a single request per host.

Pre-Upgrade ← Add ← Activate ← Post-Upgrade ← Commit

Scheduling an installation involves two phases: validation and creation. In the validation phase, user input will be analyzed, and any requests with invalid or incomplete entries will be reported back to the user for resubmission. The validation phase does not check dependencies, which are calculated in the creation phase. Once all other input in the request is valid, install jobs will be created.

Note: For an example of the format for submitting multiple install operations in a single request, refer to the Multiple Host Example in the Create Host section.

1.3.1 Install Action: Pre-Upgrade, Post-Upgrade, Commit

Request Parameters:

Parameter	Required	Type	Length	Description
hostname	Yes	string	N/A	The hostname of a managed host. The host must exist in the CSM database.
install_action	Yes	string	N/A	Either "Pre-Upgrade", "Post-Upgrade", or "Commit".
scheduled_time	No	string	N/A	The time this scheduled install should execute. The scheduled time must be in "mm-dd-yyyy hh:mm AM|PM" format. If no scheduled time is given, it will be set to the current time.
utc_offset**	No	string	N/A	The UTC offset in the form <+|->dd:dd and be between -14:00 and +12:00, e.g. +08:00 or -10:00.
command_profile	No	string	N/A	The custom command profile must exist in the CSM database. This parameter is not applicable to "Commit".
dependency*	No	string	N/A	Either the ID of the specific install job, or the install_action on which the install should be dependent.

*In the case that the dependency submitted is an install_action instead of a job ID, the install job will be dependent on the latest-scheduled job of that action type that is scheduled to run earlier than the new job. In the example below, if there are two Add jobs in the database with scheduled_time earlier than the Post-Upgrade, and one Add job scheduled later than the Post-Upgrade, the Post-Upgrade will become dependent on the second of the two earlier jobs.

Dependencies will automatically be computed for jobs that are submitted for the same host in the same request as follows: Pre-Upgrade ← Add ← Activate ← Post-Upgrade ← Commit

**If scheduled_time is submitted, utc_offset is also required.

Sample Request:

```
POST:
http://localhost:5000/api/v1/install

BODY:
[{
  "hostname": "R2",
  "install_action": "Pre-Upgrade",
  "scheduled_time": "06-15-2016 03:15 PM",
  "utc_offset": "-07:00",
  "command_profile": "Test Commands",
  "dependency": "118"
}]
```

Sample Response:

The id can be used for querying the installation status and session logs.

```
{
  "api_response": {
    "install_job_list": [
      {
        "status": "SUCCESS",
        "id": 134,
        "hostname": "R2"
        "install_action": "Pre-Upgrade"
      }
    ]
  }
}

OR

{
  "api_response": {
    "software_list": [
      {
        "status": "FAILED",
        "scheduled_time": "06-15-2016 03:15 PM",
        "hostname": "R2",
        "install_action": "Pre-Upgrade",
        "command_profile": "Test Commands",
        "status_message": "Missing utc_offset."
        "dependency": "118"
      }
    ]
  }
}
```

Possible Error Codes:

HTTP Code	Possible Scenario
400	Bad Request; check that POST or GET is correct.
	Input format is incorrect; should be a list of JSON objects.
	One or more input parameters is invalid, check the returned status messages.
	One or more submitted jobs was a duplicate.
401	Invalid credentials or expired token.
	User does not have required permissions.

1.3.2 Install Action: Add

Request Parameters:

Parameter	Required	Type	Length	Description
hostname	Yes	string	N/A	The hostname of a managed host. The host must exist in the CSM database.
install_action	Yes	string	N/A	The install_action must be "Add"
server_repository	Yes	string	N/A	The server repository where the software packages can be located. The server repository must exist in the CSM database.
server_directory	No	string	N/A	The relative path from the home directory.
software_packages	Yes	string	N/A	The software package names must be comma delimited and must be locatable in the designated server repository.
scheduled_time	No	string	N/A	The time this scheduled install should execute. The scheduled time must be in "mm-dd-yyyy hh:mm AM|PM" format. If no scheduled time is given, it will be set to the current time.
utc_offset**	No	string	N/A	The UTC offset in the form <+|->dd:dd, e.g. +08:00 or -10:00.
dependency*	No	string	N/A	Either the ID of the specific install job, or the install_action on which the install should be dependent.

*In the case that the dependency submitted is an install_action instead of a job ID, the install job will be dependent on the latest-scheduled job of that action type that is scheduled to run earlier than the new job. In the example below, if there are two Add jobs in the database with scheduled_time earlier than the Post-Upgrade, and one Add job scheduled later than the Post-Upgrade, the Post-Upgrade will become dependent on the second of the two earlier jobs.

Dependencies will automatically be computed for jobs that are submitted for the same host in the same request as follows: Pre-Upgrade ← Add ← Activate ← Post-Upgrade ← Commit

**If scheduled_time is submitted, utc_offset is also required.

Sample Request:

```
POST:
http://localhost:5000/api/v1/install

BODY:
[{
    "hostname": "R2",
    "install_action": "Add",
    "scheduled_time": "06-21-2016 07:05 AM",
    "utc_offset": "-07:00",
    "server_repository": "My_Repository",
    "software_packages": "asr9k-px-5.3.3.CSCuz05961.pie,asr9k-px-
5.3.3.CSCux89921.pie,asr9k-px-5.3.3.CSCuy03335.pie",
    "dependency": "2"
}]
```

Sample Response:

(Same as the sample response for Pre-Upgrade, Post-Upgrade, and Commit above)

Possible Error Codes:

(Same as the troubleshooting for Pre-Upgrade, Post-Upgrade, and Commit above)

1.3.3 Install Action: Activate, Remove, Deactivate

Request Parameters:

Parameter	Required	Type	Length	Description
hostname	Yes	string	N/A	The hostname of a managed host. The host must exist in the CSM database.
install_action	Yes	string	N/A	Either "Activate", "Remove", or "Deactivate".
software_packages	Yes	string	N/A	The software package names must be comma delimited. **Activate:** The software package name must appear in the Inactive Package area on the host or an external filename as on CCO. **Remove:** The software package name must appear in the Inactive Package area on the host. **Deactivate:** The software package name must appear in the Active Package area on the host.
scheduled_time	No	string	N/A	The time this scheduled install should execute. The scheduled_time must be in "mm-dd-yyyy

				hh:mm AM|PM" format. If no scheduled time is given, it will be set to the current time.
utc_offset**	No	string	N/A	The UTC offset in the form <+|->dd:dd, e.g. +08:00 or -10:00.
dependency*	No	string	N/A	Either the ID of the specific install job, or the install_action on which the install should be dependent. This parameter is not applicable to "Remove" or "Deactivate".

*In the case that the dependency submitted is an install_action instead of a job ID, the install job will be dependent on the latest-scheduled job of that action type that is scheduled to run earlier than the new job. In the example below, if there are two Add jobs in the database with scheduled_time earlier than the Post-Upgrade, and one Add job scheduled later than the Post-Upgrade, the Post-Upgrade will become dependent on the second of the two earlier jobs.

Dependencies will automatically be computed for jobs that are submitted for the same host in the same request as follows: Pre-Upgrade ← Add ← Activate ← Post-Upgrade ← Commit

**If scheduled_time is submitted, utc_offset is also required.

Sample Request:

```
POST:
http://localhost:5000/api/v1/install

BODY:
[{
    "hostname": "R1",
    "install action": "Activate",
    "scheduled_time": "06-02-2016 08:00 AM",
    "utc_offset": "-07:00",
    "software_packages": "asr9k-px-5.3.3.CSCuz05961.pie,asr9k-px-
5.3.3.CSCux89921.pie,asr9k-px-5.3.3.CSCuy03335.pie",
    "dependency": "1"
}]
```

Sample Response:
(Same as response for Pre-Upgrade, Post-Upgrade, and Commit above)

Possible Error Codes:
(Same as the troubleshooting for Pre-Upgrade, Post-Upgrade, and Commit above)

1.3.4 Get Install Jobs

Returns JSON data on install jobs specified by the id, host, install_action, or status; or jobs that have scheduled times later than or equal to the submitted scheduled_time. Multiple criteria can be used at the same time. If the submitted request would return more than 5000 entries, and error message will be shown asking the user to further refine the query until it results in fewer than 5000 entries.

Request Parameters:

Parameter	Required	Type	Length	Description
id	No	int		The id of the install job to query. If the id is specified, all other parameters will be ignored.
hostname	No	string	N/A	The host the install jobs belong to.
install_action	No	string	N/A	The install_action to query. Must be "Pre-Upgrade", "Add", "Activate", "Post-Upgrade", "Commit", "Remove", or "Deactivate".
status	No	string	N/A	Acceptable values are "scheduled", "in-progress", "completed", and "failed".
scheduled_time*	No	string	N/A	Return entries scheduled on or after the submitted time. The scheduled time must be in "mm-dd-yyyy hh:mm AM|PM" format.
utc_offset**	No	string	N/A	The UTC offset in the form <+|->dd:dd, e.g. +08:00 or -10:00.

*If scheduled_time is specified, utc_offset must also be specified.

**If utc_offset is specified, even without a scheduled_time, all time values will be returned in local time rather than GMT.

Sample Request:

```
GET:
http://localhost:5000/api/v1/install
http://localhost:5000/api/v1/install?id=1
http://localhost:5000/api/v1/install?hostname=R1
http://localhost:5000/api/v1/install?hostname=R1&install action=Add
http://localhost:5000/api/v1/install?hostname=R1&status=failed
http://localhost:5000/api/v1/install?scheduled_time= 08-02-2016 08:00
        AM&utc_offset=+07:00
```

Sample Response:

```
{
  "api_response": {
    "install_job_list": [
      {
        "id": 20,
        "install_action": "Add",
        "dependency": 19,
        "server": "My_Server",
        "packages": "asr9k-px-5.3.3.CSCuz05961.pie,
          asr9k-px-5.3.3.CSCux89921.pie,
          asr9k-px-5.3.3.CSCuy03335.pie",
        "pending_downloads": "",
        "scheduled_time": "Wed, 01 Jun 2016 15:15:00 GMT",
        "start_time": "Wed, 01 Jun 2016 15:20:00 GMT",
        "status": "scheduled",
        "status_time": ""
        "trace": "",
        "created_by": "root",
        "hostname": "R1",
        "custom_command_profile": ""
```

```
        }
      ]
    }
}
```

Possible Error Codes:

HTTP Code	Possible Scenario
400	Bad Request; check that POST or GET is correct.
	Unknown page number; there are fewer results than would require the number of pages input.
	Invalid input parameter, i.e. misspelled hostname, etc. Check that all input is correct.
	Missing required parameter, make sure all required parameters are submitted, including utc_offset if providing a scheduled_time.
	"Too many results; please refine your request." The submitted criteria return too many results. The user must further refine the query by changing or adding more parameters to the request.
401	Invalid credentials or expired token.
	User does not have required permissions.

1.3.5 Delete Install Jobs

Deletes install jobs specified by the id, host, or status. It will also delete any install jobs dependent on the jobs specified. Only jobs that are scheduled or failed, not in-progress or completed, can be deleted.

Request Parameters:

Parameter	Required	Type	Length	Description
id	No	int	N/A	The id of the install job to delete.
hostname	No	string	N/A	The hostname for which all jobs that are not in-progress should be deleted.
status	No	string	N/A	Acceptable values are "failed" or "scheduled". Cannot delete in-progress jobs or completed jobs.

Sample Request:

```
DELETE:
http://localhost:5000/api/v1/install/delete?id=180
http://localhost:5000/api/v1/install/delete?hostname=R1
http://localhost:5000/api/v1/install/delete?hostname=R1&status=failed
```

Sample Response:

```
{
  "api_response": {
    "install_job_list": [
```

```
      {
        "id": 20,
        "status": "SUCCESS"
      },
    ]
  }
}
```

Possible Error Codes:

HTTP Code	Possible Scenario
400	Bad Request; check that POST or GET is correct.
	Invalid value for a parameter.
401	Invalid credentials or expired token.
	User does not have required permissions.

1.3.6 Get Logs

Download the session logs for a particular install job.

Request Parameters:

Parameter	Required	Type	Length	Description
id	Yes	int	N/A	The id of the install job.

Sample Request:

GET:
http://localhost:5000/api/v1/install/logs/180

Receiving zip from response:

The http response will include a zip file containing all of the log files. An example in python for retrieving and extracting that zip file is as follows:

```
import requests
import zipfile
import io

resp =
requests.get("http://localhost:5000/api/v1/install/logs/13",auth=(token,
"unused"))

try:
    zip = zipfile.ZipFile(io.BytesIO(resp.content))
    zip.extractall(directory_path)
except zipfile.BadZipfile:
    print "No session logs."
```

"directory_path" is a string containing the exact path to an existing directory.

Possible Error Codes:

HTTP Code	Possible Scenario
400	Bad Request; check that POST or GET is correct.
	ID invalid or not specified.
401	Invalid credentials or expired token.
	User does not have required permissions.

End of Document