Voisus Server API

Advanced Simulation Technology inc.

500 A Huntmar Park Drive • Herndon, Virginia 20170 U.S.A.
Tel. (703)471-2104 • Fax. (703)471-2108
www.asti-usa.com

Contents

The Voisus Server API gives developers the tools to integrate Voisus into their own applications. For example, a developer might wish to add the following Voisus functionality to a custom interface:

- Create and manage Scenarios for simulated radio comms
- Download Voisus software clients
- Manage hardware and software clients
- Monitor simulated radios on the network
- Record and replay network traffic
- Bridge live radios to the network
- Remotely control live radios over the network

1 Overview

The Voisus API follows REST standards. Representational State Transfer (REST) is an architectural style for distributed systems. A basic understanding of REST is useful for understanding this API. Many resources are available online, including Roy Fielding's dissertation[1], which contains the original definition of REST.

1.1 Resources

Voisus resources are identified by the URL path, which follows the host name in the URL. All API resources are accessed from the top-level resource, `/api/`, which contains all of the Voisus resources. For example, on a Voisus server with an IP address of 10.2.141.141, the top-level API is found here:

```
https://10.2.141.141/api/
```

1.2 Methods

Standard HTTP Methods are used to interact with the Voisus resources:

HTTP Method	Result
GET	Retrieves a resource
POST	Creates a resource
PUT	Updates a resource
DELETE	Deletes a resource

If you wanted to retrieve information about a particular Scenario, you might construct the following HTTP request:

[1] http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm

`GET /api/scenarios/<scenario_ID>`

Since you want to retrieve the Scenario, you use the HTTP `GET` verb. The path `/scenarios/<scenario_ID>` describes the resource containing information for a specific Scenario.

1.3 JSON

Most HTTP requests and responses to the Voisus API use the media type (also known as MIME type) application/json[2], specifying that the content of the request or response is JSON[3] (JavaScript Object Notation) formatted.

The JSON response contains information about the specific resource as well as links to the resource's children and any related resources. For example, a GET request for the "Fire" net resource: `GET /api/scnearios/<scenario_ID>/nets/<Fire_net_ID>` would return the following JSON:

```
{
  "txfrequency": 0,
  "parent": null,
  "self": "../api/scenarios/<scenario_ID>/nets/<Fire_net_ID>/",
  "rev": "1-90f418e8412cb52ad8fe56176baece5a",
  "crypto": "../api/scenarios/<scenario_ID>/cryptos/<crypto_ID>/",
  "name": "Fire",
  "data_type": "nets",
  "waveform": "../api/scenarios/<scenario_ID>/waveforms/<waveform_ID>/",
  "frequency": 150000000,
  "satcom": null,
  "version": "v5.8.0-567-gbee9459c",
  "freqhop": null,
  "id": "<Fire_net_ID>",
  "description": "Fire Support"
}
```

2 Supported Clients

Many HTTP clients are compatible with the Voisus API, including Chrome and Firefox web browsers and programming language-specific clients such as Python Requests[4] and Java HTTP-Client[5].

When choosing a client, look for the following features:

- Implementation of HTTP methods GET, POST, PUT, DELETE
- Encryption support with HTTPS

[2] http://www.ietf.org/rfc/rfc4627.txt
[3] http://www.json.org/
[4] http://docs.python-requests.org/en/latest/
[5] http://hc.apache.org/httpcomponents-client-ga/index.html

- Digest authentication
- Automatic cookie handling
- Access to response code and headers

Since the Voisus API uses JSON formatting throughout, a separate library for JSON parsing and serialization may also be needed.

3 Using the API

The top-level Voisus API resource is located at `/api/` on each Voisus server. To access the API on a Voisus server with an IP address of 10.2.141.141, start with a `GET` request on the following URL:

```
GET https://10.2.141.141/api/
```

The response contains top-level API information, including links to other Voisus resources:


```
{
  "features": "../api/features/",
  "servers": "../api/servers/",
  "sos": "../api/sos/",
  "es3d": "../api/es3d/",
  "syspower": "../api/syspower/",
  "constructmon": "../api/constructmon/",
  "soundfiles": "../api/soundfiles/",
  "netcfg": "../api/netcfg/",
  "cloud": "../api/cloud/",
  "perfmon": "../api/perfmon/",
  "self": "../api/",
  "vclient": "../api/vclient/",
  "construct": "../api/construct/",
  "scenarios": "../api/scenarios/",
  "version": "../api/version/",
  "speech": "../api/speech/",
  "simscribe": "../api/simscribe/",
  "rc": "../api/rc/",
  "radiomon": "../api/radiomon/",
  "blocks": "../api/blocks/",
  "sessions": "../api/sessions/",
  "models": "../api/models/",
  "downloads": "../api/downloads/",
  "apiversion": 1.0,
  "services": "../api/services/",
  "radiobridges": "../api/radiobridges/",
  "devices": "../api/devices/",
  "streams": "../api/streams/",
```

```
  "backup": "../api/backup/",
  "aboutme": "../api/aboutme/"
}
```

We recommend interacting with the Voisus API by dynamically following the links returned by the API, rather than hard-coding link paths into the client application. For example, rather than hard coding the scenario URL `https://10.2.141.141/api/scenarios/`, the client performs a GET on the top-level API and looks up the Scenarios URL in the returned JSON response. This method of interacting with a REST API is called HATEOAS[6] (Hypermedia as the Engine of Application State). All resources are accessible by following a series of links starting from the top-level URL.

All client requests should provide an `Accept:` header specifying JSON as the media type, unless otherwise noted.

```
Accept: application/json
```

Consult the documentation for your HTTP client library to learn how to set request headers.

4 Scenarios

api/scenarios/

A Scenario resource is a collection of subresources containing all the information needed for a specific training task or simulation, such as:

- **nets:** sets of operational parameters such as frequency and waveform modulation type used to create simulated radios. The nets determine which radios can interoperate.
- **roles:** collections of radios that will be assigned to Voisus clients. Each radio contains one or more nets.
- **dis:** configure settings for Distributed Interactive Simulation (DIS), if applicable.
- **vsclients:** Voisus Clients, including software and hardware clients. Each client is assigned a role, which contains simulated radios.

To list all the Scenarios on the Voisus server, issue a `GET` request on the Scenarios resource:

```
GET https://10.2.141.141/api/scenarios/
```

The JSON response contains a list of Scenario templates, which are preloaded with nets, radios, and roles and can be used to quickly create new Scenarios. The "items" list displays all Scenarios on the server with links to each Scenario's subresources.

[6] http://en.wikipedia.org/wiki/HATEOAS

```
{
"templates": [
    "../api/scenarios/templates/Army_Example",
    "../api/scenarios/templates/Basic_Example",
    "../api/scenarios/templates/Construct_Example",
    "../api/scenarios/templates/EmergencyMgmt_Example",
    "../api/scenarios/templates/HWPanel_Example",
    "../api/scenarios/templates/Intercom_Example",
    "../api/scenarios/templates/Maritime_Example",
    "../api/scenarios/templates/Office_Example",
    "../api/scenarios/templates/RadioBridge4",
    "../api/scenarios/templates/RadioBridge8",
    "../api/scenarios/templates/SeriousGame_Example",
    "../api/scenarios/templates/TOC_Example"
],
"items": [
    {
    "behaviors": "../api/scenarios/<scenario_id>/behaviors/",
    "entity_filters": "../api/scenarios/<scenario_id>/entity_filters/",
    "radio_bridge_presets": "../api/scenarios/<scenario_id>/radio_bridge_presets/",
    "scenario_lockdown": "../api/scenarios/<scenario_id>/scenario_lockdown/",
    "waveforms": "../api/scenarios/<scenario_id>/waveforms/",
    "grammars": "../api/scenarios/<scenario_id>/grammars/",
    "sounds": "../api/scenarios/<scenario_id>/sounds/",
    "nets": "../api/scenarios/<scenario_id>/nets/",
    "id": "<scenario_id>",
    "description": "",
    "remote_controls": "../api/scenarios/<scenario_id>/remote_controls/",
    "response_rules": "../api/scenarios/<scenario_id>/response_rules/",
    "data_type": "scenario",
    "self": "../api/scenarios/<scenario_id>/",
    "subscriptions": "../api/scenarios/<scenario_id>/subscriptions/",
    "rev": "4-2542ac05fd55d56bbc88c86250e457a1",
    "scenario_id": "<scenario_id>",
    "entities": "../api/scenarios/<scenario_id>/entities/",
    "version": "v5.8.0-567-gbee9459c",
    "dis_domains": "../api/scenarios/<scenario_id>/dis_domains/",
    "parent": null,
    "interactions": "../api/scenarios/<scenario_id>/interactions/",
    "vsclients": "../api/scenarios/<scenario_id>/vsclients/",
    "radio_effects": "../api/scenarios/<scenario_id>/radio_effects/",
    "model_templates": "../api/scenarios/<scenario_id>/model_templates/",
    "dis_wfmaps": "../api/scenarios/<scenario_id>/dis_wfmaps/",
    "radio_bridges": "../api/scenarios/<scenario_id>/radio_bridges/",
    "asr_configs": "../api/scenarios/<scenario_id>/asr_configs/",
    "es3d_observers": "../api/scenarios/<scenario_id>/es3d_observers/",
```

```
"name": "Example_Scenario",
"roles": "../api/scenarios/<scenario_id>/roles/",
"fills": "../api/scenarios/<scenario_id>/fills/",
"freqhops": "../api/scenarios/<scenario_id>/freqhops/",
"vehicles": "../api/scenarios/<scenario_id>/vehicles/",
"satcoms": "../api/scenarios/<scenario_id>/satcoms/",
"cryptos": "../api/scenarios/<scenario_id>/cryptos/",
"entity_types": "../api/scenarios/<scenario_id>/entity_types/",
"isactive": true,
"dis": "../api/scenarios/<scenario_id>/dis/"
},
```

Since Scenarios are used in Voisus, Construct, and Radio Bridge, the JSON response contains a wide range of subresources. The following Scenario resources are commonly used to build scenarios for Voisus client communications:

4.1 Create a New Scenario

To create a new blank Scenario, use the `POST` method with the Scenarios resource, providing a JSON object with the name of the new Scenario in the body of the request. For example:

```
POST https://10.2.141.141/api/scenarios/

{ "name": "My New Scenario" }
```

Scenario Templates

api/scenarios/templates/

The Voisus software provides Scenario templates for quick and easy Scenario creation. They are preconfigured with a Comm Plan, Roles and Radios. To view a list of templates available on your Voisus server, issue a `GET` request on the top-level Scenarios template resource. For example:

```
GET https://10.2.141.141/api/scenarios/templates/
```

To create a new Scenario based on a Scenario template, use the `POST` method with the specific Scenarios template resource, providing a JSON object with the name of the new Scenario in the body of the request. For example:

```
POST https://10.2.141.141/api/scenarios/templates/Basic_Example

{ "name": "My New Scenario" }
```

The JSON response describes the new Scenario resource, including its "self" URI. The *Location* header is set in the response, as shown in this example:

```
Location: https://10.2.141.141/api/scenarios/b52f64ba39114e9eacfd0bef32138a65/
```

Table 1: Common Voisus Scenario Resources

Resource	Description
`name`	The name of this Scenario.
`id`	The unique ID for this Scenario.
`description`	An optional description of the Scenario.
`self`	Refers to the unique Scenario ID on the host.
`scenario_id`	The unique ID for this Scenario.
`nets`	Contains the common parameters that networked radios must share for interoperation. At minimum, each net must have a name, frequency, and waveform.
`waveforms`	Contains specific parameters for radio communications. Each `net` must contain a `waveform`.
`fills`	Known as "Netgroups" in the Voisus web interface; a group of nets that can be assigned to a radio.
`freqhops`	An optional parameter for nets; used to simulate frequency hopping.
`satcoms`	An optional parameter for nets; used to simulate satellite communications.
`cryptos`	An optional parameter for nets; used to simulate radios that scramble signals before they are transmitted, and only receivers who have the encryption key can decode the signal.
`roles`	Contains a collection of simulated radios.
`vsclients`	A container resource for the Voisus software and hardware clients assigned to this Scenario.
`scenario_lockdown`	Settings for unlisted Voisus Clients (those not listed in the `vsclients` resource).
`vehicles`	Typically used in VBS2/3 and other games for training. Each vehicle contains a role, and users gain access to the role's radios when they are inside the vehicle.
`dis`	DIS network and configuration settings.
`dis_domains`	Specifies the DIS domain for this Scenario.
`dis_wfmaps`	An advanced resource for DIS exercises; customize the Modulation Type record.
`isactive`	True or False; indicates whether the Scenario is running or not. The server can run one Scenario at a time. Use the Sessions (chapter 5) resource to run or stop a Scenario.

4.2 Update a Scenario Resource

To update a resource, issue a `PUT` request to the "self" URI of the scenario resource, providing the updated resource data in the body of the request:

```
PUT https://10.2.141.141/api/scenarios/b52f64ba39114e9eacfd0bef32138a65/

{ "name": "A New Scenario Name" }
```

4.3 Scenario Resources

Create and update the following resources to build a training Scenario:

Nets

api/scenarios/<scenario_id>/nets/

Communication nets describe the settings used by a radio or intercom. Nets link to waveform, crypto, and freqhop resources, which further define the capabilities of each net.

The JSON response below represents a single net named "Command Net":


```
{
  "txfrequency": 0,
  "parent": null,
  "self": "../nets/62ef404b363d4c899138f8c9be801ac7/",
  "rev": "8-4bd097672949c669de92bce9c4efbdd0",
  "crypto": "../cryptos/9534d1252d884d979d5dbe8dd060a6aa/",
  "name": "Command Net",
  "data_type": "nets",
  "waveform": "../waveforms/eb35319d81a74f0484879dd268f912a3/",
  "frequency": 101000000,
  "satcom": null,
  "version": "v5.11.0-7-gb63f61af",
  "freqhop": "../freqhops/42b655d4da0f43d892ae9b412e70dcdd/",
  "id": "62ef404b363d4c899138f8c9be801ac7",
  "description": ""
}
```


Waveforms

api/scenarios/<scenario_id>/waveforms/

Waveforms define specific parameters of a net. A single waveform may be used in multiple nets as needed.

The JSON response below represents a single waveform named "Waveform–1":


```
{
  "description": "",
```

Variable	Requirement	Definition
`description`	Optional	User-friendly description of the net
`frequency`	*Required*	Transmit and receive frequency (Hz)
`txfrequency`	Optional	Distinct transmit frequency (Hz). This is usually left as zero, which means that the `frequency` value will be used for both Rx and Tx.
`crypto`	Optional	Link to a Crypto resource
`waveform`	*Required*	Link to a Waveform resource
`satcom`	Optional	Link to a Satcom resource
`freqhop`	Optional	Link to a FreqHop resource


```
  "parent": null,
  "encoding": "MULAW",
  "self": "../waveforms/eb35319d81a74f0484879dd268f912a3/",
  "rev": "1-b29eec2523b98a345af236569e421a4d",
  "bandwidth": 25000,
  "data_type": "waveforms",
  "propagation": "RANGING",
  "rate": 8000,
  "version": "v5.11.0-7-gb63f61af",
  "power": 1.0,
  "mode": "FM",
  "modkey": "",
  "id": "eb35319d81a74f0484879dd268f912a3",
  "name": "Waveform-1"
}
```


Variable	Requirement	Definition	Supported Values
`mode`	*Required*	Modulation type	AM, FM, INTERCOM, USB, LSB, HFECCM, SSBF, SINCGARS, SINCGARS_SC, HAVEQUICK, SATCOM, and CW
`bandwidth`	*Required*	Radio passband width (Hz)	Default is 25000
`encoding`	*Required*	Audio encoding type	MULAW, PCM, and CVSD. MULAW is most common.
`rate`	*Required*	Rate of audio encoding	Recommended values: 8000 (MULAW and PCM); 16000 (CVSD)
`power`	*Required*	Tx power (watts/dBm)	Default is 1.0
`propagation`	*Required*	Propagation model type	NONE, RANGING, OCCULTING, and RANGING_AND_OCCULTING

Roles

api/scenarios/<scenario_id>/roles

Roles are collections of radios, and radios are collections of nets.

The JSON response below represents one radio in a role, named "PLT2":


```
{
  "PRC-148": "../api/scenarios/<scenario_id>/roles/<role_id>/PRC-148/",
  "id": "<role_id>",
  "description": "",
  "generic_radio": "../api/scenarios/<scenario_id>/roles/<role_id>/generic_radio/",
  "data_type": "roles",
  "self": "../scenarios/<scenario_id>/roles/<role_id>/",
  "rev": "1-24f643f13de3e1269f73b9c582213160",
  "children": "../api/scenarios/<scenario_id>/roles/<role_id>/children/",
  "version": "v5.8.0-567-gbee9459c",
  "shared": false,
  "commpanel_tmpl": "../api/scenarios/<scenario_id>/roles/<role_id>/commpanel_tmpl/",
  "PRC-119": "../api/scenarios/<scenario_id>/roles/<role_id>/PRC-119/",
  "PRC-152": "../api/scenarios/<scenario_id>/roles/<role_id>/PRC-152/",
  "parent": null,
  "autotune_enabled": false,
  "name": "PLT2",
  "radiohw": 0,
  "calling_enabled": true,
  "SINCGARS": "../api/scenarios/<scenario_id>/roles/<role_id>/SINCGARS/",
  "PRC-117G": "../api/scenarios/<scenario_id>/roles/<role_id>/PRC-117G/",
  "PRC-117F": "../api/scenarios/<scenario_id>/roles/<role_id>/PRC-117F/",
  "chat_enabled": true
}
```


Variable	Definition	Supported Values
`autotune_enabled`	Autotune enables software clients to instantly establish by clicking on a realtime list of active radios on the network.	true or false
`calling_enabled`	Enables the calling feature on Voisus software clients.	true or false
`chat_enabled`	Enables the chat feature on Voisus software clients.	true or false
`children`	The collection of radios assigned to this role.	

Add a Radio to the Role

To add a radio to the role, use the `POST` method to the role's "self" URI with the specific radio (data_type) you wish to create. Voisus currently offers six radio types:

For more information about the Original Desktop Client, Voisus Client for Desktops & Tablets, and hardware clients, see the Voisus Client User Guide[7].

In most cases you will create a generic radio and your request will look like this:

[7] voisus_client_ug.pdf

Table 2: Radio data_type

data_type	Definition
`generic_radio`	Create a standard radio for use with the Desktop Client, Voisus Client for Desktops and Tablets, and most hardware clients (except the PRC–117 panel and SINCGARS panel).
`PRC-117F`	For use with the PRC–117F radio skin in the Voisus Client for Desktops and Tablets or the ASTi PRC–117 hardware panel.
`PRC-117G`	For use with the PRC–117G radio skin in the Voisus Client for Desktops and Tablets.
`PRC-148`	For use with the PRC–148 radio skin in the Voisus Client for Desktops and Tablets.
`PRC-152`	For use with the PRC–152 radio skin in the Voisus Client for Desktops and Tablets.
`SINCGARS`	For use with the ASTi SINCGARS hardware panel.

```
POST https://10.2.141.141/api/scenarios/<scenario_id>/roles/<role_id>/generic_radio
```

The JSON response will look similar to this:

```
{
  "parent": "../api/scenarios/<scenario_id>/roles/<role_id>/",
  "description": "",
  "data_type": "generic_radio",
  "net_lock": false,
  "self": "../api/scenarios/<scenario_id>/roles/<role_id>/generic_radio/<new_radio_id>/",
  "rev": "1-6ca7eda81bef583d22ec5efbe8925595",
  "version": "v5.12.0-597-g7b2c29ae",
  "default_fill": null,
  "entries": [],
  "default_net": null,
  "shared": false,
  "sqtail_disabled": false,
  "id": "<new_radio_id>",
  "name": "Generic_radio-1"
}
```

Add Nets to the Radio

Next, use the `PUT` method on the radio's "self" URI to update the new radio resource with at least one net. This example updates the radio with three nets:

```
PUT https://10.2.141.141/api/scenarios/<scenario_id>/roles/<role_id>/generic_radio/<radio_id>

{ "entries": [
    "https://10.2.141.141/api/scenarios/<scenario_id>/nets/<first_net_id>
```

Variable	Definition	Supported Values
`entries`	The collection of nets assigned to this role.	The "self" URI of each net. The role can have one or many nets.
`default_net`	Assign a default net to this role.	The "self" URI of the net or null
`default_fill`	Assign a default netgroup to this role.	The "self" URI of the fill or null
`net_lock`	Lock the default net so the client with this role cannot change it.	true or false
`shared`	A shared radio simulates a single radio shared by every operator using this role. When a radio is not shared, each operator using this role will have their own radio that only they can control.	true or false

```
    "https://10.2.141.141/api/scenarios/<scenario_id>/nets/<second_net_id>
    "https://10.2.141.141/api/scenarios/<scenario_id>/nets/<third_net_id>
    ]
}
```

Manage Clients

api/scenarios/<scenario_id>/vsclients

Add Clients

Use the `vsclients` resource to add software and hardware clients to the Scenario:

```
POST ../api/scenarios/<scenario_id>/vsclients/

{ "name": "First_Client" }
```

The JSON response will look similar to this:

```
{
"domain": "../api/scenarios/<scenario_id>/dis_domains/a58587c1500b45609c18a590201c3429/",
"description": "",
"parent": null,
"self": "../api/scenarios/<scenario_id>/vsclients/<vsclient_id>/",
"rev": "2-547125ad98d7a291d11f4929a56662f4",
"data_type": "vsclients",
"secretkey": "First_Client",
"sincgars_en": false,
"hwptt_en": false,
"version": "v5.12.0-557-gdd2f74b7",
"role": "../api/scenarios/<scenario_id>/roles/<role_id>/",
"role_lock": false,
"id": "3be6d45a26b743ecb61da3fd320a5f8d",
"name": "First_Client"
```

```
}
```

Use the `PUT` method to update the parameters for this `vsclient`:

Variable	Definition	Supported Values
`domain`	The DIS domain assigned to this client.	The "self" URI of the `dis_domain`
`role`	The default role assigned to this client.	The "self" URI of the role. One role can be assigned to many different clients in an exercise.
`role_lock`	If the role is locked, the client will not be able to change roles	true or false
`name`	The client name used to map these settings to the client	string

Map Clients

The "name" value is used to link the `vsclient` resource settings to the software or hardware client.

Map to Software Clients

Our JSON example above uses the value `"name": "First_Client"`. If you wish to map the "First_Client" settings to a software client, enter the "name" value in the Client Name field, as shown in this example:

When the software client connects to the server, it will map to your `vsclient` settings.

Map to Hardware Clients

If you wish to map the "First_Client" settings to a hardware client, you will need to update the resource **api/devices**. The `devices` resource contains all the information regarding ACENet devices, AI-S devices, and hardware panels detected on the network.

To view all devices, use the `GET` method:

```
GET ../api/devices/
```

Use the `PUT` method to update the `desc` value for the channel that will act as a hardware client. The ACE-RIU and ACU2 each have 4 channels available, and the channels are related to the `desc` fields as follows:

To assign our "First_Client" settings to the ACE-RIU's Channel A + Serial Port A, we need to update the `desc1` value to "First_Client":

Table 3: ACE-RIU

Variable	Channel Association
`desc1`	Channel A + Serial Port A
`desc2`	Channel B
`desc3`	Channel C + Serial Port B
`desc4`	Channel D

Table 4: ACU2

Variable	Channel Association
`desc1`	Channel A + Serial Port 1
`desc2`	Channel B
`desc3`	Channel C + Serial Port 2
`desc4`	Channel D

```
PUT ../api/devices/acenet/<ace_riu_id>/

{ "desc1": "First_Client" }
```

4.4 Delete a Scenario

To delete a Scenario, issue a `DELETE` request on the "self" URI of the Scenario resource. For example:

```
DELETE ../api/scenarios/<scenario_id>
```

5 Sessions

api/sessions/

A Session is a running Scenario. The Voisus server can run one Scenario at a time. Scenario resources can be edited while the Scenario is running.

5.1 Run a Scenario

POST api/sessions/

Run a Scenario by creating a new Session, providing the Scenario URI in the body of the request:

```
POST https://10.2.141.141/api/sessions/

{ "scenario": "../scenarios/b52f64ba39114e9eacfd0bef32138a65/" }
```

The server waits for the Scenario to install before responding to the request, which may take several seconds. You may need to adjust your HTTP client timeout to accomodate this delay.

The JSON response contains a representation of the new Session:

```
{
"install_status": [100, "install finished"],
"cloud_id": "asdf",
"subscriptions": "../sessions/07a6a6cc0c864d8b84274c4b371cfbe9/subscriptions/",
"self": "../sessions/07a6a6cc0c864d8b84274c4b371cfbe9/",
"scenario_href": "../scenarios/b5d54d2a606148d897730aa11a82deca/",
"scenario_id": "b5d54d2a606148d897730aa11a82deca",
"session_id": "07a6a6cc0c864d8b84274c4b371cfbe9",
"session_errors": [],
"install_state": "INSTALLED",
"scenario_name": "test",
"scenario_host": "asdf.local"
}
```


5.2 Stop a Scenario

DELETE api/sessions/<session_id>/

`DELETE` the Session resource to stop the running Scenario:

```
DELETE https://10.2.141.141/api/sessions/07a6a6cc0c864d8b84274c4b371cfbe9/
```

6 Download Voisus Clients

GET api/downloads/

Use the `api/downloads` resource to get links to Voisus software clients available for download.