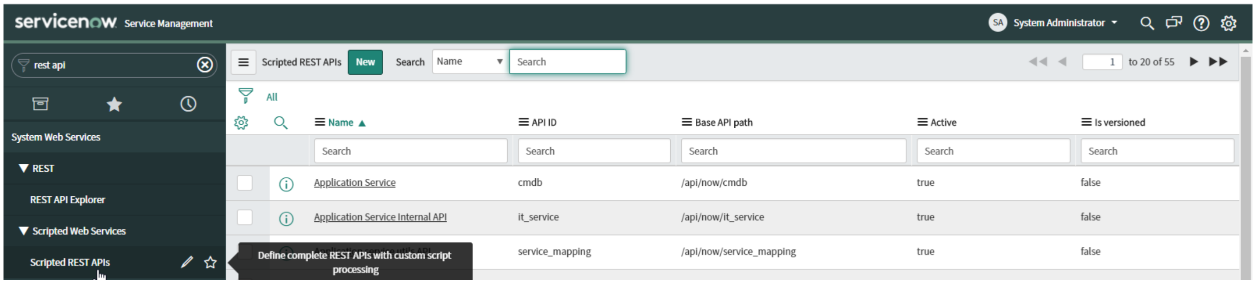
Setting up Webhooks in ServiceNow

*Tested in ServiceNow version “Rome”

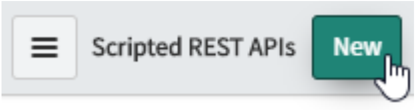
Navigate to System Web Services > Scripted Web Services > Scripted REST APIs



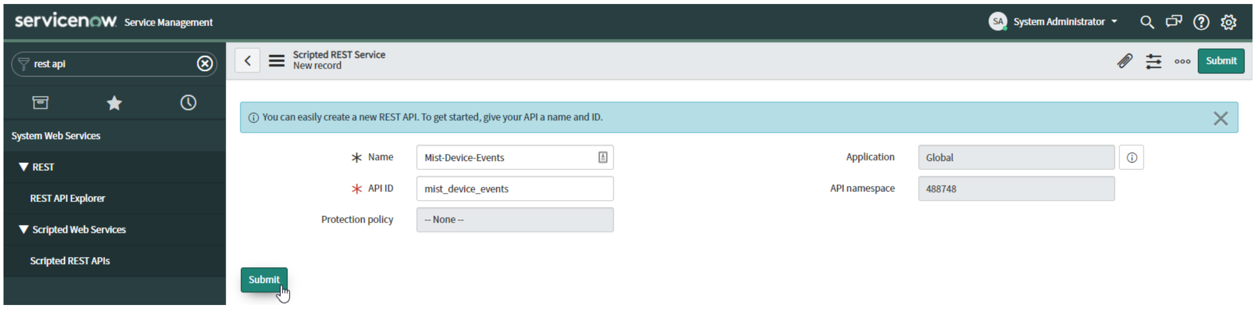
Name ▲	API ID	Base API path	Active	Is versioned
Search	Search	Search	Search	Search
Application Service	cmdb	/api/now/cmdb	true	false
Application Service Internal API	it_service	/api/now/it_service	true	false
	service_mapping	/api/now/service_mapping	true	false

Define complete REST APIs with custom script processing

Create new Scripted REST API



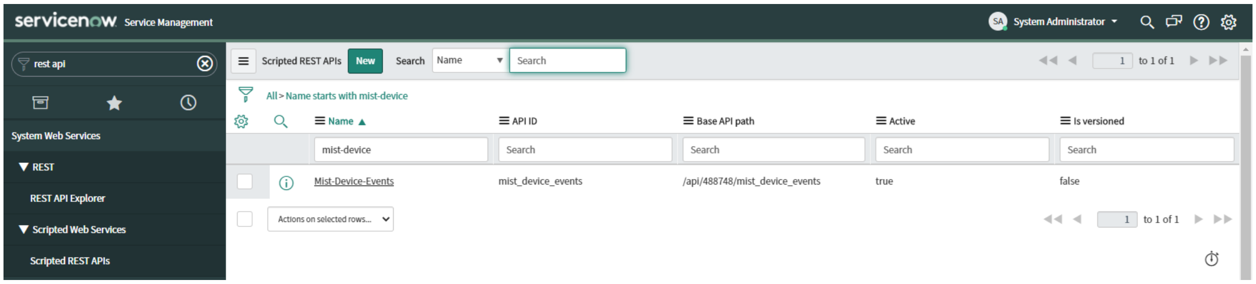
Name the new Scripted REST Service - Note API ID field automatically populates after the Name field un-focuses



ⓘ You can easily create a new REST API. To get started, give your API a name and ID.

* Name	Mist-Device-Events	Application	Global
* API ID	mist_device_events	API namespace	488748
Protection policy	-- None --		

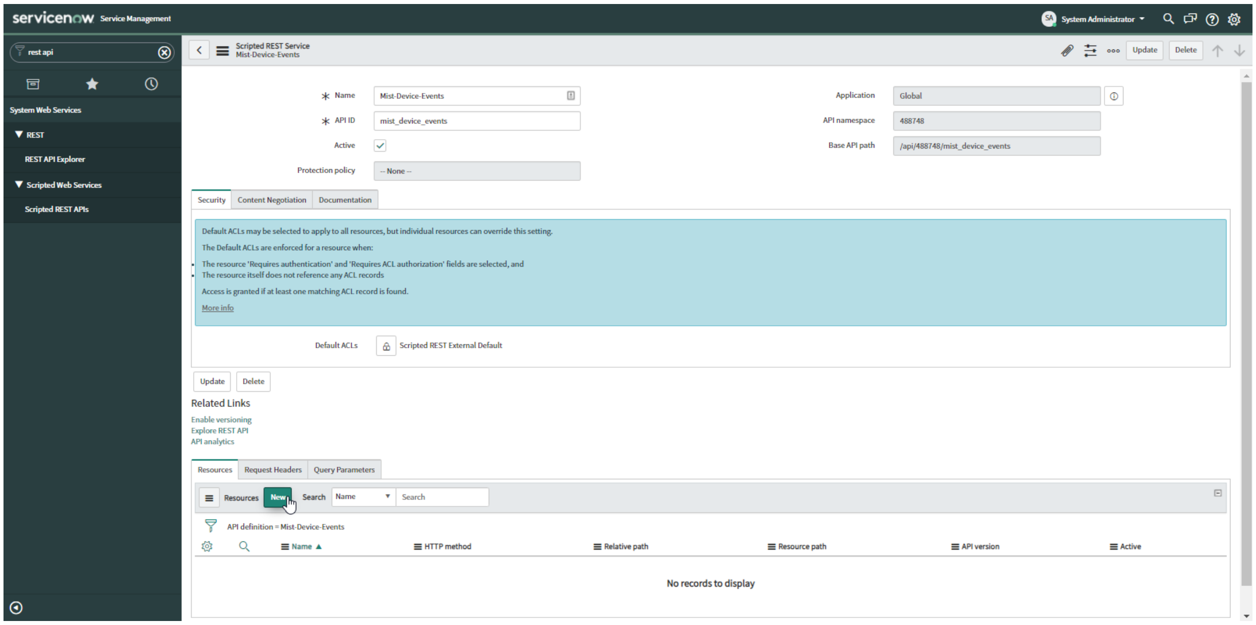
Find and click on the new Scripted REST Service



All > Name starts with mist-device

Name ▲	API ID	Base API path	Active	Is versioned
mist-device	Search	Search	Search	Search
Mist-Device-Events	mist_device_events	/api/488748/mist_device_events	true	false

Scroll to the bottom set of tabs and click New under Resources tab



Scripted REST Service
Mist-Device-Events

* Name	Mist-Device-Events	Application	Global
* API ID	mist_device_events	API namespace	488748
Active	✓	Base API path	/api/488748/mist_device_events
Protection policy	-- None --		

Default ACLs may be selected to apply to all resources, but individual resources can override this setting.

The Default ACLs are enforced for a resource when:

- The resource 'Requires authentication' and 'Requires ACL authorization' fields are selected, and
- The resource itself does not reference any ACL records

Access is granted if at least one matching ACL record is found.

More info

Default ACLs: Scripted REST External Default

Related Links

Enable versioning
Explore REST API
API analytics

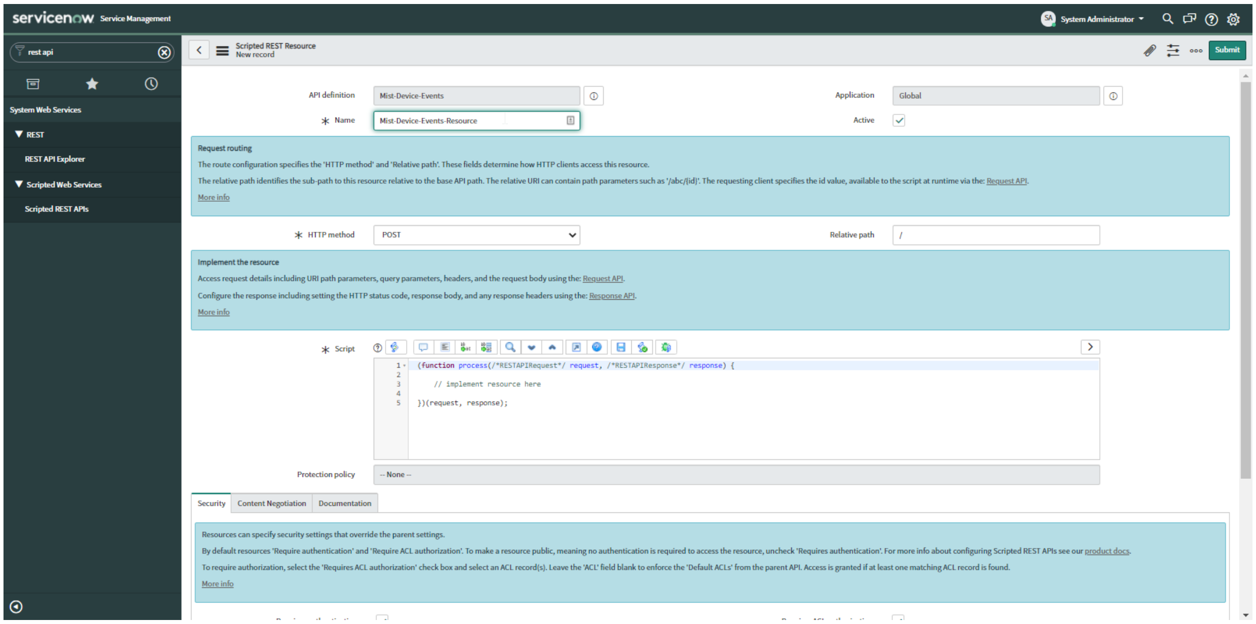
Resources — Search: Name

API definition = Mist-Device-Events

Name ▲	HTTP method	Relative path	Resource path	API version	Active

No records to display

Name the new Resource Appropriately



Scripted REST Resource
New record

API definition	Mist-Device-Events	Application	Global
* Name	Mist-Device-Events-Resource	Active	✓

Request routing

The route configuration specifies the 'HTTP method' and 'Relative path'. These fields determine how HTTP clients access this resource.

The relative path identifies the sub-path to this resource relative to the base API path. The relative URI can contain path parameters such as '/abc/{id}'. The requesting client specifies the id value, available to the script at runtime via the: Request API.

More info

* HTTP method	POST	Relative path	/

Implement the resource

Access request details including URI path parameters, query parameters, headers, and the request body using the: Request API.

Configure the response including setting the HTTP status code, response body, and any response headers using the: Response API.

More info

* Script

```
(function process(/*RESTAPIRequest*/ request, /*RESTAPIResponse*/ response) {

    // implement resource here

})(request, response);
```

Protection policy: -- None --

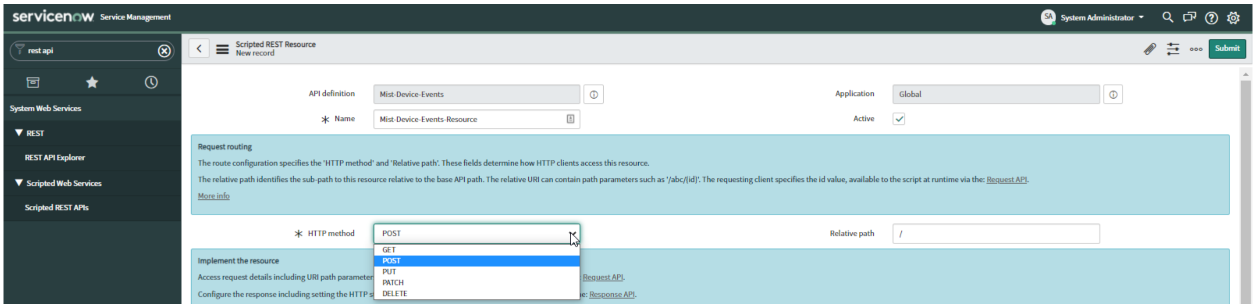
Resources can specify security settings that override the parent settings.

By default resources 'Require authentication' and 'Require ACL authorization'. To make a resource public, meaning no authentication is required to access the resource, uncheck 'Requires authentication'. For more info about configuring Scripted REST APIs see our product docs.

To require authorization, select the 'Requires ACL authorization' check box and select an ACL record(s). Leave the 'ACL' field blank to enforce the 'Default ACLs' from the parent API. Access is granted if at least one matching ACL record is found.

More info

Set HTTP Method to POST



API definition: Mist-Device-Events

Application: Global

* Name: Mist-Device-Events-Resource

Active: ☑

Request routing

The route configuration specifies the 'HTTP method' and 'Relative path'. These fields determine how HTTP clients access this resource.

The relative path identifies the sub-path to this resource relative to the base API path. The relative URI can contain path parameters such as '/abc/{id}'. The requesting client specifies the id value, available to the script at runtime via the: Request API.

More info

* HTTP method: POST

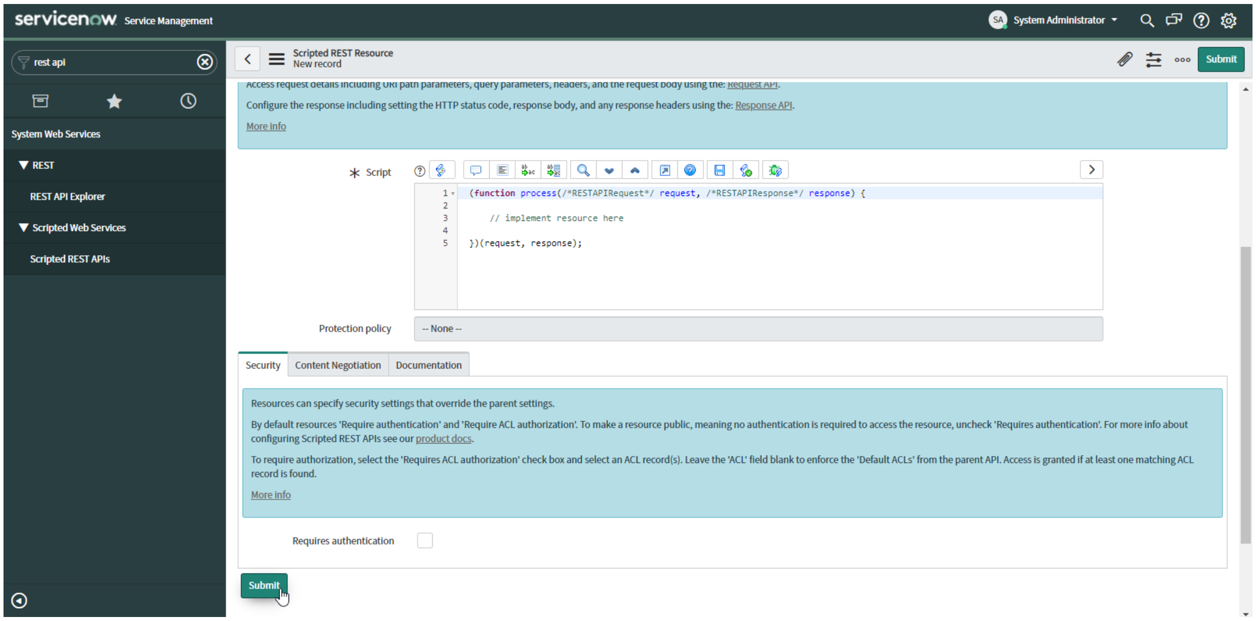
Relative path: /

Implement the resource

Access request details including URI path parameter Request API.

Configure the response including setting the HTTP s e: Response API.

Uncheck “Requires authentication” and click Submit



Access request details including URI path parameters, query parameters, headers, and the request body using the: Request API.

Configure the response including setting the HTTP status code, response body, and any response headers using the: Response API.

More info

* Script

```
(function process(/*RESTAPIRequest*/ request, /*RESTAPIResponse*/ response) {

    // implement resource here

})(request, response);
```

Protection policy: -- None --

Resources can specify security settings that override the parent settings.

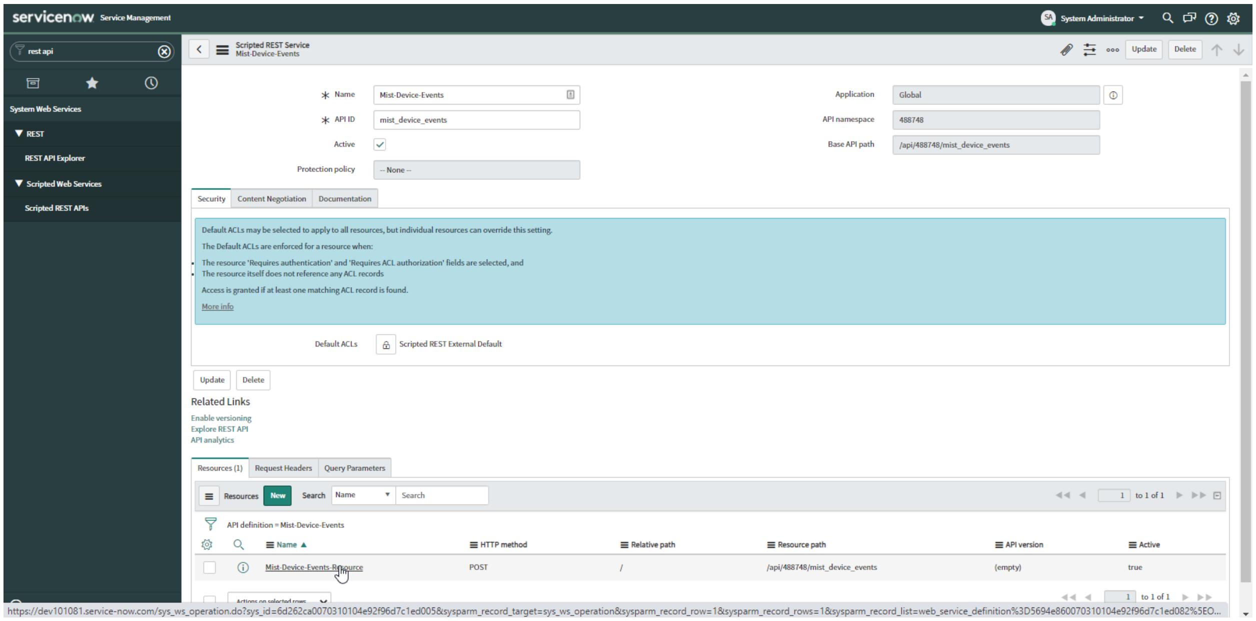
By default resources 'Require authentication' and 'Require ACL authorization'. To make a resource public, meaning no authentication is required to access the resource, uncheck 'Requires authentication'. For more info about configuring Scripted REST APIs see our product docs.

To require authorization, select the 'Requires ACL authorization' check box and select an ACL record(s). Leave the 'ACL' field blank to enforce the 'Default ACLs' from the parent API. Access is granted if at least one matching ACL record is found.

More info

Requires authentication ☐

Navigate back into the new Resource



Scripted REST Service
Mist-Device-Events

Name	Mist-Device-Events	Application	Global
API ID	mist_device_events	API namespace	488748
Active	✓	Base API path	/api/488748/mist_device_events
Protection policy	-- None --		

Default ACLs may be selected to apply to all resources, but individual resources can override this setting.

The Default ACLs are enforced for a resource when:

- The resource 'Requires authentication' and 'Requires ACL authorization' fields are selected, and
- The resource itself does not reference any ACL records

Access is granted if at least one matching ACL record is found.

More info

Default ACLs: Scripted REST External Default

Related Links

Enable versioning
Explore REST API
API analytics

API definition = Mist-Device-Events

1 to 1 of 1

Name ▲	HTTP method	Relative path	Resource path	API version	Active
Mist-Device-Events-Resource	POST	/	/api/488748/mist_device_events	(empty)	true

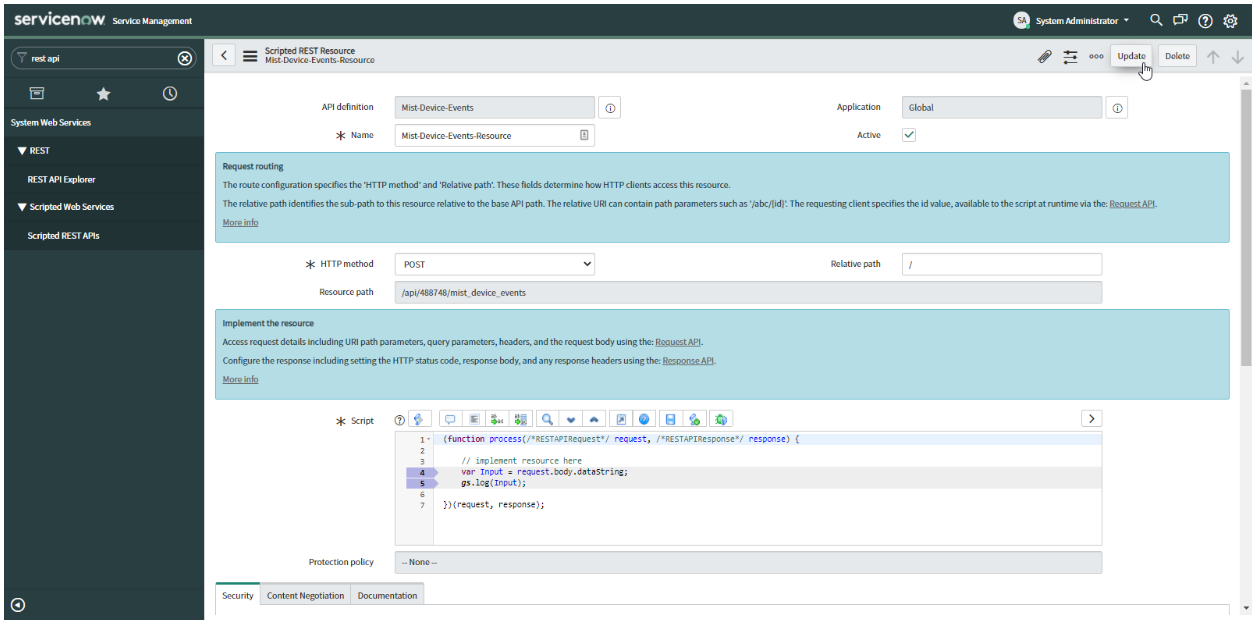
https://dev101081.service-now.com/sys_ws_operation.do?sys_id=6d262ca0070310104e92f96d7c1ed005&sysparm_record_target=sys_ws_operation&sysparm_record_row=1&sysparm_record_rows=1&sysparm_record_list=web_service_definition%3D5694e860070310104e92f96d7c1ed082%5EO...

Make note of the Resource path. In the below example, the resource path is "*/api/488748/mist_device_events*"

Add the following Javascript to the Script section and click Update

```
var Input = request.body.dataString;
gs.log(Input);
```

Note: This code defines variable "Input" with the contents of the webhook as a string and gs.log() sends it to ServiceNow's system logs.



Scripted REST Resource
Mist-Device-Events-Resource

API definition	Mist-Device-Events	Application	Global
Name	Mist-Device-Events-Resource	Active	✓

Request routing

The route configuration specifies the 'HTTP method' and 'Relative path'. These fields determine how HTTP clients access this resource.

The relative path identifies the sub-path to this resource relative to the base API path. The relative URI can contain path parameters such as '/abc/{id}'. The requesting client specifies the id value, available to the script at runtime via the: Request API.

More info

HTTP method	POST	Relative path	/
Resource path	/api/488748/mist_device_events		

Implement the resource

Access request details including URI path parameters, query parameters, headers, and the request body using the: Request API.

Configure the response including setting the HTTP status code, response body, and any response headers using the: Response API.

More info

Script

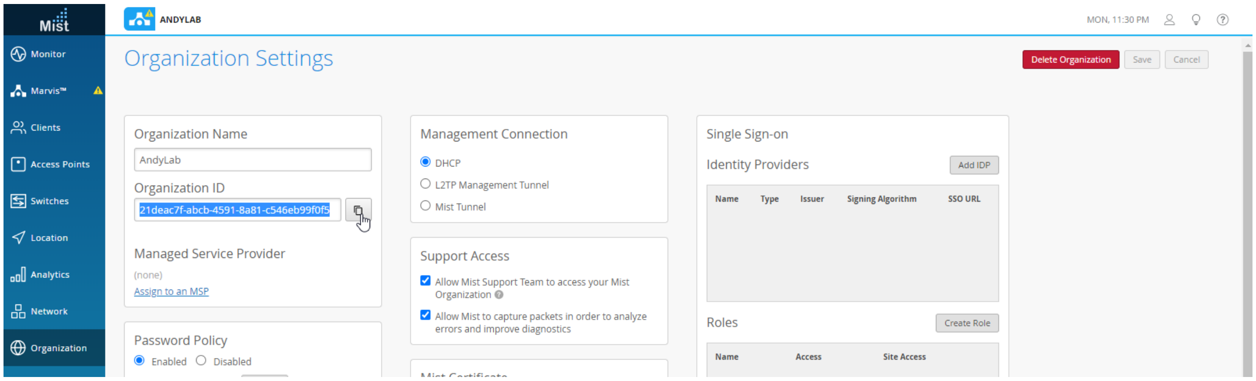
```
(function process(/*RESTAPIRequest*/ request, /*RESTAPIResponse*/ response) {

    // implement resource here
    var Input = request.body.dataString;
    gs.log(Input);

})(request, response);
```

Protection policy: -- None --

This completes the initial setup for webhooks in ServiceNow. The Script section of the request resource is a great place to parse incoming data or create incidents based on the information coming into this resource.

Setting up Webhooks in Mist

Navigate to https://api.mist.com/api/v1/orgs/:org_id/webhooks where ":org_id" = your Mist Organization ID. This can be found by navigating to Organization > Settings



Organization Settings

Organization Name: AndyLab

Organization ID: 21deac7f-abcb-4591-8a81-c546eb99f0f5

Managed Service Provider: (none)
Assign to an MSP

Password Policy: ● Enabled ○ Disabled

Management Connection: ● DHCP ○ L2TP Management Tunnel ○ Mist Tunnel

Support Access
☑ Allow Mist Support Team to access your Mist Organization
☑ Allow Mist to capture packets in order to analyze errors and improve diagnostics

Single Sign-on

Identity Providers

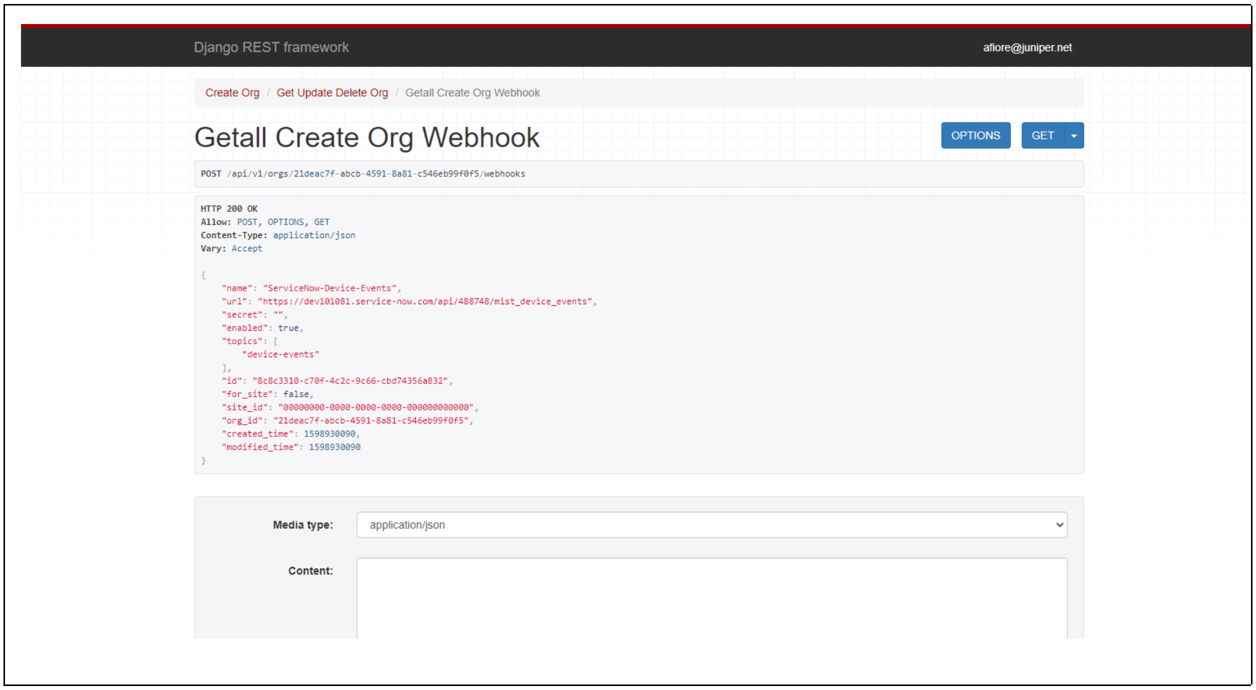
Name	Type	Issuer	Signing Algorithm	SSO URL

Roles

Name	Access	Site Access

Copy the below text into the Content section of the form and replace the section [URL-to-your-ServiceNow-Instance] to the actual location of your ServiceNow instance and add the resource path from ServiceNow press POST.

```
{
    "name": "SNOW Test",
    "url": "https://{{ service_now_instance_url }}/new_resource_path",
    "enabled": true,
    "verify_cert": true,
    "topics": [
        "device-events"
    ]
}
```

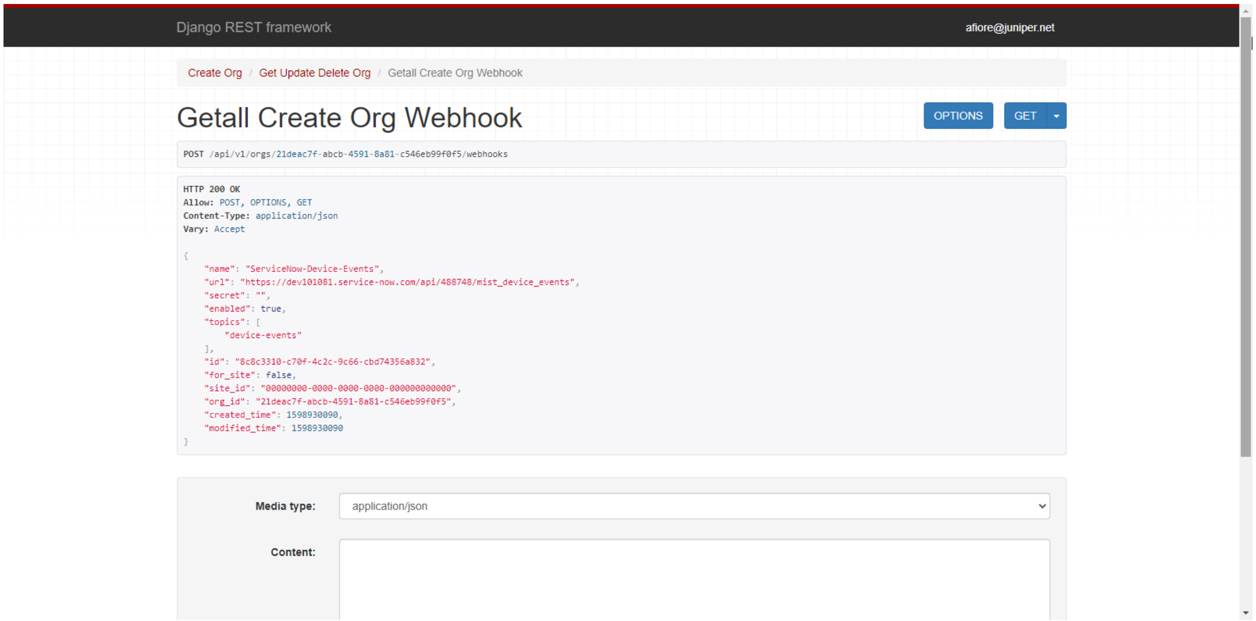


```
POST /api/v1/orgs/21deac7f-abcb-4591-8a81-c546eb99f0f5/webhooks
```

```
HTTP 200 OK
Allow: POST, OPTIONS, GET
Content-Type: application/json
Vary: Accept

{
    "name": "ServiceNow-Device-Events",
    "url": "https://dev101081.service-now.com/api/488748/mist_device_events",
    "secret": "",
    "enabled": true,
    "topics": [
        "device-events"
    ],
    "id": "8c8c3310-c70f-4c2c-9c66-cbd74356a832",
    "for_site": false,
    "site_id": "00000000-0000-0000-0000-000000000000",
    "org_id": "21deac7f-abcb-4591-8a81-c546eb99f0f5",
    "created_time": 1598930090,
    "modified_time": 1598930090
}
```

Media type: application/json

Content:


Django REST framework
afiore@juniper.net
Create Org / Get Update Delete Org / Getall Create Org Webhook
Getall Create Org Webhook
OPTIONS
GET

POST /api/v1/orgs/21deac7f-abcb-4591-8a81-c546eb99f0f5/webhooks

HTTP 200 OK
Allow: POST, OPTIONS, GET
Content-Type: application/json
Vary: Accept

{
    "name": "ServiceNow-Device-Events",
    "url": "https://dev101081.service-now.com/api/488748/mist_device_events",
    "secret": "",
    "enabled": true,
    "topics": [
        "device-events"
    ],
    "id": "8c8c3310-c70f-4c2c-9c66-cbd74356a832",
    "for_site": false,
    "site_id": "00000000-0000-0000-0000-000000000000",
    "org_id": "21deac7f-abcb-4591-8a81-c546eb99f0f5",
    "created_time": 1598930090,
    "modified_time": 1598930090
}

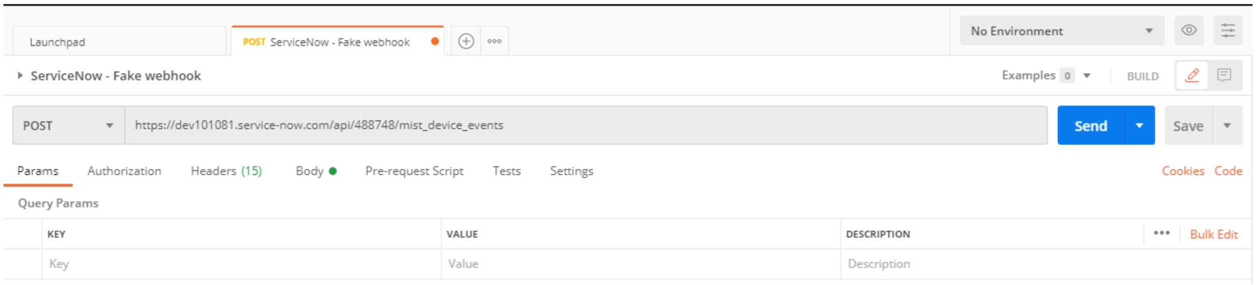
Media type: application/json

Content:

Testing with Postman

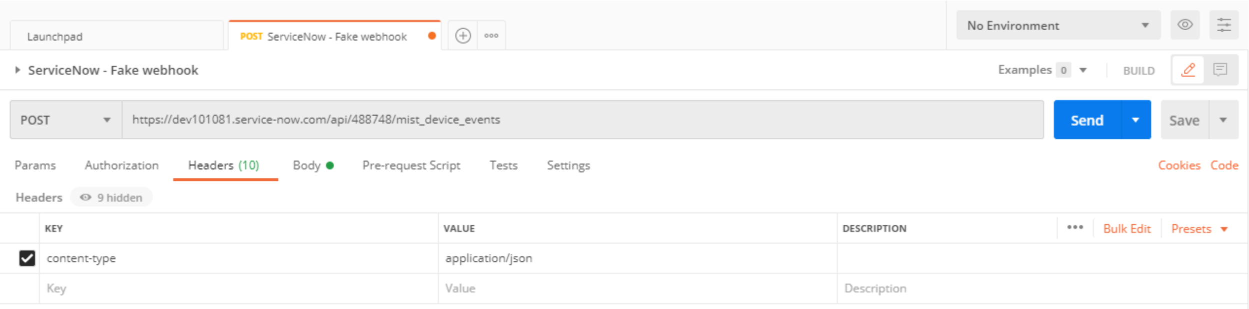
Sending a Fake Webhook to ServiceNow

Create a new request in Postman, select request type as POST and the URL the same as in the webhook you created in Mist.



POST https://dev101081.service-now.com/api/488748/mist_device_events

Click Headers and create a new key named Content-Type and set the value to application/json

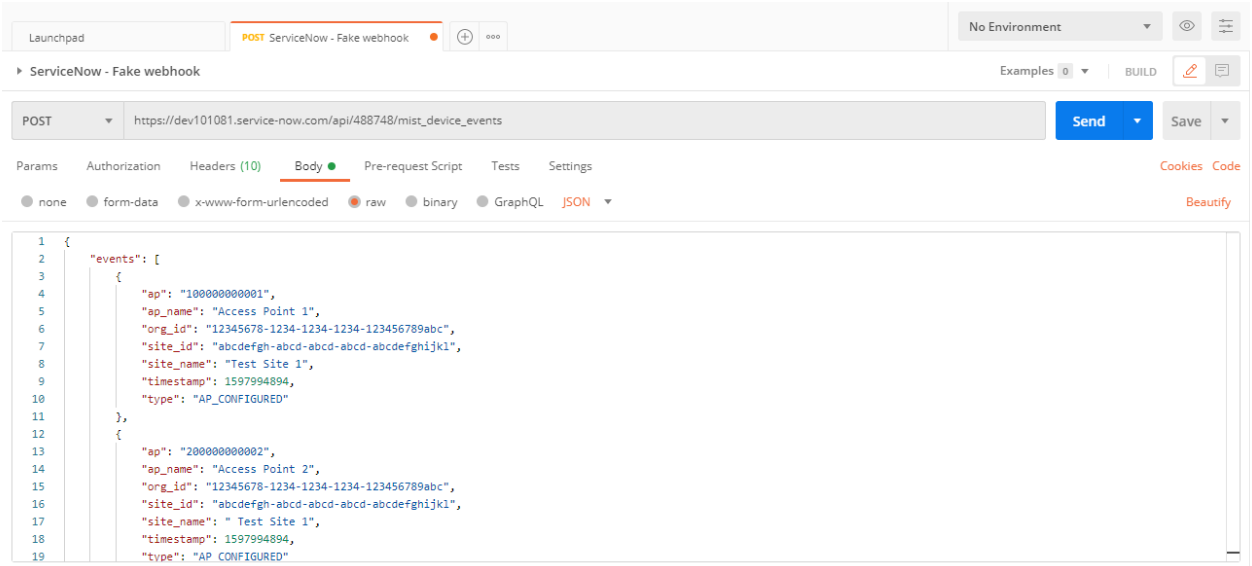


POST https://dev101081.service-now.com/api/488748/mist_device_events

KEY	VALUE	DESCRIPTION
☑ content-type	application/json	

Click Body and select raw. Optionally set highlighting to JSON then copy/paste this information into the body of the request:

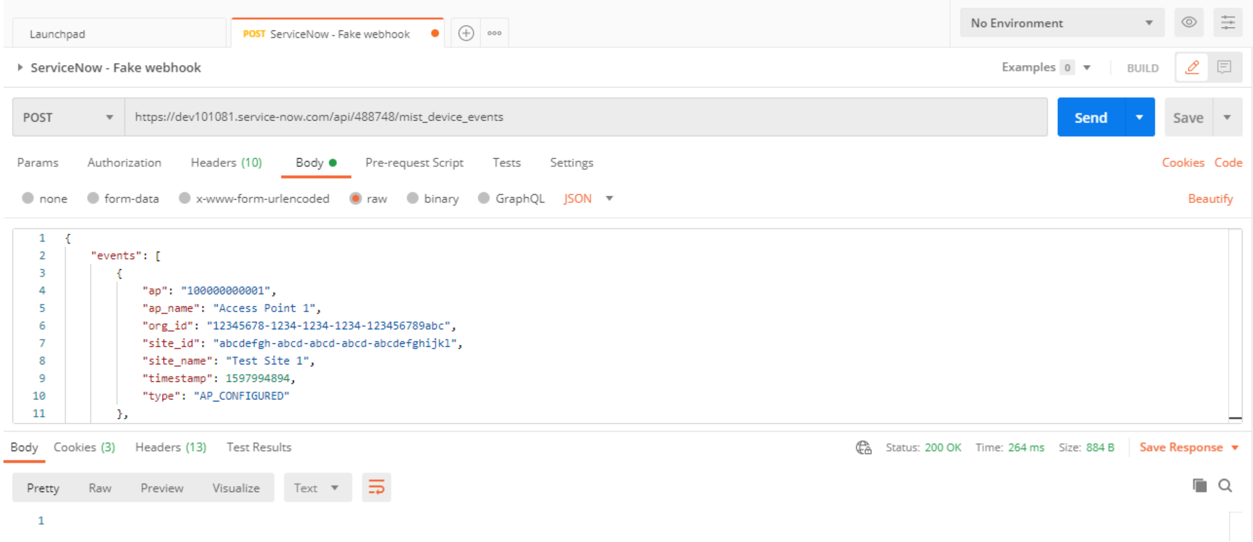
```
{
    “topic”: “device-events”,
    "events": [
        {
            "ap": "100000000001",
            "ap_name": "Access Point 1",
            "org_id": "12345678-1234-1234-1234-123456789abc",
            "site_id": "abcdefgh-abcd-abcd-abcd-abcdefghijkl",
            "site_name": "Test Site 1",
            "timestamp": 1597994894,
            "type": "AP_CONFIGURED"
        },
        {
            "ap": "200000000002",
            "ap_name": "Access Point 2",
            "org_id": "12345678-1234-1234-1234-123456789abc",
            "site_id": "abcdefgh-abcd-abcd-abcd-abcdefghijkl",
            "site_name": " Test Site 1",
            "timestamp": 1597994894,
            "type": "AP_CONFIGURED"
        },
        {
            "ap": "300000000003",
            "ap_name": "Access Point 3",
            "org_id": "12345678-1234-1234-1234-123456789abc",
            "site_id": "abcdefgh-abcd-abcd-abcd-abcdefghijkl",
            "site_name": " Test Site 1",
            "timestamp": 1597994894,
            "type": "AP_CONFIGURED"
        },
        {
            "ap": "400000000004",
            "ap_name": "Access Point 4",
            "org_id": "12345678-1234-1234-1234-123456789abc",
            "site_id": "abcdefgh-abcd-abcd-abcd-abcdefghijkl",
            "site_name": " Test Site 1",
            "timestamp": 1597994894,
            "type": "AP_CONFIGURED"
        }
    ]
}
```



POST https://dev101081.service-now.com/api/488748/mist_device_events

```
{
    "events": [
        {
            "ap": "100000000001",
            "ap_name": "Access Point 1",
            "org_id": "12345678-1234-1234-1234-123456789abc",
            "site_id": "abcdefgh-abcd-abcd-abcd-abcdefghijkl",
            "site_name": "Test Site 1",
            "timestamp": 1597994894,
            "type": "AP_CONFIGURED"
        },
        {
            "ap": "200000000002",
            "ap_name": "Access Point 2",
            "org_id": "12345678-1234-1234-1234-123456789abc",
            "site_id": "abcdefgh-abcd-abcd-abcd-abcdefghijkl",
            "site_name": " Test Site 1",
            "timestamp": 1597994894,
            "type": "AP_CONFIGURED"
```

Click Send and review response



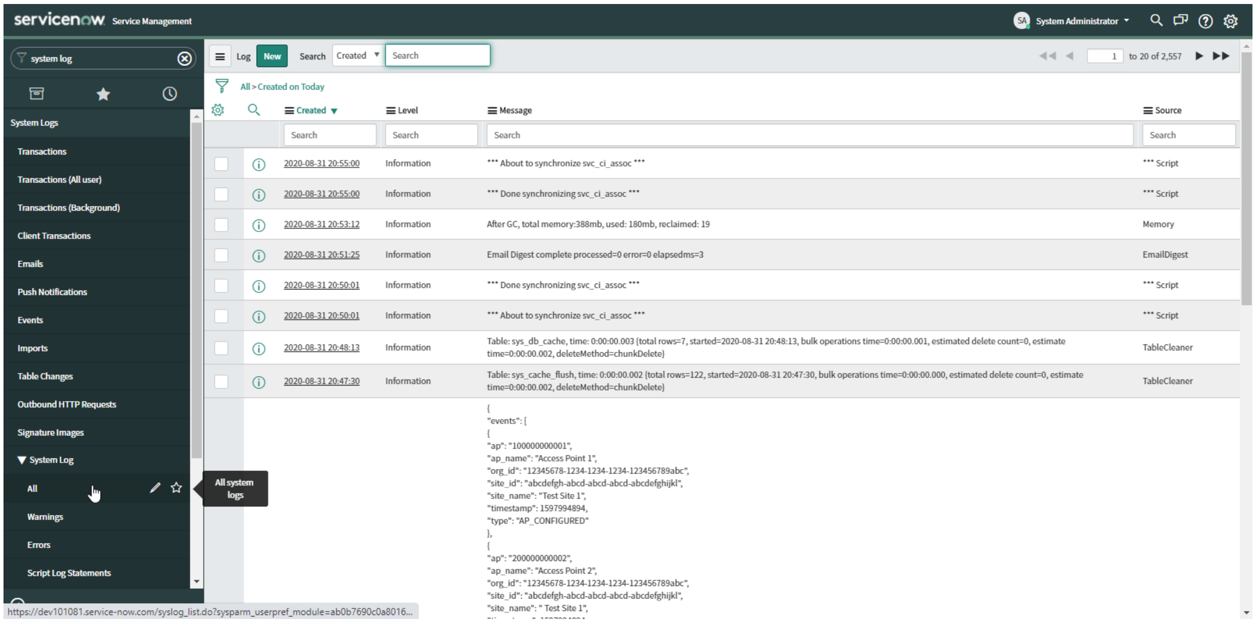
POST https://dev101081.service-now.com/api/488748/mist_device_events

```
{
    "events": [
        {
            "ap": "100000000001",
            "ap_name": "Access Point 1",
            "org_id": "12345678-1234-1234-1234-123456789abc",
            "site_id": "abcdefgh-abcd-abcd-abcd-abcdefghijkl",
            "site_name": "Test Site 1",
            "timestamp": 1597994894,
            "type": "AP_CONFIGURED"
        },
```

Status: 200 OK Time: 264 ms Size: 884 B

Reviewing Webhook Info in ServiceNow

Navigate to System Log > All

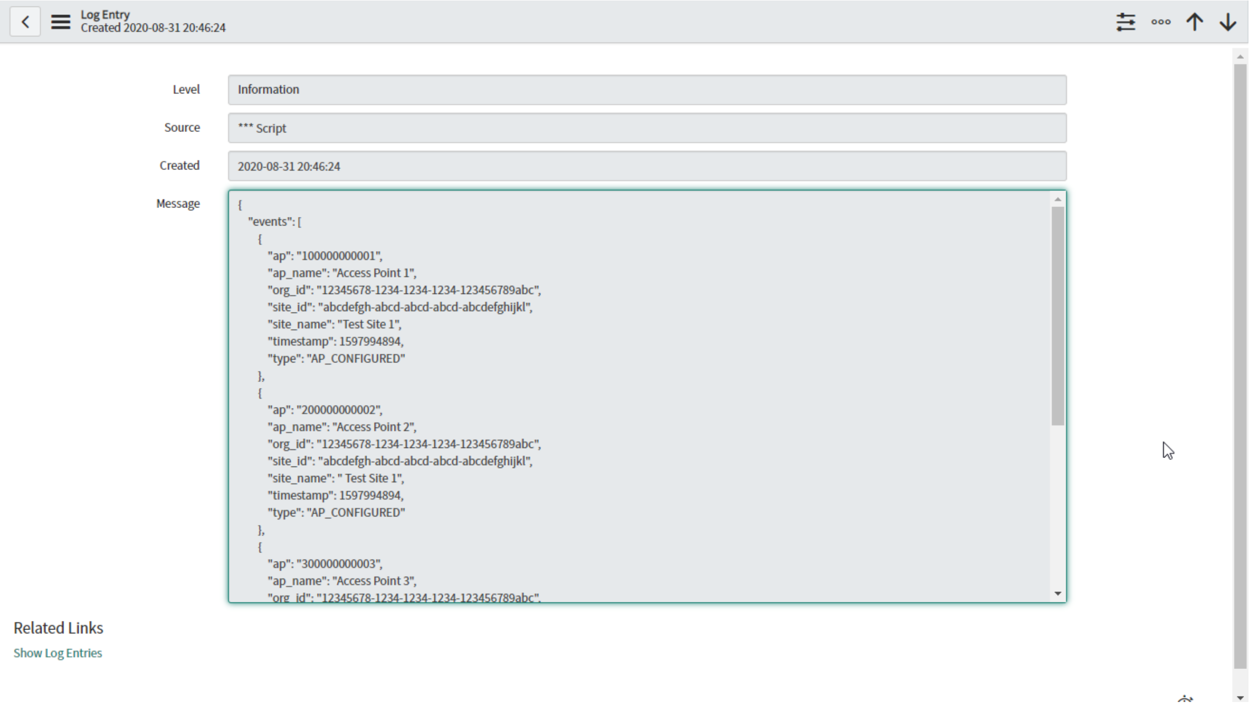


Created	Level	Message	Source
2020-08-31 20:55:00	Information	*** About to synchronize svc_ci_assoc ***	*** Script
2020-08-31 20:55:00	Information	*** Done synchronizing svc_ci_assoc ***	*** Script
2020-08-31 20:53:12	Information	After GC, total memory:388mb, used: 180mb, reclaimed: 19	Memory
2020-08-31 20:51:25	Information	Email Digest complete processed=0 error=0 elapsedms=3	EmailDigest
2020-08-31 20:50:01	Information	*** Done synchronizing svc_ci_assoc ***	*** Script
2020-08-31 20:50:01	Information	*** About to synchronize svc_ci_assoc ***	*** Script
2020-08-31 20:48:13	Information	Table: sys_db_cache, time: 0:00:00.003 {total rows=7, started=2020-08-31 20:48:13, bulk operations time=0:00:00.001, estimated delete count=0, estimate time=0:00:00.002, deleteMethod=chunkDelete}	TableCleaner
2020-08-31 20:47:30	Information	Table: sys_cache_flush, time: 0:00:00.002 {total rows=122, started=2020-08-31 20:47:30, bulk operations time=0:00:00.000, estimated delete count=0, estimate time=0:00:00.002, deleteMethod=chunkDelete}	TableCleaner

```
{
"events": [
{
"ap": "100000000001",
"ap_name": "Access Point 1",
"org_id": "12345678-1234-1234-1234-123456789abc",
"site_id": "abcdefgh-abcd-abcd-abcd-abcdefghijkl",
"site_name": "Test Site 1",
"timestamp": 1597994894,
"type": "AP_CONFIGURED"
},
{
"ap": "200000000002",
"ap_name": "Access Point 2",
"org_id": "12345678-1234-1234-1234-123456789abc",
"site_id": "abcdefgh-abcd-abcd-abcd-abcdefghijkl",
"site_name": " Test Site 1",
```

https://dev101081.service-now.com/syslog_list.do?sysparm_userpref_module=ab0b7690c0a8016...

Payload of Postman request should be shown in the system log



Level: Information

Source: *** Script

Created: 2020-08-31 20:46:24

Message:

```
{
  "events": [
    {
      "ap": "100000000001",
      "ap_name": "Access Point 1",
      "org_id": "12345678-1234-1234-1234-123456789abc",
      "site_id": "abcdefgh-abcd-abcd-abcd-abcdefghijkl",
      "site_name": "Test Site 1",
      "timestamp": 1597994894,
      "type": "AP_CONFIGURED"
    },
    {
      "ap": "200000000002",
      "ap_name": "Access Point 2",
      "org_id": "12345678-1234-1234-1234-123456789abc",
      "site_id": "abcdefgh-abcd-abcd-abcd-abcdefghijkl",
      "site_name": " Test Site 1",
      "timestamp": 1597994894,
      "type": "AP_CONFIGURED"
    },
    {
      "ap": "300000000003",
      "ap_name": "Access Point 3",
      "org_id": "12345678-1234-1234-1234-123456789abc",
```

Related Links

Show Log Entries

ADDITIONAL RESOURCES

ServiceNow Developer Instance
https://developer.servicenow.com/

Mist Documentation
https://api.mist.com/api/v1/docs/Org#org-webhooks

Postman
https://www.postman.com/