gluware®

Getting Started Using GluAPI

Application Note

July 2021

Contents

Overview 3

REST API 4

Introduction 4

Understanding, integrating, and testing API calls 4

Getting Started with GluAPI 7

Ensure your organizations and users are defined 7

Enable the GluAPI endpoint 7

Use Case: Automating Device Manager 8

Get the organization 8

Add a Device 10

Trigger discovery of device details 11

Retrieve device details 12

Conclusion 14

Additional Gluware resources 14

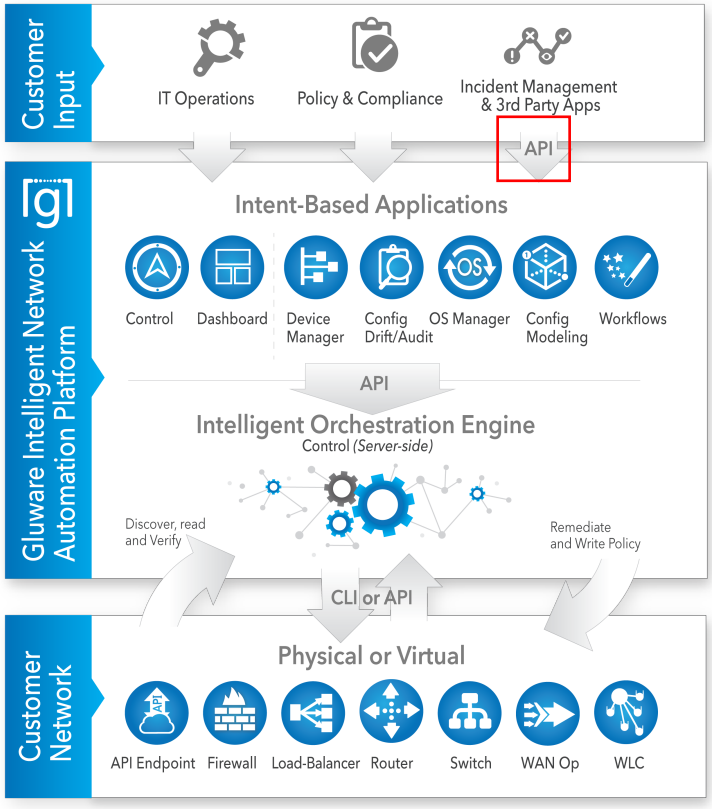
Additional external resources 14

Overview

Gluware is a software platform for intent-based enterprise network automation and orchestration. The Gluware architecture leverages a common intelligent engine and applications built on top to automate the physical or virtual multi-vendor network layer. of the suite of Gluware automation applications is accessed by users through a web user-interface (UI) that drives the orchestration engine through API calls. For customers looking to drive Gluware programmatically, they can drive functions in applications using Gluware's published RESTful API, called GluAPI functions include:

- **Device Manager –** The API provides programmatic capability to add devices to Gluware and trigger the device discovery used to provide a detailed inventory.
- **Config Drift and Audit –** API calls are provided to determine if Config Drift has occurred along with showing any differences between config snapshots, trigger new snapshots, and run configuration audits.
- **Config Modeling –** API calls enable the ability to trigger provisioning of the defined configuration assemblies and related network features.

This Application Note focuses on API calls related to operation performed in the Gluware Device Manager App. GluAPI can also be extended to enable custom data models for more advanced use cases, but that is out of scope for this document.



Gluware high-level architecture enabling the GluAPI to drive Gluware programmatically

REST API

Introduction

APIs (application programming interfaces) in the context of network automation are used to enable end-to-end process automation driven programmatically instead of manually. While DevOps organizations have led the way in using automation to drive CI/CD pipelines to speed application development and delivery, NetOps has fallen behind largely due to the complexity of the underlying network. The network layer (and related domains) is not yet fully programmatic and can require a vendor-specific command-line interface (CLI) to configure individual network devices. Management layer components like ITSM, CMDB, IPAM, SNMP, log servers provide a programmatic API interface.

RESTful (REST-based) APIs have become the de facto standard in the context of network automation. Gluware as an automation and orchestration platform that resides in the management plane and enables a programmatic API interface to then automate the multi-domain, multi-vendor networks using a vendor adaptor layer to talk natively to each vendor in the CLI/semantic required. When considering implementing APIs for programmatic automation, it is important to consider which component is initiating the conversation/request to a REST API endpoint. In this application note will introduce using the Gluware published RESTful API, GluAPI, as a REST endpoint to interact programmatically with Gluware. Gluware can also make API calls to the customer network, or management plane layer, but that is out of scope for this application note.

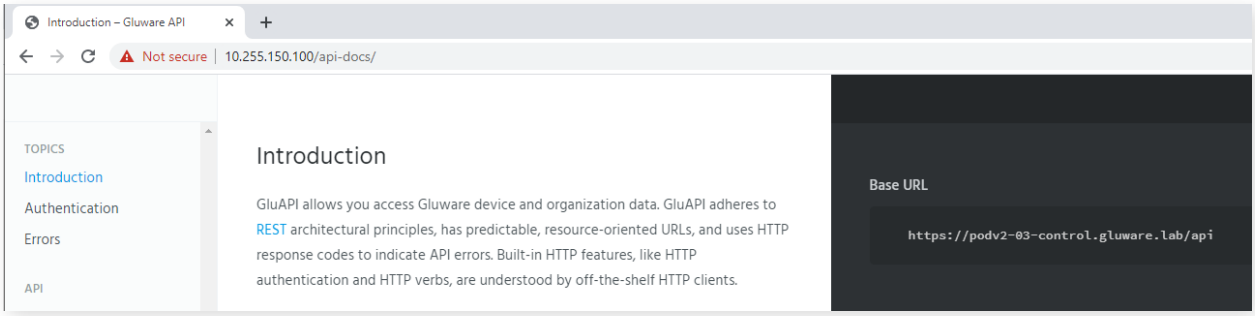
Understanding, integrating, and testing API calls

Define the use case to understand what API calls are required and how they can be implemented. It is important to be familiar with the following seven steps when beginning to use a vendor's API calls and integrate them into the process automation.

1. It's critical to clearly define the use case and have each step well defined and even execute it manually using the vendors user interface. For example, in the first use case we will add a device in Gluware Device Manager. In this example, we will:
    a. Sign in to the Gluware System (requires username/password credentials).
    b. Navigate to the proper organization. Gluware is multi-tenant and each tenant network is called an organization).
    c. Devices can be added by providing a few of the required fields including device name, IP address, and credentials.
2. Review the vendor documentation on the API to determine what calls are available and what format is required. Gluware is REST-based (RESTful), which means it conforms to a standard for the information exchange over the http(s) protocol. Using REST, a

request is made from a client to the URI (Uniform Resource Identifier) of a server (API endpoint) that responds with a payload using a specific format. Gluware uses JSON formatted payload.

The GluAPI documentation is available at https://(your-gluware instance)**/api-docs/**



3. Use the use case and the documentation to determine the API to accomplish the use case. For example, in the use case defined in Step 1, the goal is to add devices via the REST API. For GluAPI, that would require the following call:

   POST https://demo.gluware.com/api/devices

   With the following example body content that has the device details:

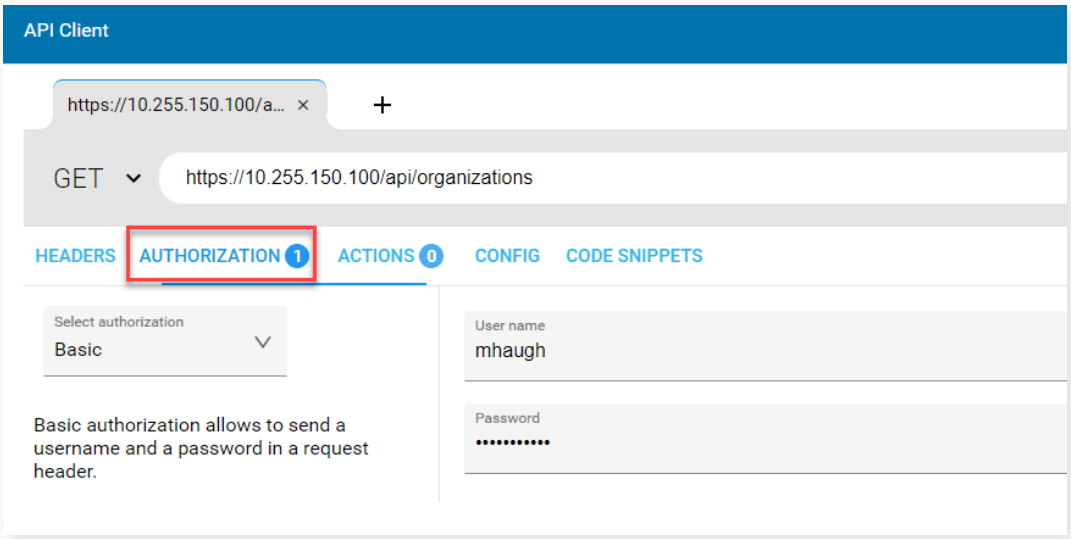
```
{
 "orgId": "457ba514-28c2-4574-87b3-9295e121c2ab",
 "name": "mynewdevice-Cisco",
 "description": "New device created through API",
 "connectionInformation": {
  "ip": "10.93.213.90",
  "userName": "admin",
  "password": "cisco",
  "type": "ssh",
  "port": 22
 }
}
```

   Note – Many of the GluAPI calls require specific variables to be used for the call to complete successfully. Here the proper call is identified. When looking at the required Body content, a variable is required for the Organization ID. This is so the new device is added to the correct organization. An additional call is needed to get the existing Orgs. Then use the variable in the POST operation to add the device.

   The GluAPI call to get the current organizations and their IDs is:

   GET https://demo.gluware.com/api/organizations

4. Understand the authentication/authorization mechanisms used. When enabling programmatic interaction, the platform will enforce Identify and Access Management (IAM) to ensure the proper permissions are enabled to allow the call. Options can included HTTP protocol auth mechanismcs, API Keys, OAuth2, and others. GluAPI uses HTTP protocol authentication mechanisms that includes a username and password in each call. This has to be set up in your automation system to ensure proper authentication for the API calls.



5. Test the API programmatic interaction before moving on to integrate it in the process automation. API toolsets are available such as Postman or Advanced REST Client (ARC), which is used in this app note, to test and exercise each call to understand the details of the input/output parameters and format. These products provide the ability to define the HTTP operation along with the URL, required header parameters, and authentication to define and test any REST API calls. They also provide code snippets in various languages to integrate directly into your implementation.

6. Now that the API calls have been tested and exercised to ensure they perform the desired automated action, its time to integrate. This step is highly dependent on what platform/solution/script you are using to automate the end-to-end process. For some examples this might be a Python based home-grown solution. If you were using Gluware Lab to automate API calls, this uses a JSON and JavaScript based integration. As mentioned in Step 5, the testing tool sets often provide code snippets to minimize the development time.

7. Test, test, test, then deploy to production automation. Ensure your environment supports the ability to fully test the end-to-end process so that each step can be verified. Once tested, ensure you have a process to deliver to production for operational use.

Getting Started with GluAPI

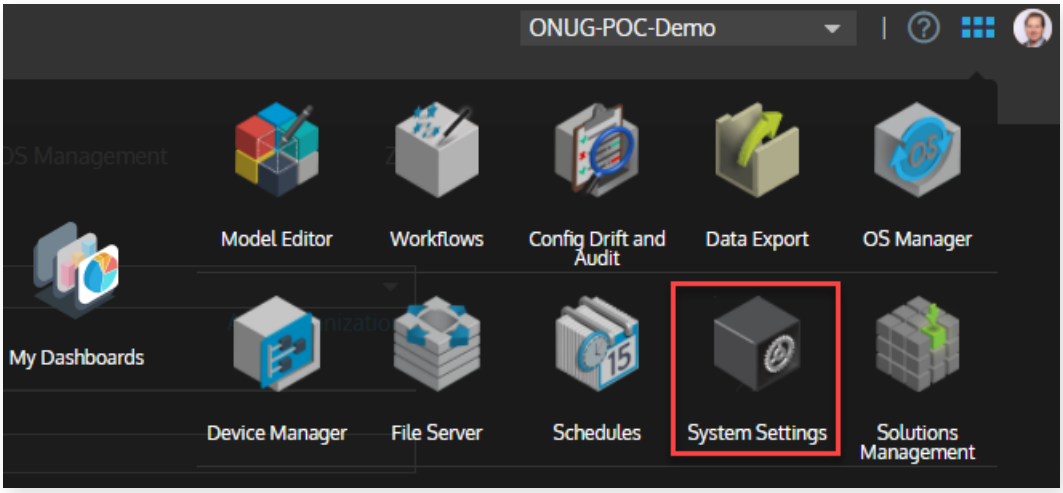
Ensure your organizations and users are defined

Gluware creates a default organization, but it is a best practice to add organizations (tenants) to logically group the devices you are automating. Gluware also requires the creation of users and credentials so that identity and access management can be provided. The user credentials used for the API calls must have permissions enabled in the Role-Based Access Control (RBAC) management.

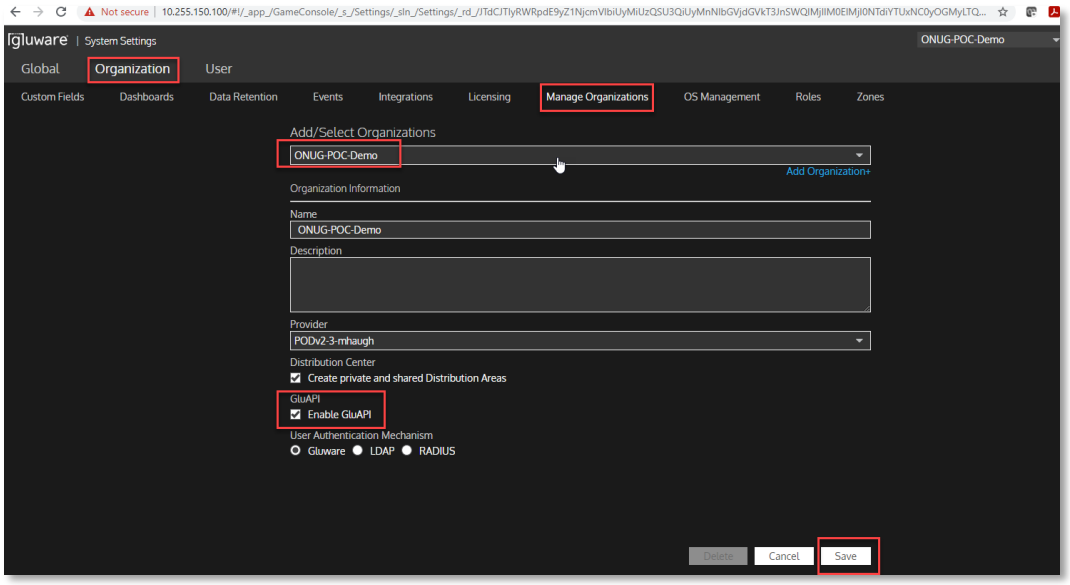
Enable the GluAPI endpoint

The first step in using Gluware via the GluAPI REST endpoint, is to enable it in the platform.

1. Sign in and select **System Settings from the menu**.



2. Go to **Organization** > **Manage Organizations** and ensure the proper organization is selected. Check the box to **Enable GluAPI** and click **Save**.

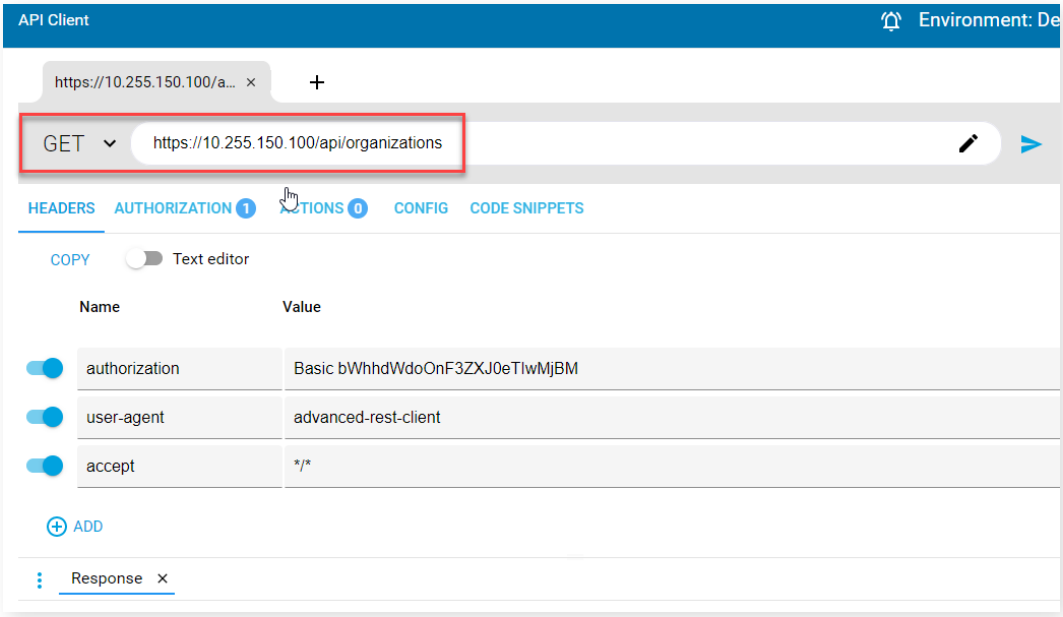


Use Case: Automating Device Manager

In this example, API calls will be used to add network devices and trigger a device discovery. As described earlier, a pre-requisite API call will be used to get the organizations so the proper Org ID can be used in the call.

Get the organization

Use the **GET** call to request the configured organization information. Note that for all the calls used in this example the Authorization is configured with a valid Gluware credential (username/password).

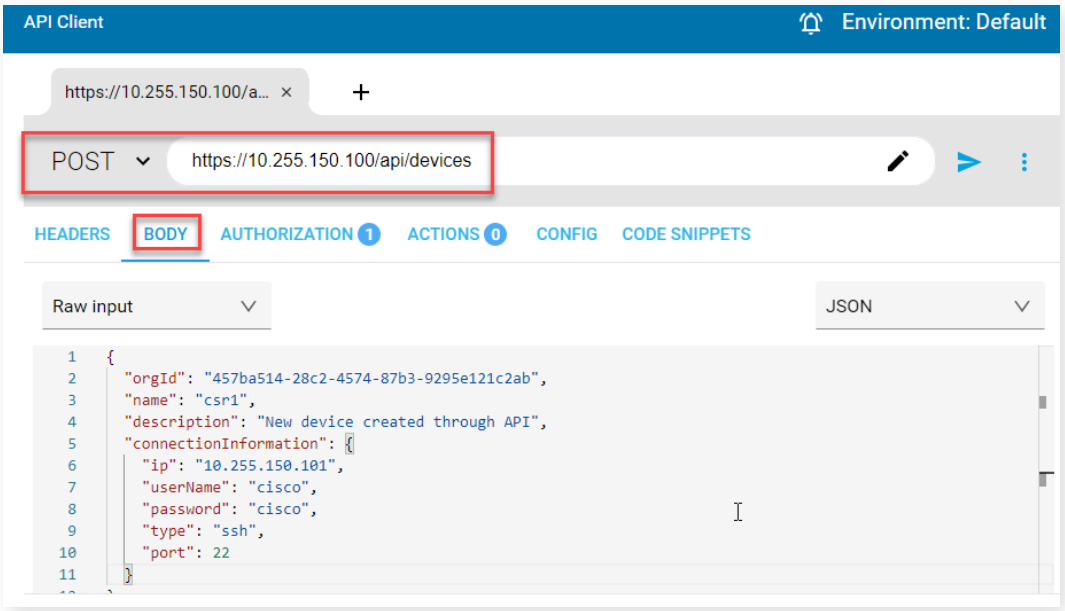


Execute the **GET** call and examine the response to capture the required variable, the Org ID.

Example response from API call to capture the Org ID

Add a Device

Use the **POST** call to add a device.

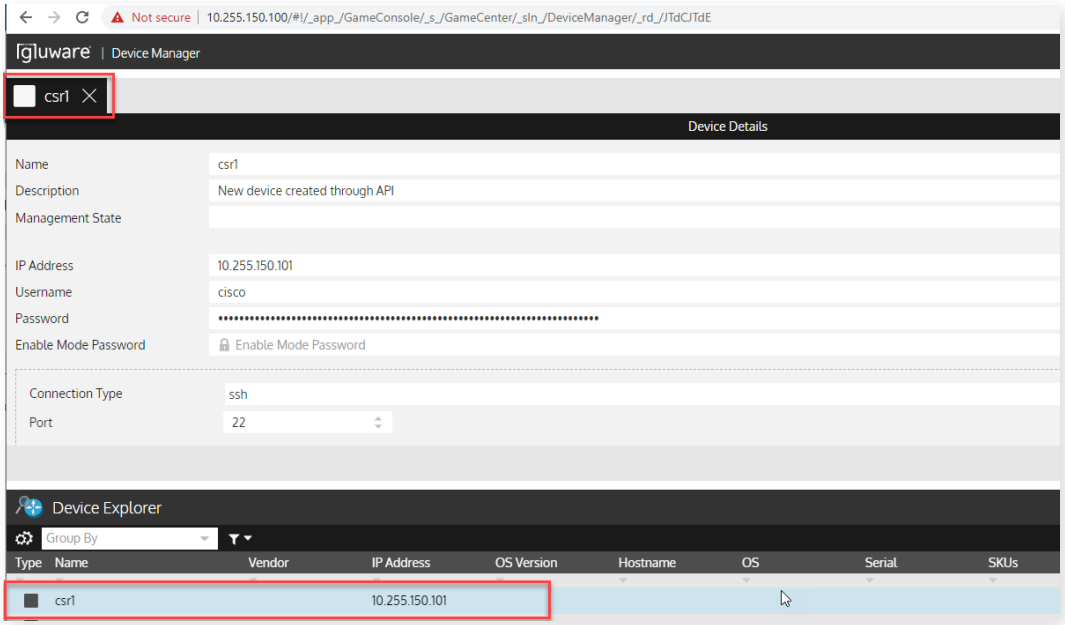


Example **POST** API call to add a device with the required Body content including the Org ID

Once the **POST** call is executed, ensure the response is OK in ARC. The response includes the assigned device ID and creation details.

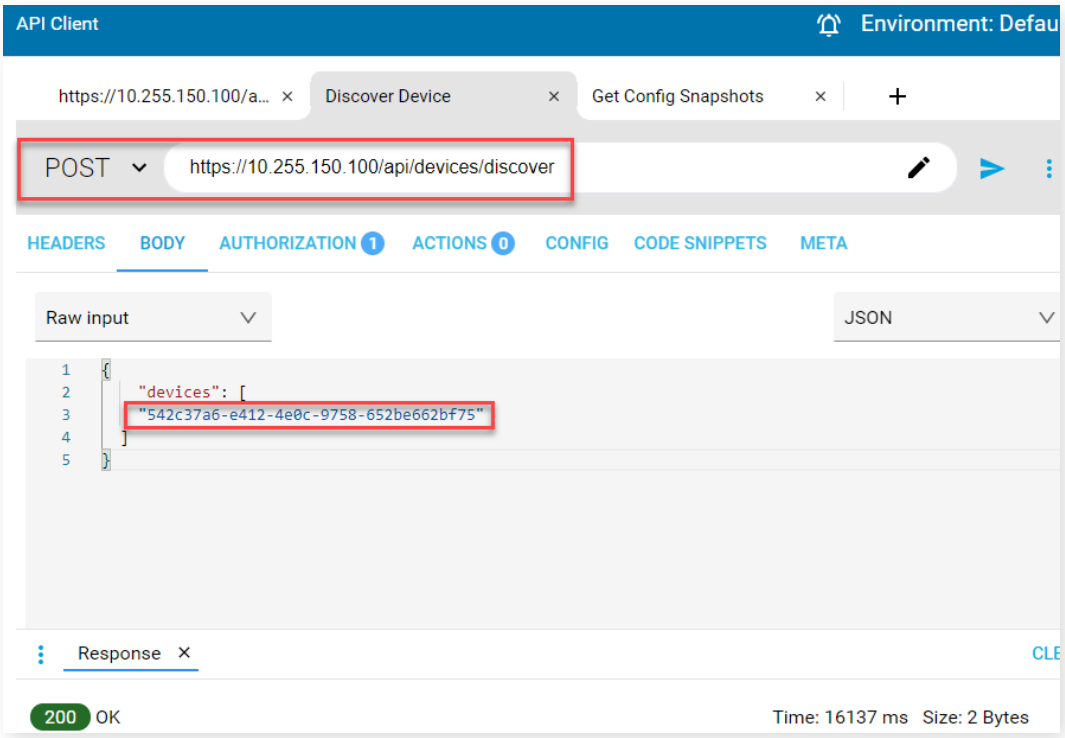
Example response from the **POST** call shows the detailed output

Next, confirm the device was added in Gluware **Device Manager**.



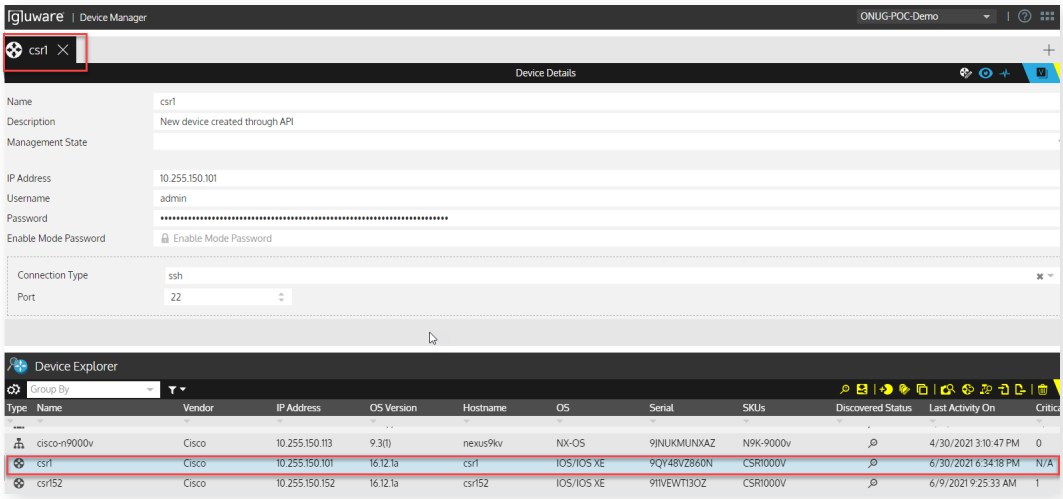
Trigger discovery of device details

Use the **POST** call to trigger device discovery, including the device ID, in the JSON formatted body of the call.



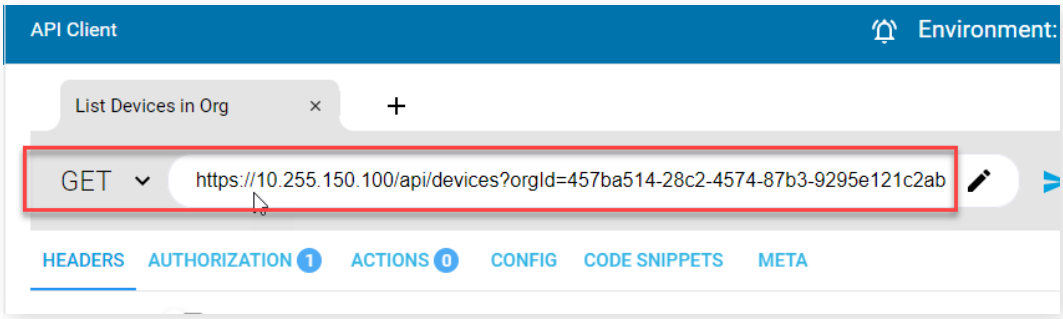
Example POST call to trigger device discovery on the specified device IDs

In **Device Manager**, validate that the discovery operation completed, and the device details are in the grid and the detailed view on top.



Retrieve device details

Use the **GET** call to retrieve the device details. This requires the organization be specified as a ***query parameter*** in the call.



Example GET call to retrieve the device details

The call response will return all the device details in the org, including the newly added csr1 device.

Example response data from the call to retrieve device information in the organization

Conclusion

This application note is intended to help jumpstart the process using GluAPI for programmatic interaction with Gluware via the published REST-based API. Important concepts and pre-requirements are outlined to understand how to enable the API endpoint on the Gluware system and have Organizations and users defined. It also introduces API tools like the ARC client used in this example to test and exercise the API calls before integrating them in a development process. One important takeaway when using the API calls is understanding the syntax of the call and how to define the payload data and query parameters along with the calls to have success. Also, understanding that Gluware assigns IDs to everything created in the system, like the Org IDs, Device IDs, and more, are important components to successfully use specific API calls.

Additional Gluware resources

- Explore the GluAPI product page
- Watch a video demonstrating the GluAPI
- Read a blog about onboarding APIs

Additional external resources

- Learn more about REST
- Check out the REST tool Postman
- Check out the REST tool Advanced REST Client (ARC)

Additional Gluware Material

Tutorials | Collateral | Webinars | Blog posts

gluware®

2020 L Street, Suite 130
Sacramento, CA 95811

www.gluware.com