

# Book of APIs - REST APIs

[ PDF generated August 17 2023. For all recent updates please see the Nutanix Bible releases notes located at [https://nutanixbible.com/release\\_notes.html](https://nutanixbible.com/release_notes.html). Disclaimer: Downloaded PDFs may not always contain the latest information. ]

## REST API Introduction

The REST API exposes every capability and data point of the Prism UI and allows for orchestration or automation tools to easily drive Nutanix action. This enables tools like Saltstack, Puppet, vRealize Operations, System Center Orchestrator, Ansible and more to easily create custom workflows for Nutanix. Also, this means that any third-party developer could create their own custom UI and pull in Nutanix data via REST.

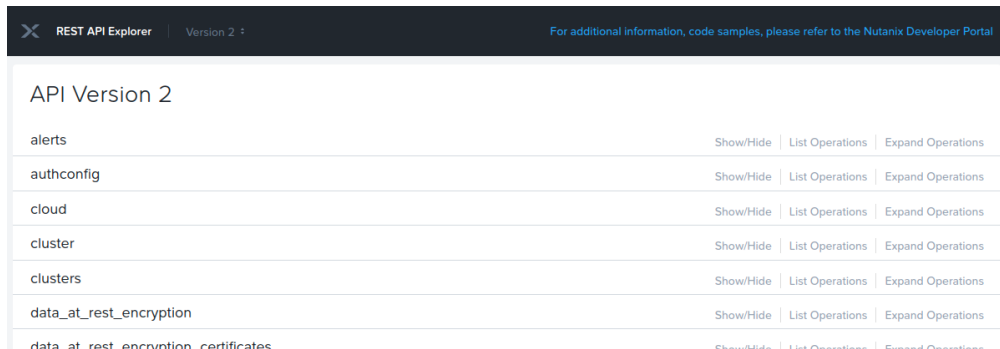
The following figure shows a small snippet of the Nutanix REST API explorer which allows developers to interact with the API and see expected data formats.

## REST API Explorer

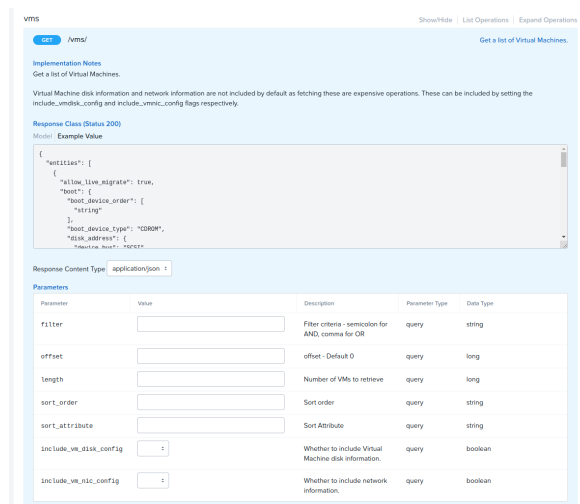
The REST API Explorer is a way of reading API usage info while logged into Prism Element or Prism Central. The API endpoints shown will vary depending on the current login session: Prism Element or Prism Central.

For example, a user currently logged into Prism Central may need to send an API request that creates a virtual machine. The Prism Central v3 **vms** API reference can be quickly viewed in the REST API Explorer.

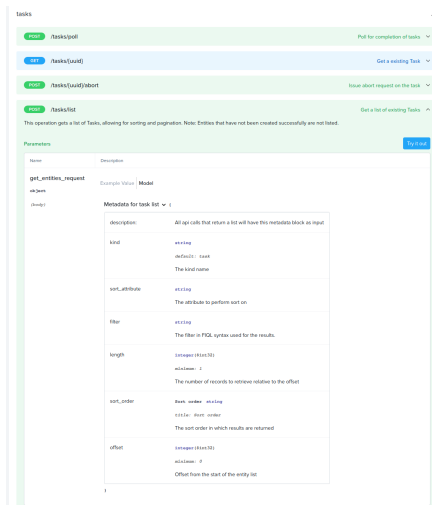
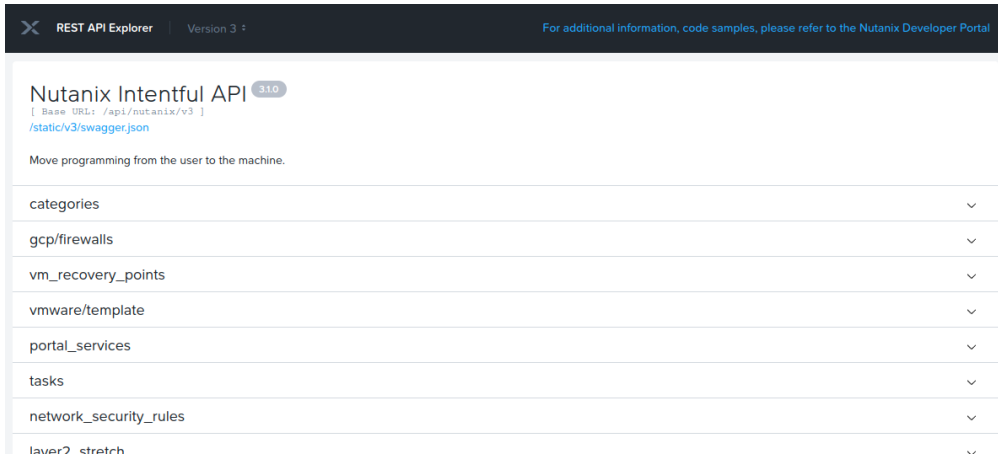
## Prism Element REST API Explorer



Operations can be expanded to display details and examples of the REST call:



# Prism Central REST API Explorer



## REST API Authentication

Nutanix REST APIs support HTTP Basic Authentication for all endpoints. Authentication may be completed using two methods:

1. Read-only: To collect and inspect information exposed by API only.
2. Administrative: All actions available to Read-Only user accounts plus entity management, including but not exclusive to CRUD operations on virtual machines, NCM Self Service Blueprints and Nutanix Flow Network Security policies.

## REST API Versions

Nutanix Prism Element and Prism Central currently offer two generally available APIs:

- v2.0, available in Prism Element only. Used for cluster-local activities such as storage container management operations.
- v3, available in Prism Central only. Used for multicluster activities such as virtual machine management, Nutanix product management including NCM Self Service (formerly Calm), Flow Network Security.

For detailed information on the available API versions and the operations they expose, see [API Versions](#) on [Nutanix.dev](#).

## Nutanix Products Providing APIs

A non-exhaustive list of Nutanix products that can be fully or partially managed by API is as follows. Documentation is linked, where appropriate.

- [Nutanix Prism Element](#)
- [Nutanix Prism Central](#)
- [NCM Self Service \(formerly Calm\)](#)

- [Nutanix Database Service \(formerly Era\)](#)

For a complete list of all API references, see [Nutanix.dev API Reference](#).

## Prism Element v2.0 vs Prism Central v3 APIs

The chosen API will depend on the required operations.

Prism Element v2.0 is cluster-local and available through Prism Element only. It exposes operations specific to the local cluster entities such as storage containers, data-at-rest encryption, storage pools and hosts.

Prism Central v3 is multi-cluster and available through Prism Central only. It exposes operations that can impact multiple clusters such as distributed disk images, NCM Self Service (formerly Calm) blueprints and apps, marketplace items and network security rules.

## Usage Examples: Prism API v2.0

Nutanix Prism API v2.0 is available through Prism Element only. All API requests are as constructed as follows.

```
METHOD https://CVM_OR_CLUSTER_IP:9440/api/nutanix/v2.0/API_NAME/VARIABLES
```

Example of a request to list all storage containers, assuming the cluster IP is 192.168.1.100.

```
GET https://192.168.1.100:9440/api/nutanix/v2.0/storage-containers
```

Example of VM creation:

```
POST https://192.168.1.100:9440/api/nutanix/v2.0/vms
```

Accompanied by VM configuration in JSON format. For example:

```
{
  "description": "VM created by v2.0 API",
  "memory_mb": 1024,
  "name": "vm_api_v2.0",
  "num_vcpus": 1,
  "num_cores_per_vcpu": 1,
  "vm_disks": [
    {
      "is_cdrom": false,
      "vm_disk_create": {
        "size": 128849018880,
        "storage_container_uuid": "b2fefc62-6274-4c84-8e6c-a61f5313ea0e"
      }
    }
  ]
}
```

## Response

The response from this request contains a Prism task UUID that can be queried using the **tasks** API.

```
{
  "task_uuid": "eec7d743-b6c7-4c6d-b821-89b91b142f1d"
}
```

To use this example in your own environment, change **storagecontaineruuid** to a value appropriate for your cluster.

For more information see [Create a Virtual Machine](#) in the Nutanix.dev API reference.

Note: The **/api/nutanix/v2.0** path is an alias for **/PrismGateway/services/rest/v2.0**. The **PrismGateway** path will often be referenced in the Prism API v2.0 documentation; both paths can be used interchangeably and will produce identical results.

## Usage Examples: Prism API v3.0

Nutanix Prism API v3 is available through Prism Central only. All API requests are as constructed as follows.

```
METHOD https://PRISM_CENTRAL_IP:9440/api/nutanix/v3/API_NAME/VARIABLES
```

Example of a request to list all virtual machines, assuming the Prism Central IP is 192.168.1.110.

```
POST https://192.168.1.110:9440/api/nutanix/v3/vms/list
```

Accompanied by an appropriate payload in JSON format. In this example Prism Central is instructed to return all items of type **vm**.

```
{
  "kind": "vm"
}
```

Example of VM creation:

```
POST https://192.168.1.110:9440/api/nutanix/v3/vms
```

Accompanied by VM configuration in JSON format. For example:

```
{
  "spec": {
    "name": "vm_api_v3",
    "resources": {},
    "cluster_reference": {
      "uuid": "0005f2f7-eee7-1995-6145-ac1f6b35fe5e",
      "kind": "cluster"
    }
  },
  "metadata": {
    "kind": "vm"
  }
}
```

The response from this request contains the task status, currently **PENDING**, and task details including the cluster that owns the new VM and a **task\_uuid** that can be queried using the **tasks** API.

```
{
  "status": {
    "state": "PENDING",
    "execution_context": {
      "task_uuid": "1c64cc8b-241d-4132-955f-f6e26239ac02"
    }
  },
  "spec": {
    "name": "vm_api_v3",
    "resources": {},
    "cluster_reference": {
      "kind": "cluster",
      "uuid": "0005f2f7-eee7-1995-6145-ac1f6b35fe5e"
    }
  },
  "api_version": "3.1",
  "metadata": {
    "use_categories_mapping": false,
    "kind": "vm",
    "spec_version": 0,
    "uuid": "d965afdc-4606-4a0b-bc99-f868ae615039"
  }
}
```

To use this example in your own environment, change the cluster **uuid** to a value appropriate for your cluster.

For more information see, [Create a new VM](#) in the Nutanix.dev API reference.