omnissa™

Using Omnissa App Volumes API

December 9, 2024

Release Version: Omnissa App Volumes - 2412

Contents

About This Programming Guide

The *Omnissa App Volumes® API Programming Guide* provides information about the Omnissa App Volumes REST APIs, and how to construct and use the APIs.

Intended Audience

This information is for administrators and programmers who want to configure and manage Omnissa App Volumes programmatically using the available REST APIs.

For information about Omnissa App Volumes documentation, see Omnissa Product Documentation.

Understanding the App Volumes API

To make HTTP requests to the server and to retrieve the required information from the server, Omnissa App Volumes uses REST APIs. You can use the App Volumes APIs to automate workflows within your code.

Overview of REST APIs

REST is an acronym for Representational State Transfer and describes an architectural style characteristic of applications that use the Hypertext Transfer Protocol (HTTP) to exchange serialized representations of objects between a client and a server.

App Volumes API Reference is available at: Omnissa Developer Portal.

Connecting to the App Volumes API

To connect to the App Volumes API and retrieve data from the API, you must have a client which uses the HTTP/URL syntax.

You can use any client that meets this requirement, such as Curl or Postman.

To establish a connection, you must create an authenticated session. After a successful authentication, you receive a session cookie. Note down the session cookie information for future use.

Note: `/app_volumes/version` can be used even when App Volumes Manager is not configured. Hence to use this API, an authenticated session is not necessary.

See your client documentation for specific information about saving and referencing the session cookie.

You can perform the following operations using the App Volumes APIs:

- View System Activity
- Application Operations
    - List Details of an Application
    - List All Available Applications
    - List Assignments of an Application
    - Assign an Application to a Target
    - Remove Application Assignments
    - List All Available Assignments
- Package Operations
    - List Details of a Package
    - List All Available Packages
    - List Assignments of a Package
    - List Programs of a Package
- Writable Volume Operations
    - List Writable Volumes and Datastore information
    - List Details of a Writable Volume
    - Expand a Writable Volume
    - Update a Writable Volume
    - Delete a Writable Volume

App Volumes API Responses

All HTTP responses from Omnissa App Volumes include a status code and, depending on the request, might include a body or a URL. Some responses might be empty.

HTTP Response Codes

The App Volumes API sends the following HTTP response codes when you send an API request.

HTTP Status Codes

Status Code	Status Description
`200 OK`	The request is valid. Some APIs send this code for both a successful response and if there are errors in completing the request. The accompanying message in the document body indicates success or failure.
`400 Missing ID parameter`	The request body is missing key parameters required to complete the request.
`400 Bad session parameters`	Usually indicates a missing or an invalid parameter.
`400 Bad request`	Missing required data.
`500 Server error`	A server error with details provided in the response body.
`401`	User session does not have permissions to run the API.
`404 Resource not found`	Unable to locate the desired volume or AppStack, for example.
`403 Session expired. Create a session and make the request with the _session_id cookie.`	Expired session. User must log in again.
`403 You do not have permission to perform this action.`	The `User` does not have permissions to access the resource.

Getting Started with the API

Before you can begin to use the App Volumes APIs, you must create an authenticated API session and secure a channel between the browser and the App Volumes server.

To create an authentication session for the user, you must first POST a request with a username and password. You can send further API requests only after you receive a successful response.

Create an Authorized API Session

You must create an authorized API session for a user before you can send other API requests.

Prerequisites

To create an authorized session for a user, you must know the user name and password of the user.

Procedure

1. Send the following POST request to the Omnissa App Volumes server with the `username` and `password` in the request body.

   ```
   POST AV_Manager_URL/app_volumes/sessions
   ```

   You can provide the `username` in one of these formats: `accountname`, `NETBIOS_DOMAIN_NAME\accountname`, or `UPN (accountname@DOMAIN)`.

   For example:

   ```
   {
     "username": "username",
     "password": "goodcomplexpassword"
   }
   ```

2. Examine the response.

   A successful request returns a session cookie in the response header and an HTTP status 200 with the following message:

   ```
   {
     "success": "ok"
   }
   ```

   If there is an error, a 400 HTTP status code is returned with an error message. You might get an error if one of the following is true:

   - If the required parameters are missing.
   - If the account for which you are trying to create an authentication does not have sufficient privileges.
   - If the username or password is invalid. For example: When the `username` is not provided, the following is displayed:

```
{
  "error": "User name is required"
}
```

What to do next

Depending on the client you are using to connect to the App Volumes API, the session cookie might have to be saved and referenced for other API requests.

Delete an Authorized API Session

Delete an authorized session for a user.

Prerequisites

To delete the session for a user, you must know the username of the user.

Procedure

1. Send the following DELETE request to the Omnissa App Volumes server. The username is embedded in the cookie header.

   ```
   DELETE AV_Manager_URL/app_volumes/sessions
   ```

2. Examine the response.

   A successful request returns HTTP status 200 and a success message.

   For example:

   ```
   {
     "success": "Destroying session for \"USERNAME\""
   }
   ```

General App Volumes APIs

This section lists the general Omnissa App Volumes APIs.

View System Activity

Retrieve the activity logs to view system activity.

The Activity Log contains information about user logins, computer power-ups, and volume attachments. System messages include messages and errors generated from internal events such as polling for domain controllers, Active Directory access, and so on.

You can also filter the logs by username. Look up the **Down-level Logon Name** of the user in the Active Directory services and find the username.

Procedure

1. Make a GET request to retrieve the activity logs.

   ```
   GET AV_Manager_URL/app_volumes/activity_logs
   ```

2. Examine the response.

   A successful response returns a list of records with HTTP status code 200.

   For example:

   ```
   {
     "actlogs": {
       "logs": [
         {
           "source_id": 0,
           "source_type": "User",
           "source_name": "Administrator",
           "source_url": "/directory#/Users/1",
           "target_id": 0,
           "target_type": "Snapvol",
           "target_name": "Administrator",
           "target_description": "Administrator",
           "target_url": "/directory#/Users/1",
           "action": "Assign",
           "log": "Success",
           "event_time": "2022-01-01 12:00:00 -0700",
           "event_time_human": "January 1, 2022 12:00PM",
           "admin_user_upn": "DOMAIN\\Administrator",
           "admin_user_name": "Administrator",
           "admin_user_url": "/directory#/Users/2",
           "id": 1
         }
       ]
     }
   ```

```
}
```

View Version Details of App Volumes Manager

Use this API to view the version details of App Volumes Manager.

In addition to the version details, you can also view other information such as copyright, version number of App Volumes Manager in the internal format, configuration status of the App Volumes Manager, offset in time between the local time zone and UTC, uptime (duration for which the App Volumes Manager has been running), and the UUID of the database configured in App Volumes Manager.

You can use this API even when App Volumes Manager is not configured. An authenticated session is not required to run this API.

Procedure

1. Make the following `GET` request to the Omnissa App Volumes server:

   ```
   GET AV_Manager_URL/app_volumes/version
   ```

2. Examine the response.

   A successful response returns HTTP status `200` and includes the version, copyright, internal (version of App Volumes Manager in the internal format), configured status, uptime, offset in time between the local time zone and UTC, and database UUID.

   For example:

   ```
   {
     "version": {
       "version": "App Volumes 4, version 2203",
       "internal": "4.6.0",
       "copyright": "Copyright 2011-2022",
       "configured": true,
       "time_offset": 5,
       "uptime": "less than a minute",
       "database_uuid": "654723bc-09de-4d9b-806c-9cd1d231a0de"
     }
   }
   ```

List All Available App Links

To retrieve all available App Links in App Volumes Manager, use this API. A successful API response retrieves data such as App Link name, URI, entry point, app run command-line, all the application packages created for the application, and so on.

You can also choose to list the App Links of a specific page using the pagination parameters such as `page[number]` and `page[size]`.

All parameters except the `api_version` parameters are optional. The `api_version` parameter is mandatory.

- **`page[number]`**

This parameter indicates the page number for which the App Links data must be listed.

Note: Only one page number can be provided as the parameter value at a time.

The default value is `1`.

- **`page[size]`**

  This parameter indicates the number of App Link records listed on a page.

  The default value is `10`.

- **`api_version`**

  This parameter indicates the version of the REST API used in App Volumes.

  The value of the parameter must be `4110`.

- **`include`**

  This parameter fetches relationships of each App Link such as the application (`app_product`) and the packages (`app_packages`).

  The values supported by this parameter are as follows: `app_product` and `app_packages`.

  - **relationships**

    Lists the relationships (application and packages) per App Link.

- **`filter[id]`, `filter[name]`, `filter[product_guid]`, `filter[launch_guid]`**

  The `filter` parameters allow retrieving specific App Links records based on App Link ID, App Link name, application GUID, and application launch ID.

Procedure

1. Make a GET request to obtain the App Links.

   ```
   GET AV_Manager_URL/app_volumes/app_links?api_version=4110
   ```

   Note: If no pagination parameters are specified in the GET request, the first ten App Link records are fetched.

2. Examine the response.

   A successful response returns the available App Links data with HTTP status 200. If the `include` parameter is provided in the GET request, the response displays the `included` object.

   In the following example, App Links are fetched for App Link id `1`.

   ```
   {
     "data": [
       {
         "id": 1,
         "type": "app_links",
   ```

```
      "links": {
        "self": "http://localhost:3000/app_volumes/app_links/1"
      },
      "attributes": {
        "name": "string",
        "launch_uri": "applink://deliver?app=e57afe60-704d-4fbf-8276-3cdc7
8689b96&launch=7e9e4004-742c-e493-1def-e2ba1332ee07",
        "entry_point": "C:\\ProgramData\\Microsoft\\Windows\\Start Menu\\P
rograms\\Notepad++.lnk",
        "app_run_command": "\"%svagent%svservice\" app run e57afe60-704d-4
fbf-8276-3cdc78689b96 \"C:\\ProgramData\\Microsoft\\Windows\\Start Menu\\P
rograms\\Notepad++.lnk\"",
        "description": "string",
        "created_at": "2021-12-03 13:10:13 -0800",
        "updated_at": "2021-12-03 13:10:13 -0800"
      },
      "relationships": {
        "app_product": {
          "data": {
            "type": "app_products",
            "id": 1
          }
        },
        "app_packages": [
          {
            "data": {
              "type": "app_packages",
              "id": 1
            }
          }
        ]
      }
    }
  ],
  "included": [
    {
      "id": 1,
      "type": "app_packages",
      "attributes": {
        "name": "Notepad++ 7.2.0",
        "guid": "b9421adb-d6cb-47f0-81ba-10718d3a8611",
        "app_product_id": 1,
        "state": "Package",
        "version": "7.2.0",
        "programs_count": 1,
        "operating_systems_count": 1,
        "description": "string"
      }
    }
  ],
  "meta": {
    "total": 18,
    "filtered": 18,
    "page_count": 6
  },
  "links": {
    "first": "http://localhost:3000/app_volumes/app_links?include=app_prod
uct%2Capp_packages&page%5Bnumber%5D=1&page%5Bsize%5D=3",
```

```
    "next": "http://localhost:3000/app_volumes/app_links?include=app_produ
ct%2Capp_packages&page%5Bnumber%5D=2&page%5Bsize%5D=3",
    "last": "http://localhost:3000/app_volumes/app_links?include=app_produ
ct%2Capp_packages&page%5Bnumber%5D=2&page%5Bsize%5D=3"
  }
}
```

- **links**

  Displays the links to the first, next, and last pages of the App Link records.

  - `first` - displays the link to the first page of the App Link records
  - `next` - displays the link to the page next to the page number specified in the GET request (`page[num]`).
  - `last` - displays the link to the last page of the App Link records. If any incorrect pagination parameter is provided, an HTTP error code 400 is returned with the following error message:

```
{
  "errors": [
    {
      "title": "Invalid page value",
      "detail": "0 is not a valid value for number page parameter.",
      "code": 118,
      "status": 400
    }
  ]
}
```

Working with Application APIs

You can perform various operations on applications using the APIs.

You can get information about an application, assign or unassign an application from a target (entity), list all available applications, and so on.

Note: If the `User` belongs to a `Group` which is assigned the Applications Owners role, then the `User` can access applications (`app_product_id`) and other application related information such as assignments, App Links, packages, and, so on only for those applications that are either directly owned by the `User` or the `Group` where the `User` is a member. As a result, a successful API response returns records of only those applications (`app_product_id`) to which the `User` has access. Additionally, the `User` can perform operations such as assign entity, remove application, update packages and, so on only for such applications. For more information about the Applications Owners role and the privileges, see the *Managing Admin Roles* section in the *Omnissa App Volumes 4 Administration Guide* at Omnissa Product Documentation.

List Details of an Application

You can list several details about an application such as the name, GUID, status, number of entities assigned, number of application packages, metadata properties, and so on.

Prerequisites

To list the information for an application, you must be aware of the application ID. To obtain the application ID, use the `GET /app_volumes/app_products` API. For more information about this API, see List All Available Applications.

Procedure

1. Make a GET request with the application ID to obtain all the details of the application:

   ```
   GET AV_Manager_URL/app_volumes/app_products/{id}
   ```

2. Examine the response.

   - A successful response returns the application details with an HTTP status 200.

     For example:

     ```
     {
       "data": {
         "id": 0,
         "name": "Notepad++",
         "guid": "1b38f44f-cb39-46fa-bcac-d72af508ddee",
         "icon": null,
         "assignment_count": 1,
         "description": "This application is best suitable for text file
     s.",
         "app_packages_count": 0,
         "app_packages": [
     ```

```
      {
        "id": 0,
        "name": "Notepad-7.0.1",
        "guid": "14848eb3-169b-4e8a-bcb7-556f1811c08c",
        "app_product_id": 3,
        "lifecycle_stage_id": 1,
        "state": "Package",
        "version": "7.0.1",
        "description": "This package is on its latest version
7.0.1",
        "note": "The Package is best suited for development and HR d
ept.",
        "display_delivery": "Classic",
        "delivery": "classic",
        "capable_of_on_demand": true,
        "status": "enabled",
        "programs_count": 2,
        "icon": "/snapvols/9/Notepad++.7.0.1.png",
        "operating_systems_count": 0,
        "delete_status": "deleting",
        "deleted_at": "2021-12-04 13:10:13 -0800",
        "deleted_at_human": "Dec 04 2021",
        "created_at": "2021-12-03 13:10:13 -0800",
        "created_at_human": "Dec 03 2021",
        "updated_at": "2021-12-03 13:10:13 -0800",
        "updated_at_human": "Dec 03 2021",
        "added_at": "string",
        "added_at_human": "string",
        "attachment_count": 2,
        "type": "AppPackage",
        "format": "AV",
        "path": "appvolumes/packages",
        "filename": "Notepad++.vmdk",
        "enabled": true,
        "writable": false,
        "datastore_name": "datastore1",
        "files_count": 2,
        "total_use_count": 5,
        "provision_uuid": "string",
        "provisioning": false,
        "provision_completed_at": "2021-08-17 10:32:32 +0530",
        "provision_started_at": "2021-08-17 10:30:26 +0530",
        "size_mb": 73,
        "size_human": "83.00 MB",
        "assignment_count": 1,
        "volume_guid": "{4f949610-35c1-4e52-9f71-89d899150007}",
        "snapvol_version_id": 0,
        "mount_prefix": "",
        "mounted_at": "2021-08-17 11:00:32 +0530",
        "template_file_name": "[data.1] appvolumes/packages_template
s/template.vmdk",
        "template_version": "2.10.0.709",
        "missing": false,
        "protected": true,
        "agent_version": "4.5.0.922D",
        "capture_version": "4.0",
        "free_mb": 0,
        "total_mb": 0,
```

```
        "attachment_limit": 2,
        "reachable": true,
        "provision_duration": "string",
        "primordial_os_id": 2,
        "primordial_os_name": "Windows 10 (x64)"
      }
    ],
    "owner_guid": "5F02DE3B-BAB1-4FDB-8065-ABB8711013EF",
    "status": "active",
    "delete_status": "deleting",
    "sync_status": null,
    "sync_message": null,
    "synced_at": "2021-12-03 13:10:13 -0800",
    "synced_at_human": "Dec 03 2021",
    "deleted_at": "2021-12-04 13:10:13 -0800",
    "deleted_at_human": "Dec 04 2021",
    "created_at": "2021-12-03 13:10:13 -0800",
    "created_at_human": "Dec 03 2021",
    "updated_at": "2021-12-03 13:10:13 -0800",
    "updated_at_human": "Dec 03 2021",
    "metadata_sync_at": "2021-12-03 15:10:14 -0800",
    "metadata_sync_at_human": "Dec 03 2021",
    "metadata_properties_updated_at": "2021-12-03 15:10:14 -0800",
    "metadata_properties_updated_at_human": "Dec 03 2021",
    "metadata_version": null
  }
}
```


- If Omnissa App Volumes is unable to locate the application, an HTTP error code 404 is returned with an error message.

  For example:


```
{
  "errors": [
    {
      "title": "Application \"0\" was not found",
      "meta": {
        "manager": {
          "title": "Application \"0\" was not found"
        }
      }
    }
  ]
}
```


List All Available Applications

You can use this API to get a list of all available applications and the information associated with each application such as the ID, name, GUID, icon, description, number of packages for that application, and so on, in your Omnissa App Volumes setup.

Procedure

1. Make a GET request to obtain the list of applications.

```
GET AV_Manager_URL/app_volumes/app_products
```

2. Examine the response.

   A successful response returns an HTTP status 200 with list of available applications and associated information about the application.

   For example:


```
{
  "data": [
    {
      "id": 0,
      "name": "Notepad++",
      "guid": "1b38f44f-cb39-46fa-bcac-d72af508ddee",
      "icon": null,
      "assignment_count": 1,
      "description": "This application is best suitable for text files.",
      "app_packages_count": 0,
      "app_packages": [
        {
          "id": 0,
          "name": "Notepad-7.0.1",
          "guid": "14848eb3-169b-4e8a-bcb7-556f1811c08c",
          "app_product_id": 3,
          "lifecycle_stage_id": 1,
          "state": "Package",
          "version": "7.0.1",
          "description": "This package is on its latest version 7.0.1",
          "note": "The Package is best suited for development and HR dep
t.",
          "display_delivery": "Classic",
          "delivery": "classic",
          "capable_of_on_demand": true,
          "status": "enabled",
          "programs_count": 2,
          "icon": "/snapvols/9/Notepad++.7.0.1.png",
          "operating_systems_count": 0,
          "delete_status": "deleting",
          "deleted_at": "2021-12-04 13:10:13 -0800",
          "deleted_at_human": "Dec 04 2021",
          "created_at": "2021-12-03 13:10:13 -0800",
          "created_at_human": "Dec 03 2021",
          "updated_at": "2021-12-03 13:10:13 -0800",
          "updated_at_human": "Dec 03 2021",
          "added_at": "string",
          "added_at_human": "string",
          "attachment_count": 2,
          "type": "AppPackage",
          "format": "AV",
          "path": "appvolumes/packages",
          "filename": "Notepad++.vmdk",
```

```
          "enabled": true,
          "writable": false,
          "datastore_name": "datastore1",
          "files_count": 2,
          "total_use_count": 5,
          "provision_uuid": "string",
          "provisioning": false,
          "provision_completed_at": "2021-08-17 10:32:32 +0530",
          "provision_started_at": "2021-08-17 10:30:26 +0530",
          "size_mb": 73,
          "size_human": "83.00 MB",
          "assignment_count": 1,
          "volume_guid": "{4f949610-35c1-4e52-9f71-89d899150007}",
          "snapvol_version_id": 0,
          "mount_prefix": "",
          "mounted_at": "2021-08-17 11:00:32 +0530",
          "template_file_name": "[data.1] appvolumes/packages_templates/te
mplate.vmdk",
          "template_version": "2.10.0.709",
          "missing": false,
          "protected": true,
          "agent_version": "4.5.0.922D",
          "capture_version": "4.0",
          "free_mb": 0,
          "total_mb": 0,
          "attachment_limit": 2,
          "reachable": true,
          "provision_duration": "string",
          "primordial_os_id": 2,
          "primordial_os_name": "Windows 10 (x64)"
        }
      ],
      "owner_guid": "5F02DE3B-BAB1-4FDB-8065-ABB8711013EF",
      "status": "active",
      "delete_status": "deleting",
      "sync_status": null,
      "sync_message": null,
      "synced_at": "2021-12-03 13:10:13 -0800",
      "synced_at_human": "Dec 03 2021",
      "deleted_at": "2021-12-04 13:10:13 -0800",
      "deleted_at_human": "Dec 04 2021",
      "created_at": "2021-12-03 13:10:13 -0800",
      "created_at_human": "Dec 03 2021",
      "updated_at": "2021-12-03 13:10:13 -0800",
      "updated_at_human": "Dec 03 2021",
      "metadata_sync_at": "2021-12-03 15:10:14 -0800",
      "metadata_sync_at_human": "Dec 03 2021",
      "metadata_properties_updated_at": "2021-12-03 15:10:14 -0800",
      "metadata_properties_updated_at_human": "Dec 03 2021",
      "metadata_version": null
    }
  ]
}
```


List App Packages for an App Volumes Application

To retrieve the application packages for an Omnissa App Volumes application, use this API. An application can

have multiple packages. A successful API response retrieves package information such as package name, delivery mode, status, and so on.

- **include**

  This parameter fetches relationships of a package such as the marker properties, lifecycle stages, base application package if the retrieved package information is an updated package, application for which the package is created, and the source App Volumes Manager instance (if the package is present on a target App Volumes Manager instance).

  This parameter accepts comma separated values of the relationships of the package. The supported values for relationships are as follows: `app_markers`, `lifecycle_stage`, `base_app_package`, `app_product`, and `source_manager_instance`.

  This is an optional parameter.

Prerequisites

Application ID is provided as the input parameter. To obtain the application ID, use the `GET AV_Manager_URL/app_volumes/app_products` API. For more information, see List All Available Applications.

Procedure

1. Make a GET request to obtain the list of packages for a specific application.

   ```
   GET AV_Manager_URL/app_volumes/app_products/{id}/app_packages
   ```

2. Examine the response.

   A successful response returns the list of packages for the application with HTTP status 200.

   For example:


   ```
   {
     "data": [
       {
         "id": 0,
         "name": "Notepad-7.0.1",
         "guid": "14848eb3-169b-4e8a-bcb7-556f1811c08c",
         "app_product_id": 3,
         "lifecycle_stage_id": 1,
         "state": "Package",
         "version": "7.0.1",
         "description": "This package is on its latest version 7.0.1",
         "note": "The Package is best suited for development and HR dept.",
         "display_delivery": "Classic",
         "delivery": "classic",
         "capable_of_on_demand": true,
         "status": "enabled",
         "programs_count": 2,
         "icon": "/snapvols/9/Notepad++.7.0.1.png",
         "operating_systems_count": 0,
         "delete_status": "deleting",
         "deleted_at": "2021-12-04 13:10:13 -0800",
         "deleted_at_human": "Dec 04 2021",
   ```


```
      "created_at": "2021-12-03 13:10:13 -0800",
      "created_at_human": "Dec 03 2021",
      "updated_at": "2021-12-03 13:10:13 -0800",
      "updated_at_human": "Dec 03 2021",
      "added_at": "2021-12-03 13:10:13 -0800",
      "added_at_human": "Dec 04 2021",
      "attachment_count": 2,
      "type": "AppPackage",
      "format": "AV",
      "path": "appvolumes/packages",
      "filename": "Notepad++.vmdk",
      "enabled": true,
      "writable": false,
      "datastore_name": "datastore1",
      "files_count": 2,
      "total_use_count": 5,
      "provision_uuid": "string",
      "provisioning": false,
      "provision_completed_at": "2021-08-17 10:32:32 +0530",
      "provision_started_at": "2021-08-17 10:30:26 +0530",
      "size_mb": 73,
      "size_human": "83.00 MB",
      "assignment_count": 1,
      "volume_guid": "{4f949610-35c1-4e52-9f71-89d899150007}",
      "snapvol_version_id": 0,
      "mount_prefix": "",
      "mounted_at": "2021-08-17 11:00:32 +0530",
      "template_file_name": "[data.1] appvolumes/packages_templates/templa
te.vmdk",
      "template_version": "2.10.0.709",
      "missing": false,
      "protected": true,
      "agent_version": "4.5.0.922D",
      "capture_version": "4.0",
      "free_mb": 0,
      "total_mb": 0,
      "attachment_limit": 2,
      "reachable": true,
      "provision_duration": "2 minutes",
      "primordial_os_id": 2,
      "primordial_os_name": "Windows 10 (x64)"
    }
  ]
}
```


Make a GET request to obtain the list of packages for a specific application with the `include` parameter.

```
GET AV_Manager_URL/app_volumes/app_products/1/app_packages?include=app_mar
kers,lifecycle_stage
```

In the following example, the `include` parameter is used with two values: `app_markers` and `lifecycle_stage`. As a result, the response displays marker and lifecycle stage information for that package.


```
{
    "data": [
```

```
        {
            "id": 1,
            "name": "vlc",
            "guid": "3325446b-098d-4495-9d62-64e0e8ad5c42",
            "app_product_id": 1,
            "lifecycle_stage_id": 1,
            "state": "Package",
            "version": "2.2.4",
            "description": null,
            "note": null,
            "display_delivery": "Classic",
            "delivery": "classic",
            "capable_of_on_demand": true,
            "status": "enabled",
            "programs_count": 1,
            "icon": "/snapvols/1/VLC_media_player.png",
            "operating_systems_count": 1,
            "delete_status": null,
            "deleted_at": null,
            "deleted_at_human": null,
            "created_at": "2023-05-26 11:36:48 +0530",
            "created_at_human": "May 26 2023",
            "updated_at": "2023-05-26 12:03:02 +0530",
            "updated_at_human": "May 26 2023",
            "added_at": "2023-05-26 11:36:48 +0530",
            "added_at_human": "May 26 2023",
            "attachment_count": 1,
            "type": "AppPackage",
            "format": "AV",
            "path": "494-2/appvolumes/packages",
            "filename": "vlc.vmdk",
            "enabled": true,
            "writable": false,
            "datastore_name": "AV-3",
            "files_count": 1,
            "total_use_count": 3,
            "provision_uuid": null,
            "provisioning": false,
            "provision_completed_at": "2023-05-26 11:56:07 +0530",
            "provision_started_at": "2023-05-26 11:38:17 +0530",
            "size_mb": 193,
            "size_human": "193.00 MB",
            "assignment_count": 0,
            "volume_guid": "{3c4d0693-ada0-4693-a732-aa830ce97241}",
            "snapvol_version_id": null,
            "mount_prefix": null,
            "mounted_at": "2023-05-30T06:10:46Z",
            "template_file_name": "[AV-3] 494-2/appvolumes/packages_templa
tes/template.vmdk",
            "template_version": null,
            "missing": false,
            "protected": true,
            "agent_version": "4.9.4.2R",
            "capture_version": "4.0",
            "free_mb": 20409,
            "total_mb": 20477,
            "attachment_limit": null,
            "reachable": true,
```

```
            "provision_duration": "18 minutes",
            "primordial_os_id": 2,
            "primordial_os_name": "Windows 10 (x64)",
            "app_markers": [
                {
                    "id": 1,
                    "name": "CURRENT",
                    "app_product_id": 1,
                    "app_product_name": "vlc",
                    "app_package_id": 1,
                    "user_id": 1,
                    "user_name": "Administrator",
                    "created_at": "2023-05-26 11:36:28 +0530",
                    "created_at_human": "May 26 2023",
                    "updated_at": "2023-05-26 12:03:02 +0530",
                    "updated_at_human": "May 26 2023",
                    "assignable": "Available"
                }
            ],
            "lifecycle_stage": {
                "id": 1,
                "name": "New",
                "priority": 0,
                "created_at": "2023-05-25T22:42:16+05:30",
                "created_at_human": "May 25 2023",
                "updated_at": "2023-05-25T22:42:16+05:30",
                "updated_at_human": "May 25 2023"
            }
        }
    ]
}
```


If App Volumes is unable to locate the application, an HTTP error code 404 is returned with an error message.


```
{
  "errors": [
    {
      "title": "Application \"0\" was not found",
      "meta": {
        "manager": {
          "title": "Application \"0\" was not found"
        }
      }
    }
  ]
}
```


List App Links for an Application

To retrieve the App Links for an application, use this API. An application can have multiple App Links. A successful API response retrieves data such as App Link name, URI, entry point, app run command-line, all the application packages created for the application, and so on.

To understand about App Links, see the *Launch an App Volumes Application with an App Link* section in the

Omnissa App Volumes 4, version 2306, Administration Guide.

Prerequisites

Application ID is provided as the input parameter. To obtain the application ID, use the `GET AV_Manager_URL/app_volumes/app_products` API. For more information, see List All Available Applications.

Procedure

1. Make a GET request to obtain the list of App Links for a specific application.

   ```
   GET AV_Manager_URL/app_volumes/app_products/{id}/app_links
   ```

2. Examine the response.

   A successful response returns the list of App Links for the application with HTTP status 200.

   For example:

   ```
   {
     "data": [
       {
         "id": 1,
         "name": "Notepad++",
         "launch_uri": "applink://deliver?app=e57afe60-704d-4fbf-8276-3cdc786
   89b96&launch=7e9e4004-742c-e493-1def-e2ba1332ee07",
         "entry_point": "C:\\ProgramData\\Microsoft\\Windows\\Start Menu\\Pro
   grams\\Notepad++.lnk",
         "app_run_command": "\"%svagent%svservice\" app run e57afe60-704d-4fb
   f-8276-3cdc78689b96 \"C:\\ProgramData\\Microsoft\\Windows\\Start Menu\\Pro
   grams\\Notepad++.lnk\"",
         "description": "string",
         "app_packages": [
           {
             "id": 3,
             "name": "string"
           }
         ]
       }
     ]
   }
   ```

   If Omnissa App Volumes is unable to locate the application, an HTTP error code 404 is returned with an error message.

   ```
   {
     "errors": [
       {
         "title": "Application \"0\" was not found",
         "meta": {
           "manager": {
             "title": "Application \"0\" was not found"
           }
   ```

```
        }
      }
    ]
}
```

List Assignments of an Application

List all the assignments for an application. An application assignment is represented by an ID using the `id` parameter. The number of IDs indicate the number of assignments present for this application. For each application assignment, the API also contains the application package name, package id, marker id for the package which has `CURRENT` marker, entity type and related information to which the application is assigned, filters (computer prefix name), and so on.

- **`entities`**

  The parameter indicates the type of entity to which the application is assigned. Entity types are `User`, `Computer`, `Group`, and `Organization Unit (OU)`. If an application has multiple assignments, then all the entities are listed.

- **`filters`**

  If you have assigned an application to a non-computer entity and want to limit the delivery of the application to computers by using a specific prefix of a computer name, then the `filters` parameter can be used. The `value` indicates the prefix of the computer name. This parameter is displayed only when an application is assigned to a non-computer entity type (`User`, `Group`, and `Organizational Unit (OU)`).

Prerequisites

Ensure that you are aware of the application ID. To obtain the application ID, use the `GET AV_Manager_URL/app_volumes/app_products` API. For more information, see List All Available Applications.

Procedure

1. Make a GET request to obtain the list of assignments for a specific application.

   ```
   GET AV_Manager_URL/app_volumes/app_products/{id}/assignments
   ```

2. Examine the response.

   A successful response returns the list of assignments for the application with HTTP status 200.

   For example:

   ```
   {
     "data": [
       {
         "id": 1,
         "description": "This application is best suitable for text files.",
         "app_product_id": 1,
         "app_product_name": "Notepad++",
   ```

```
      "app_package_id": null,
      "app_package_name": null,
      "app_marker_id": 1,
      "app_marker_name": "CURRENT",
      "priority": 0,
      "mount_prefix": "",
      "created_at": "2021-12-03 13:10:13 -0800",
      "created_at_human": "Dec 03 2021",
      "updated_at": "2021-12-03 13:10:13 -0800",
      "updated_at_human": "Dec 03 2021",
      "entities": [
        {
          "id": 1,
          "entity_type": "User",
          "name": "testuser-1",
          "account_name": "testuser-1",
          "upn": "SNAPVOLUMES\\testuser-1",
          "distinguished_name": "CN=testuser-1,OU=Domain Users,DC=Snapvolu
mes,DC=com"
        }
      ],
      "filters": [
        {
          "id": 3,
          "type": "ComputerPrefixFilter",
          "value": "COMP"
        }
      ]
    }
  ]
}
```


If Omnissa App Volumes is unable to locate the application, an HTTP error code 404 is returned with an error message.


```
{
  "errors": [
    {
      "title": "Application \"0\" was not found",
      "meta": {
        "manager": {
          "title": "Application \"0\" was not found"
        }
      }
    }
  ]
}
```


Assign an Application to a Target

Assign an application to an entity (target). Entity types are as follows: `User`, `Group`, `Computer`, or `Organizational Unit (OU)`.

- **delivery**

This parameter indicates the mode of delivery of an application package when assigned to an entity.

- `default` - the Do not deliver for these assignments at startup or login option is turned off in App Volumes Manager

  When this option is turned off, application assignments are delivered to an entity based on the package delivery mode option (`classic`, `on-demand`) which is configured in App Volumes Manager.

  For more information about this feature, see *Understanding Package Delivery Modes in App Volumes* in the *Omnissa App Volumes 4 Administration Guide* at Omnissa Product Documentation.

- `on_trigger` - indicates that an application package is not delivered at user login and computer startup and instead, command-line delivery options can be used to deliver applications in real time.

  For more information about the command-line delivery options, see *Command-line Delivery of Applications in App Volumes* in the *Omnissa 4 App Volumes Administration Guide*.

`delivery` is an optional parameter and by default the value of this parameter is `default`.

For information about the `entities` and `filters` parameters, see List Assignments of an Application.

Prerequisites

You must be aware of the following:

- The ID of the application to which you want to assign an entity.

  For more information about how to obtain the ID of an application, see List All Available Applications.

- The Active Directory path of the entity.
- Depending on the type of assignment (`Package` or `Marker`), the ID of the package or marker respectively.

  For more information about how to obtain the package ID or marker ID, see List Assignments of an Application.

Procedure

1. Make a POST request to the App Volumes server to assign an application.

   ```
   POST AV_Manager_URL/app_volumes/app_assignments
   ```

   For example:

   ```
   {
     "data": [
       {
         "app_product_id": 1,
         "entities": [
           {
             "entity_type": "Group",
             "path": "CN=test1,CN=Users,DC=test,DC=local"
   ```

```
        }
      ],
      "app_package_id": null,
      "app_marker_id": 1,
      "delivery": "default",
      "filters": [
        {
          "type": "ComputerPrefixFilter",
          "value": "COMP"
        }
      ]
    }
  ]
}
```


2. Examine the response.

   A successful request returns an HTTP status code 200.

   The `restricted_app_product_ids` attribute lists the application product IDs (`app_product_id`) to which the `User` has no access. For information about why a `User` might not have access to that application, see the note about Applications Owners role in Working with Application APIs.

   In the following example, Notepad++ is assigned to a single entity.


```
{
  "data": [
    {
      "id": 0,
      "description": "This application is best suitable for text files.",
      "app_product_id": 0,
      "app_product_name": "Notepad++",
      "app_package_id": 0,
      "app_package_name": "Notepad-7.0.1",
      "app_marker_id": 0,
      "app_marker_name": "CURRENT",
      "priority": 0,
      "mount_prefix": "",
      "delivery": "default",
      "created_at": "2021-12-03 13:10:13 -0800",
      "created_at_human": "Dec 03 2021",
      "updated_at": "2021-12-03 13:10:13 -0800",
      "updated_at_human": "Dec 03 2021",
      "filters": [
        {
          "id": 3,
          "type": "ComputerPrefixFilter",
          "value": "COMP"
        }
      ]
    }
  ],
      "restricted_app_product_ids": [
        2
      ]
}
```

If you do not provide the required data in the request body, an HTTP status code 400 is returned.

In the following example, a package is not packaged, or the package is disabled:

```
{
  "errors": [
    {
      "title": "Unable to create assignment. Package must be enabled"
      "meta": {
        "manager": {
          "title": "Unable to create assignment. Package must be enabled"
        }
      }
    }
  ]
}
```

In the following example, one of these parameters `app_package_id` or `app_marker_id` is not valid.

```
{
  "errors": [
    {
      "title": "Unable to save assignment"
      "meta": {
        "manager": {
          "title": "Unable to save assignment"
        }
      }
    }
  ]
}
```

In the following example, the parameters that are provided lead to duplicate creation of an assignment:

```
{
  "errors": [
    {
      "title": "Unable to create duplicate assignment with entity SNAPVOLU
MES\\test-user to the same application"
      "meta": {
        "manager": {
          "title": "Unable to create duplicate assignment with entity SNAP
VOLUMES\\test-user to the same application"
        }
      }
    }
  ]
}
```

In the following example, an administrator has entered an invalid delivery parameter:

```
{
"errors": [
   {
```

```
      "title": "Invalid delivery mode 'custom_mode' passed, it must belong
to: [\"default\", \"on_trigger\"]",
      "meta": {
        "manager": {
          "title": "Invalid delivery mode 'custom_mode' passed, it must be
long to: [\"default\", \"on_trigger\"]"
        }
      }
    }
  ]
}
```

Remove Application Assignments

Remove assignments of an application by using the application assignment ID. You can remove multiple assignments for different applications at the same time by mentioning the application assignment ID for each assignment.

Prerequisites

Ensure that you are aware of the application assignment ID of the assignment that you choose to remove from the application. To obtain this ID, use the `GET AV_Manager_URL/app_volumes/app_products/{id}/assignments` API. For more information about this API, see List Assignments of an Application.

Procedure

1. Make a DELETE request to the Omnissa App Volumes server to unassign an application from one or more entities.

   ```
   DELETE AV_Manager_URL/app_volumes/app_assignments
   ```

   In the following example, application assignment ID `1` is unassigned from the entity:

   ```
   {
     "ids": [
       1
     ]
   }
   ```

2. Examine the response.

   A successful request returns an HTTP status code 200, and a message indicating success or failure. If you do not provide the required data in the request body, an HTTP status code 400 is returned.

   The following example shows that assignment ID `1` is successfully removed from an entity:

   ```
   {
     "data": {
       "deleted": [
         {
   ```

```
        "id": "1"
      }
     ],
     "not_deleted": []
}
```

The following example shows that the assignment ID `1` is not successfully removed:

Some of the scenarios when an unassign operation fails are as follows: the application corresponding to the assignment ID does not exist, the specified assignment ID has an attachment, or the user has no access to the corresponding application.

```
{
   "data": {
    "deleted": [],
    "not_deleted": [
      {
        "id": "1"
      }
    ]
}
```

List All Available Assignments

List all the available assignments in App Volumes Manager. All the records in the **Assignments** tab of the admin UI are listed. You can also choose to list the assignments of a specific page by using the pagination parameters such as `page[number]` and `page[size]`.

If you choose to list page-specific assignments, the following pagination parameters must be used:

- **`page[number]`**

  Indicates the page number for which the assignments must be listed. If you are unaware of this information, you can navigate to the App Volumes Manager admin UI and see the **Assignments** tab.

  Note: Only one page number can be provided as the parameter value at a time.

  The default value is `1`.

- **`page[size]`**

  Indicates the number of assignment records that must be listed on a page.

  The default value is `1`.

- **`api_version`**

  Indicates the version of the REST API used in Omnissa App Volumes.

  The value of this parameter must be `4040`.

The following is an optional parameter:

- **`include`**

Fetches relationships of the assignments such as entity details, computer prefix (for entity type `computer`), and application and packages including those set with the `CURRENT` marker.

The values supported by this parameter are as follows: `app_assignment_entities`, `assignment_filters`, `app_marker`, `app_package`, and `app_product`.

Procedure

1. Make a GET request to obtain the assignments.

   ```
   GET AV_Manager_URL/app_volumes/app_assignments
   ```

   Note: If no pagination parameters are specified in the GET request, all the available assignment records are fetched.

2. Examine the response.

   A successful response returns the available assignments with HTTP status 200.

   When assignments are fetched as per pagination parameters provided in the GET request, some of the response parameters are as follows:

   - **relationships**

     Lists the relationships per application assignment ID for all assignments fetched in the GET request.

     Note: The `included` parameter has the detailed description of the relationship values.

   - **links**

     Displays the links to the first, next, and last pages of the **Assignments** tab in the App Volumes Manager admin UI.

     `next` - links to the page available next to the page number specified in the GET request (`page[num]`).

   In the following example, assignments are fetched for application assignment id `1`. As the pagination parameters were specified in the GET request, this response also includes the relationship values and the links to the application assignment pages for the application assignment id:

   ```
   {
     "data": [
       {
         "id": 1,
         "type": "app_assignments",
         "links": {
           "self": "http://localhost:3000/app_volumes/app_assignments/1"
         },
         "attributes": {
           "app_marker_id": 0,
           "app_product_id": 0,
           "app_package_id": 0,
           "created_at": "2021-12-03 13:10:13 -0800",
           "created_at_human": "Dec 03 2021",
   ```

```
      "updated_at": "2021-12-03 13:10:13 -0800",
      "updated_at_human": "Dec 03 2021",
      "delivery": "default"
    },
    "relationships": {
      "app_product": {
        "data": {
          "type": "app_products",
          "id": 1
        }
      },
      "app_marker": {
        "data": {
          "type": "app_markers",
          "id": 1
        }
      },
      "app_package": {
        "data": {
          "type": "app_packages",
          "id": 1
        }
      },
      "app_assignment_entities": {
        "data": [
          {
            "type": "app_assignment_entities",
            "id": 1
          }
        ]
      },
      "assignment_filters": {
        "data": [
          {
            "type": "assignment_filters",
            "id": 1
          }
        ]
      }
    }
  }
],
"included": [
     {
          "id": "1",
          "type": "app_assignment_entities",
          "attributes": {
              "target_type": "User",
              "target_id": 1
          },
          "relationships": {
              "target": {
                  "data": {
                      "type": "users",
                      "id": "1"
                  }
              }
          }
```

```
        },
        {
            "id": "1",
            "type": "users",
            "attributes": {
                "name": "test-user",
                "last_login": null,
                "email": null,
                "upn": "SNAPVOLUMES\\test-user",
                "account_name": "test-user",
                "uuid": null,
                "distinguished_name": "CN=test-user,OU=Users,OU=SNAPVOLUME
S,DC=local",
                "created_at": "2022-02-03T13:04:30Z",
                "updated_at": "2022-02-10T12:03:43Z"
            }
        },
        {
            "id": "1",
            "type": "app_markers",
            "attributes": {
                "name": "CURRENT",
                "app_product_id": 1,
                "app_package_id": null,
                "created_at": "2022-02-03T13:04:10Z",
                "updated_at": "2022-02-03T13:04:10Z"
            }
        },
        {
            "id": "1",
            "type": "app_products",
            "attributes": {
                "name": "Notepad++",
                "guid": "60414c9d-a47c-4b6c-9f45-30434ced8398",
                "icon": null,
                "created_at": "2022-02-03T13:04:10Z",
                "updated_at": "2022-02-03T13:04:10Z",
                "delete_status": null,
                "status": "active",
                "sync_status": null,
                "synced_at": null,
                "sync_message": null
            }
        },
        {
            "id": 1,
            "type": "app_packages",
            "attributes": {
            "name": "Notepad++ 7.2.0",
            "guid": "b9421adb-d6cb-47f0-81ba-10718d3a8611",
            "app_product_id": 1,
            "state": "Package",
            "version": "7.2.0",
            "programs_count": 1,
            "operating_systems_count": 1,
            "description": "string"
            }
        }
```

```
  ],
  "meta": {
    "total": 18,
    "filtered": 18,
    "page_count": 6
  },
  "links": {
    "first": "http://localhost:3000/app_volumes/app_assignments?include=ap
p_assignment_entities%2Cassignment_filters%2Capp_marker%2Capp_package%2Cap
p_marker.app_package%2Capp_product%2Capp_assignment_entities.target&page%5
Bnumber%5D=1&page%5Bsize%5D=3",
    "next": "http://localhost:3000/app_volumes/app_assignments?include=ap
p_assignment_entities%2Cassignment_filters%2Capp_marker%2Capp_package%2Cap
p_marker.app_package%2Capp_product%2Capp_assignment_entities.target&page%5
Bnumber%5D=2&page%5Bsize%5D=3",
    "last": "http://localhost:3000/app_volumes/app_assignments?include=ap
p_assignment_entities%2Cassignment_filters%2Capp_marker%2Capp_package%2Cap
p_marker.app_package%2Capp_product%2Capp_assignment_entities.target&page%5
Bnumber%5D=6&page%5Bsize%5D=3"
  }
}
```


In the following example, the response is obtained when the GET request contains only the `include` parameter and no pagination parameters. All available assignments are listed. In this example, only two application assignments are available.


```
{
    "data": [
        {
            "id": 1,
            "description": null,
            "app_product_id": 1,
            "app_product_name": "EmEditor",
            "app_package_id": null,
            "app_package_name": null,
            "app_marker_id": 1,
            "app_marker_name": "CURRENT",
            "priority": 0,
            "mount_prefix": "",
            "delivery": "on_trigger",
            "created_at": "2022-02-04 14:40:50 +0530",
            "created_at_human": "Feb 04 2022",
            "updated_at": "2022-02-04 14:40:50 +0530",
            "updated_at_human": "Feb 04 2022",
            "app_marker": {
                "id": 1,
                "name": "CURRENT",
                "app_product_id": 1,
                "app_product_name": "EmEditor",
                "app_package_id": null,
                "created_at": "2022-02-03 18:34:10 +0530",
                "created_at_human": "Feb 03 2022",
                "updated_at": "2022-02-03 18:34:10 +0530",
                "updated_at_human": "Feb 03 2022",
                "assignable": "Available",
                "app_package": null
            },
```

```
            "app_package": null
        },
        {
            "id": 2,
            "description": null,
            "app_product_id": 2,
            "app_product_name": "Microsoft Office",
            "app_package_id": 2,
            "app_package_name": "Office 2019",
            "app_marker_id": null,
            "app_marker_name": null,
            "priority": 0,
            "mount_prefix": "",
            "delivery": "default",
            "created_at": "2022-02-07 18:24:09 +0530",
            "created_at_human": "Feb 07 2022",
            "updated_at": "2022-02-07 18:24:09 +0530",
            "updated_at_human": "Feb 07 2022",
            "app_marker": null,
            "app_package": {
                "id": 2,
                "name": "Office 2019",
                "guid": "357a1ef2-1568-4dac-95be-874b910c6a46",
                "app_product_id": 2,
                "lifecycle_stage_id": 4,
                "state": "Package",
                "version": "16.0.10358.20061",
                "description": null,
                "note": null,
                "display_delivery": "Classic",
                "delivery": "classic",
                "capable_of_on_demand": true,
                "status": "enabled",
                "programs_count": 4,
                "icon": "/snapvols/14/Office_16_Click-to-Run_Localizatio
n_Component.png",
                "operating_systems_count": 1,
                "delete_status": null,
                "deleted_at": null,
                "deleted_at_human": null,
                "created_at": "2022-02-03 18:34:47 +0530",
                "created_at_human": "Feb 03 2022",
                "updated_at": "2022-02-03 18:34:48 +0530",
                "updated_at_human": "Feb 03 2022",
                "added_at": "2022-02-03 18:34:47 +0530",
                "added_at_human": "Feb 03 2022",
                "attachment_count": 0,
                "type": "AppPackage",
                "format": "AV",
                "path": "appvolumes/packages",
                "filename": "Office!20!2019.vmdk",
                "enabled": true,
                "writable": false,
                "datastore_name": "datastore",
                "files_count": 1,
                "total_use_count": 0,
                "provision_uuid": null,
                "provisioning": false,
```

```
                "provision_completed_at": "2021-08-25 10:38:24 +0530",
                "provision_started_at": "2021-08-25 10:31:05 +0530",
                "size_mb": 2343,
                "size_human": "2.29 GB",
                "assignment_count": 0,
                "volume_guid": "{a671f795-fb2c-497a-844f-b152d356ff9d}",
                "snapvol_version_id": null,
                "mount_prefix": null,
                "mounted_at": null,
                "template_file_name": "[datastore] appvolumes/packages_tem
plates/template.vmdk",
                "template_version": null,
                "missing": false,
                "protected": true,
                "agent_version": "4.5.0.932U",
                "capture_version": "4.0",
                "free_mb": 0,
                "total_mb": 0,
                "attachment_limit": null,
                "reachable": true,
                "provision_duration": "7 minutes",
                "primordial_os_id": 2,
                "primordial_os_name": "Windows 10 (x64)"
            }
        }
    ]
}
```

Working with Package APIs

You can perform various operations on packages using the APIs.

You can get information about a package, list all packages of an application, view all the assignments for a specific package of the application, and list the programs present within a package.

Note: If the `User` belongs to a `Group` which is assigned the Applications Owners role, then the `User` can access applications (`app_product_id`) and other application related information such as assignments, App Links, packages, and, so on only for those applications that are either directly owned by the `User` or the `Group` where the `User` is a member. As a result, a successful API response returns records of only those applications (`app_product_id`) to which the `User` has access. Additionally, the `User` can perform operations such as assign entity, remove application, update packages and, so on only for such applications.

For more information about the Applications Owners role and the privileges, see the *Managing Admin Roles* section in the *Omnissa App Volumes 4 Administration Guide* at Omnissa Product Documentation.

List Details of a Package

You can list several details about a package such as the name, ID, application ID to which the package belongs, current state, delivery mode (`on-demand` and `classic`), number of programs in the package, and so on.

Prerequisites

You must be aware of the package ID for which you want to list the information. To obtain the package ID, use the `GET /app_volumes/app_packages` API. This API lists several details about a package including the package ID. For more information, see List All Available Packages.

Procedure

1. Make a GET request with the package ID to obtain all the details of the package.

   ```
   GET AV_Manager_URL/app_volumes/app_packages/{id}
   ```

2. Examine the response.

   - A successful response returns the package details with an HTTP status 200.

     For example:

     ```
     {
       "data": {
         "id": 0,
         "name": "Notepad-7.0.1",
         "guid": "14848eb3-169b-4e8a-bcb7-556f1811c08c",
         "app_product_id": 3,
         "lifecycle_stage_id": 1,
         "state": "Package",
         "version": "7.0.1",
         "description": "This package is on its latest version 7.0.1",
     ```

```
    "note": "The Package is best suited for development and HR dep
t.",
    "display_delivery": "Classic",
    "delivery": "classic",
    "capable_of_on_demand": true,
    "status": "enabled",
    "programs_count": 2,
    "icon": "/snapvols/9/Notepad++.7.0.1.png",
    "operating_systems_count": 0,
    "delete_status": "deleting",
    "deleted_at": "2021-12-04 13:10:13 -0800",
    "deleted_at_human": "Dec 04 2021",
    "created_at": "2021-12-03 13:10:13 -0800",
    "created_at_human": "Dec 03 2021",
    "updated_at": "2021-12-03 13:10:13 -0800",
    "updated_at_human": "Dec 03 2021",
    "added_at": "string",
    "added_at_human": "string",
    "attachment_count": 2,
    "type": "AppPackage",
    "format": "AV",
    "path": "appvolumes/packages",
    "filename": "Notepad++.vmdk",
    "enabled": true,
    "writable": false,
    "datastore_name": "datastore1",
    "files_count": 2,
    "total_use_count": 5,
    "provision_uuid": "string",
    "provisioning": false,
    "provision_completed_at": "2021-08-17 10:32:32 +0530",
    "provision_started_at": "2021-08-17 10:30:26 +0530",
    "size_mb": 73,
    "size_human": "83.00 MB",
    "assignment_count": 1,
    "volume_guid": "{4f949610-35c1-4e52-9f71-89d899150007}",
    "snapvol_version_id": 0,
    "mount_prefix": "",
    "mounted_at": "2021-08-17 11:00:32 +0530",
    "template_file_name": "[data.1] appvolumes/packages_templates/te
mplate.vmdk",
    "template_version": "2.10.0.709",
    "missing": false,
    "protected": true,
    "agent_version": "4.5.0.922D",
    "capture_version": "4.0",
    "free_mb": 0,
    "total_mb": 0,
    "attachment_limit": 2,
    "reachable": true,
    "provision_duration": "string",
    "primordial_os_id": 2,
    "primordial_os_name": "Windows 10 (x64)"
  }
}
```


- If Omnissa App Volumes is unable to locate the package, an HTTP error code 404 is returned with an error message.

For example:

```
{
  "errors": [
    {
      "title": "Incorrect package id 0 passed",
      "meta": {
        "manager": {
          "title": "Incorrect package id 0 passed"
        }
      }
    }
  ]
}
```

Update a Package

You can update properties of a package when you want to associate a package with another application, change the lifecycle stage of the package, specify the package delivery type, and add notes and description.

- **`app_product_id`**

  If you want to move a package to another application, you can use this parameter to specify the application to which you want to associate the package.

  You can also use the `app_product_guid` parameter to specify the application for package move.

  Note: When both parameters are mentioned, `app_product_id` takes precedence.

  To obtain the `app_product_id` or `app_product_guid`, see List All Available Applications.

- **`lifecycle_stage_name`**

  If you want to specify the lifecycle stage of a package, you can use this parameter.

  The lifecycle stages of a package are as follows: `New`, `Tested`, `Published`, and `Retired`.

  You can also use the `lifecycle_stage_id` parameter to specify the lifecycle stage of a package. This parameter value can be obtained from the *Get Lifecycle stages* API. Each ID is associated with one of the lifecycle stages.

  Note: When both parameters are mentioned, `lifecycle_stage_name` takes precedence.

  For information about the `lifecycle_stage_id` or `lifecycle_stage_name`, see List All Lifecycle Stages.

  For more information about the package stages, see *Lifecycle of a Package* in the *Omnissa App Volumes 4 Administration Guide* at Omnissa Product Documentation.

- **delivery**

  Determines the delivery mechanism of an application package to the end user. The supported values for this parameter are `classic` and `on-demand`.

  For information about the package delivery modes, see *Understanding Package Delivery Modes in App Volumes* in the *Omnissa App Volumes 4 Administration Guide* at Omnissa Product Documentation.

Prerequisites

To obtain the package ID whose properties you choose to update, see List All Available Packages.

Procedure

1. Make a PUT request to update the properties of a package.

```
PUT AV_Manager_URL/app_volumes/app_packages/{id}
```

```
{
  "data": {
    "app_product_id": 1,
    "lifecycle_stage_name": "Tested",
    "note": "The Package is best suited for development and HR dept.",
    "name": "Notepad-7.0.1",
    "description": "This package is on its latest version 7.0.1.",
    "delivery": "classic"
  }
}
```

2. Examine the response.
   - A successful response returns the package details with an HTTP status `200`


```
{
  "data": {
    "id": 0,
    "name": "Notepad-7.0.1",
    "guid": "14848eb3-169b-4e8a-bcb7-556f1811c08c",
    "app_product_id": 3,
    "lifecycle_stage_id": 1,
    "state": "Package",
    "version": "7.0.1",
    "description": "This package is on its latest version 7.0.1",
    "note": "The Package is best suited for development and HR dep
t.",
    "display_delivery": "Classic",
    "delivery": "classic",
    "capable_of_on_demand": true,
    "status": "enabled",
    "programs_count": 2,
    "icon": "/snapvols/9/Notepad++.7.0.1.png",
    "operating_systems_count": 0,
    "delete_status": "deleting",
    "deleted_at": "2021-12-04 13:10:13 -0800",
    "deleted_at_human": "Dec 04 2021",
    "created_at": "2021-12-03 13:10:13 -0800",
    "created_at_human": "Dec 03 2021",
    "updated_at": "2021-12-03 13:10:13 -0800",
    "updated_at_human": "Dec 03 2021",
    "added_at": "2021-12-03 13:10:13 -0800",
    "added_at_human": "Dec 04 2021",
    "attachment_count": 2,
```

```
    "type": "AppPackage",
    "format": "AV",
    "path": "appvolumes/packages",
    "filename": "Notepad++.vmdk",
    "enabled": true,
    "writable": false,
    "datastore_name": "datastore1",
    "files_count": 2,
    "total_use_count": 5,
    "provision_uuid": "string",
    "provisioning": false,
    "provision_completed_at": "2021-08-17 10:32:32 +0530",
    "provision_started_at": "2021-08-17 10:30:26 +0530",
    "size_mb": 73,
    "size_human": "83.00 MB",
    "assignment_count": 1,
    "volume_guid": "{4f949610-35c1-4e52-9f71-89d899150007}",
    "snapvol_version_id": 0,
    "mount_prefix": "",
    "mounted_at": "2021-08-17 11:00:32 +0530",
    "template_file_name": "[data.1] appvolumes/packages_templates/te
mplate.vmdk",
    "template_version": "2.10.0.709",
    "missing": false,
    "protected": true,
    "agent_version": "4.5.0.922D",
    "capture_version": "4.0",
    "free_mb": 0,
    "total_mb": 0,
    "attachment_limit": 2,
    "reachable": true,
    "provision_duration": "2 minutes",
    "primordial_os_id": 2,
    "primordial_os_name": "Windows 10 (x64)"
  }
}
```


- If you do not provide the required data in the request body, an HTTP status code `400` is returned.


```
"param is missing or the value is empty: data"
```


- If Omnissa App Volumes is unable to locate the package, an HTTP error code `404` is returned with an error message:


```
{
  "errors": [
    {
      "title": "Incorrect Package id 0 passed",
      "meta": {
        "manager": {
          "title": "Incorrect Package id 0 passed"
        }
      }
    }
  ]
```

```
}
```

List All Available Packages

You can get a list of all available packages and the information associated with each package such as the ID, name, GUID, application ID to which the package belongs, current state of the package, delivery mode (`classic` or `on-demand`), status of the package, number of programs in a package, number of package assignments, and so on.

Procedure

1. Make a GET request to obtain the list of packages.

   ```
   GET AV_Manager_URL/app_volumes/app_packages
   ```

2. Examine the response.

   A successful response returns an HTTP status 200 with list of available applications and associated information about the application.

   For example:

   ```
   {
     "data": [
       {
         "id": 0,
         "name": "Notepad-7.0.1",
         "guid": "14848eb3-169b-4e8a-bcb7-556f1811c08c",
         "app_product_id": 3,
         "lifecycle_stage_id": 1,
         "state": "Package",
         "version": "7.0.1",
         "description": "This package is on its latest version 7.0.1",
         "note": "The Package is best suited for development and HR dept.",
         "display_delivery": "Classic",
         "delivery": "classic",
         "capable_of_on_demand": true,
         "status": "enabled",
         "programs_count": 2,
         "icon": "/snapvols/9/Notepad++.7.0.1.png",
         "operating_systems_count": 0,
         "delete_status": "deleting",
         "deleted_at": "2021-12-04 13:10:13 -0800",
         "deleted_at_human": "Dec 04 2021",
         "created_at": "2021-12-03 13:10:13 -0800",
         "created_at_human": "Dec 03 2021",
         "updated_at": "2021-12-03 13:10:13 -0800",
         "updated_at_human": "Dec 03 2021",
         "added_at": "string",
         "added_at_human": "string",
         "attachment_count": 2,
         "type": "AppPackage",
         "format": "AV",
   ```

```
      "path": "appvolumes/packages",
      "filename": "Notepad++.vmdk",
      "enabled": true,
      "writable": false,
      "datastore_name": "datastore1",
      "files_count": 2,
      "total_use_count": 5,
      "provision_uuid": "string",
      "provisioning": false,
      "provision_completed_at": "2021-08-17 10:32:32 +0530",
      "provision_started_at": "2021-08-17 10:30:26 +0530",
      "size_mb": 73,
      "size_human": "83.00 MB",
      "assignment_count": 1,
      "volume_guid": "{4f949610-35c1-4e52-9f71-89d899150007}",
      "snapvol_version_id": 0,
      "mount_prefix": "",
      "mounted_at": "2021-08-17 11:00:32 +0530",
      "template_file_name": "[data.1] appvolumes/packages_templates/templa
te.vmdk",
      "template_version": "2.10.0.709",
      "missing": false,
      "protected": true,
      "agent_version": "4.5.0.922D",
      "capture_version": "4.0",
      "free_mb": 0,
      "total_mb": 0,
      "attachment_limit": 2,
      "reachable": true,
      "provision_duration": "string",
      "primordial_os_id": 2,
      "primordial_os_name": "Windows 10 (x64)"
    }
  ]
}
```


List App Links for a Package

To retrieve App Links for an application package, use this API. A successful API response retrieves data such as App Link name, URI, entry point, app run command-line, and so on.

To understand about App Links, see the *Launch an App Volumes Application with an App Link* section in the *Omnissa App Volumes 4, version 2306, Administration Guide*.

Prerequisites

Package ID is provided as the input parameter. To obtain the package ID, use the `GET AV_Manager_URL/app_volumes/app_packages` API. For more information about the API, see List All Available Packages.

Procedure

1. Make a GET request to obtain the list of App Links for a package.

```
GET AV_Manager_URL/app_volumes/app_packages/{id}/app_links
```

2. Examine the response.

   A successful response returns the list of App Links data for the package with HTTP status 200.

   For example:

```
{
  "data": [
    {
      "id": 1,
      "name": "Notepad++",
      "launch_uri": "applink://deliver?app=e57afe60-704d-4fbf-8276-3cdc786
89b96&launch=7e9e4004-742c-e493-1def-e2ba1332ee07",
      "entry_point": "C:\\ProgramData\\Microsoft\\Windows\\Start Menu\\Pro
grams\\Notepad++.lnk",
      "app_run_command": "\"%svagent%svservice\" app run e57afe60-704d-4fb
f-8276-3cdc78689b96 \"C:\\ProgramData\\Microsoft\\Windows\\Start Menu\\Pro
grams\\Notepad++.lnk\"",
      "description": "string"
    }
  ]
}
```

   If Omnissa App Volumes is unable to locate the package, an HTTP error code 404 is returned with an error message:

```
{
  "errors": [
    {
      "title": "Incorrect package id 0 passed",
      "meta": {
        "manager": {
          "title": "Incorrect package id 0 passed"
        }
      }
    }
  ]
}
```

List Assignments for a Package

List all the assignments for an application package. An assignment for the package is represented by an ID using the `id` parameter. The number of IDs indicate the number of assignments present for this package. The API contains the package name, package id, entity type and related information to which the application is assigned, filters (computer prefix), and so on.

- **`entities`**

  The parameter indicates the type of entity to which the application package is assigned. Entity types are `User`, `Computer`, `Group`, and `Organization Unit (OU)`. If an application package has multiple assignments, then all the entities are listed.

- **`filters`**

  If you have assigned an application package to a non-computer entity and want to limit the delivery of the package to computers by using a specific prefix of a computer name, then the `filters` parameter can be used. The `value` indicates the prefix of the computer name. This parameter is displayed only when an application package is assigned to a non-computer entity type (`User`, `Group`, and `Organizational Unit (OU)`).

Prerequisites

Ensure that you are aware of the package ID for which you choose to list the assignments. To obtain the package ID, use the `GET AV_Manager_URL/app_volumes/app_packages` API. For more information about the API, see List All Available Packages.

Procedure

1. Make a GET request to obtain the list of assignments for a specific package ID.

   ```
   GET AV_Manager_URL/app_volumes/app_packages/{id}/assignments
   ```

2. Examine the response.

   A successful response returns the list of assignments for the package with HTTP status 200.

   For example:


```
{
  "data": [
    {
      "id": 0,
      "description": "This package is on its latest version 7.0.1",
      "app_product_id": 0,
      "app_product_name": "Notepad++",
      "app_package_id": 0,
      "app_package_name": "Notepad-7.0.1",
      "app_marker_id": 0,
      "app_marker_name": "CURRENT",
      "priority": 0,
      "mount_prefix": "",
      "created_at": "2021-12-03 13:10:13 -0800",
      "created_at_human": "Dec 03 2021",
      "updated_at": "2021-12-03 13:10:13 -0800",
      "updated_at_human": "Dec 03 2021",
      "entities": [
        {
          "id": 1,
          "entity_type": "User",
          "name": "testuser-1",
          "account_name": "testuser-1",
          "upn": "SNAPVOLUMES\\testuser-1",
          "distinguished_name": "CN=testuser-1,OU=Domain Users,DC=Snapvolu
mes,DC=com"
        }
      ],
```

```
      "filters": [
        {
          "id": 3,
          "type": "ComputerPrefixFilter",
          "value": "COMP"
        }
      ]
    }
  ]
}
```

If Omnissa App Volumes is unable to locate the package, an HTTP error code 404 is returned with an error message:

```
{
  "errors": [
    {
      "title": "Incorrect package id 0 passed",
      "meta": {
        "manager": {
          "title": "Incorrect package id 0 passed"
        }
      }
    }
  ]
}
```

List of Programs for a Package

You can get a list of all programs that are present in a specific application package and related information such as name of the publisher, install location of the program, version, package ID for which the program is listed, and so on.

Prerequisites

Ensure that you are aware of the package ID for which you choose to list the programs. To obtain the package ID, use the `GET AV_Manager_URL/app_volumes/app_packages` API. For more information, see List All Available Packages.

Procedure

1. Make a GET request to obtain the list of programs for a specific package ID.

   ```
   GET AV_Manager_URL/app_volumes/app_packages/{id}/programs
   ```

2. Examine the response.

   A successful response returns the list of programs for the package with HTTP status 200.

   For example:

```
{
  "data": [
    {
      "id": 0,
      "name": "Office 16",
      "publisher": "Microsoft Corporation",
      "install_location": "string",
      "version": "16.0.10358.20061",
      "icon": "/snapvols/19/Office_16.png",
      "created_at": "2021-12-03 13:10:13 -0800",
      "created_at_human": "Dec 03 2021",
      "updated_at": "2021-12-03 13:10:13 -0800",
      "updated_at_human": "Dec 03 2021",
      "app_package_id": 0
    }
  ]
}
```


If Omnissa App Volumes is unable to locate the package, an HTTP error code 404 is returned with an error message:


```
{
  "errors": [
    {
      "title": "Incorrect package id 0 passed",
      "meta": {
        "manager": {
          "title": "Incorrect package id 0 passed"
        }
      }
    }
  ]
}
```


List All Lifecycle Stages

Use this API to list all the lifecycle stages of a package with the corresponding lifecycle stage ID. To update the lifecycle stage of a package, you can either use the ID or the name of the lifecycle stage when updating the package properties.

To update package properties, see Update a Package.

Some of the parameters used in this API are as follows:

- **`name`**

  This parameter indicates the lifecycle stage name of a package.

  The stages of a package are as follows: `New`, `Tested`, `Published`, and `Retired`. For more information about the package stages, see *Lifecycle of a Package* in the *Omnissa App Volumes 4 Administration Guide* at Omnissa Product Documentation.

- **`priority`**

  This is a read-only parameter used for displaying the values of the Stage field in the App Volumes

Manager admin UI.

Procedure

1. Make a GET request to obtain all the lifecycle stages of a package:

```
GET AV_Manager_URL/app_volumes/lifcycle_stages
```

2. Examine the response.

   A successful response returns an HTTP status `200` with the list of lifecycle stages of the package.


```
{
    "data": [
        {
            "id": 1,
            "name": "New",
            "priority": 0,
            "created_at": "2022-08-14 23:41:30 +0530",
            "created_at_human": "Aug 14 2022",
            "updated_at": "2022-08-14 23:41:30 +0530",
            "updated_at_human": "Aug 14 2022"
        },
        {
            "id": 2,
            "name": "Tested",
            "priority": 1,
            "created_at": "2022-08-14 23:41:30 +0530",
            "created_at_human": "Aug 14 2022",
            "updated_at": "2022-08-14 23:41:30 +0530",
            "updated_at_human": "Aug 14 2022"
        },
        {
            "id": 3,
            "name": "Published",
            "priority": 2,
            "created_at": "2022-08-14 23:41:30 +0530",
            "created_at_human": "Aug 14 2022",
            "updated_at": "2022-08-14 23:41:30 +0530",
            "updated_at_human": "Aug 14 2022"
        },
        {
            "id": 4,
            "name": "Retired",
            "priority": 3,
            "created_at": "2022-08-14 23:41:30 +0530",
            "created_at_human": "Aug 14 2022",
            "updated_at": "2022-08-14 23:41:30 +0530",
            "updated_at_human": "Aug 14 2022"
        }
    ]
}
```

Working with APIs for Managing App Volumes Integration with Third-party Services

Currently in App Volumes, you can configure the integration between App Volumes Manager and Azure Virtual Desktop. You can use APIs to register an integration between App Volumes Manager and Azure Virtual Desktop and manage (update and delete) existing integrations. In addition to these operations, you can retrieve the list of all integrations, synchronize all integrations at the same time, or perform the retrieve and synchronize operations for just a specific integration.

Alternately, you can also use the App Volumes Manager admin UI for these operations. For more information about integrating App Volumes with third-party services, see the *Integrate App Volumes with Third Party Services* chapter in the App Volumes Administration Guide.

Prerequisites

Ensure that you have read through and performed all actions as mentioned in the *Deploy App Volumes Manager through Azure Marketplace* chapter in the App Volumes Install Guide.

Configure an Integration between App Volumes Manager and Azure Virtual Desktop

Use this API to configure an integration between App Volumes Manager and Azure Virtual Desktop. With this integration, `CURRENT` packages and certain application properties between App Volumes Manager and Azure Virtual Desktop are synchronized.

Note: For an integration, two authorization methods are supported: Managed Identity and Application Registration. Tenant ID, Client ID, and Client Secret are required when using Application Registration.

Parameters

`api_version` is a mandatory parameter to be used with this API. This parameter indicates the version of the REST API used in App Volumes. The value of the parameter must be `4160`.

Prerequisites

Ensure that you are aware of the prerequisites and other related information before configuring this integration. See the *Configure App Volumes Manager Integration with Azure Virtual Desktop* topic in the App Volumes Administration Guide.

Procedure

1. Make a POST request to configure an integration between App Volumes Manager and Azure Virtual Desktop.

   ```
   POST AV_Manager_URL/app_volumes/avd_integrations
   ```

   In the following example, `type`, and attributes such as `name`, `organization`, `group`, `tenant`, `client_id`, `client_secret`, and `integration tags` are provided.

```
{
    "data": {
        "type": "avd_integrations",
        "attributes": {
            "name": "Prod West US Integration",
            "organization": "f68c64fd-870c-4d07-a9f6-b6ff41b76c38",
            "group": "ResourceGroup",
            "tenant": "e59c64fd-870c-4d07-a9f6-b6ff41b76d11",
            "client_id": "d79c64fd-870c-4d07-a9f6-b6ff41b76d18",
            "client_secret": "client_secret_value",
            "integration_tags": {
                "tag_key": "tag_value"
            }
        }
    }
}
```

2. Examine the response.
    - A successful request returns an HTTP status code 200. In this example, Prod West US Integration is the name of the integration configured between App Volumes Manager and Azure Virtual Desktop. The response includes all details of this integration including the relationships (`integration_tags`) and the data of the `integration_tags` is present within the `included` parameter.

```
{
"data": {
    "id": "5F02DE3B-BAB1-4FDB-8065-ABB8711013EF",
    "type": "avd_integrations",
    "links": {
        "self": "http://localhost:3000/app_volumes/...?page%5Bnumber%5
D=1&page%5Bsize%5D=3"
    },
    "attributes": {
        "name": "Prod West US Integration",
        "organization": "f68c64fd-870c-4d07-a9f6-b6ff41b76c38",
        "group": "ResourceGroup",
        "tenant": "e59c64fd-870c-4d07-a9f6-b6ff41b76d11",
        "client_id": "d79c64fd-870c-4d07-a9f6-b6ff41b76d18",
        "credential_id": "118",
        "guid": "1b38f44f-cb39-46fa-bcac-d72af508ddee",
        "configuration_id": null,
        "configuration_message": "Works great!",
        "configuration_status": "Active",
        "configuration_type": "AVD",
        "enabled": true,
        "quota_message": null,
        "quota_status": null,
        "integration_count": 0,
        "status": "active",
        "delete_status": "deleting",
        "synced_at": "2021-12-03 13:10:13 -0800",
        "synced_at_human": "Dec 03 2021",
        "deleted_at": "2021-12-04 13:10:13 -0800",
        "deleted_at_human": "Dec 04 2021",
        "created_at": "2021-12-03 13:10:13 -0800",
        "created_at_human": "Dec 03 2021",
        "updated_at": "2021-12-03 13:10:13 -0800",
```

```
        "updated_at_human": "Dec 03 2021",
        "sync_scheduled_at": "2021-12-03 13:10:13 -0800",
        "sync_scheduled_at_human": "Dec 03 2021"
    },
    "relationships": {
        "integration_tags": {
            "data": [
            {
                "type": "integration_tags",
                "id": 20
            }
            ]
        }
    }
},
"included": [
    {
        "id": 20,
        "type": "integration_tags",
        "attributes": {
            "key": "environment",
            "value": "test",
            "parent_guid": "5F02DE3B-BAB1-4FDB-8065-ABB8711013EF",
            "parent_id": 0,
            "created_at": "2021-12-03 13:10:13 -0800",
            "created_at_human": "Dec 03 2021",
            "updated_at": "2021-12-03 13:10:13 -0800",
            "updated_at_human": "Dec 03 2021"
        }
    }
]
}
```


- If you do not provide the required data in the request, an HTTP status code 400 is returned. For example, if the client secret ID is provided instead of the client secret value in the request, then the following response is fetched:


```
{
    "errors": [
        {
            "title": "Invalid authorizatn parameters",
            "detail": "Unable to authorize application registration
for the specified tenant, subscription and resource group: Failed to
connect ARM. Invalid Client Secret. Ensure the secret being sent in
the request is the client secret value, not the client secret ID, fo
r a secret added to app \"609b7870-f8ce-47cf-aaf7-1d6f87f2e481\"",
            "code": "400",
            "status": "400"
        }
    ]
}
```

Update an Existing App Volumes Manager and Azure Virtual Desktop Integration

Use this API to update an existing, specific integration between App Volumes Manager and Azure Virtual Desktop.

Note: All attributes except `organization` (subscription ID) and `group` can be edited. For more information about editing an integration, see the *What To Do Next* section in *Configure App Volumes Manager Integration with Azure Virtual Desktop* topic in the App Volumes Administration Guide.

Parameters

These are the mandatory parameters that must be used with this API.

- **`api_version`**

  This parameter indicates the version of the REST API used in App Volumes. The value of the parameter must be `4160`.

- **`id`**

  This parameter indicates a specific integration which you choose to update. To obtain the `id` of the integration, see List All App Volumes Manager and Azure Virtual Desktop Integrations.

Prerequisites

For information about using the API to configure an integration, see the *Configure App Volumes Manager Integration with Azure Virtual Desktop* topic in the App Volumes Administration Guide.

Procedure

1. Make a PUT request to update details of an existing integration by specifying the integration ID.

   ```
   PUT AV_Manager_URL/app_volumes/avd_integrations/{id}
   ```

   For example:

   ```
   {
       "data": {
           "type": "avd_integrations",
           "attributes": {
               "name": "Prod West US Integration",
               "organization": "f68c64fd-870c-4d07-a9f6-b6ff41b76c38",
               "group": "ResourceGroup",
               "tenant": "e59c64fd-870c-4d07-a9f6-b6ff41b76d11",
               "client_id": "d79c64fd-870c-4d07-a9f6-b6ff41b76d18",
               "client_secret": "updated_client_secret_value",
               "integration_tags": {
                   "tag_key": "updated_tag_value"
               }
           }
   ```

```
    }
}
```

2. Examine the response.

   - A successful response returns the details of the updated integration with an HTTP status `200`.

     For example:


```
{
    "data": {
        "id": "5F02DE3B-BAB1-4FDB-8065-ABB8711013EF",
        "type": "avd_integrations",
        "links": {
            "self": "http://localhost:3000/app_volumes/...?page%5Bnu
mber%5D=1&page%5Bsize%5D=3"
        },
        "attributes": {
            "name": "Prod West US Integration",
            "organization": "f68c64fd-870c-4d07-a9f6-b6ff41b76c38",
            "group": "ResourceGroup",
            "tenant": "e59c64fd-870c-4d07-a9f6-b6ff41b76d11",
            "client_id": "d79c64fd-870c-4d07-a9f6-b6ff41b76d18",
            "credential_id": "118",
            "guid": "1b38f44f-cb39-46fa-bcac-d72af508ddee",
            "configuration_id": null,
            "configuration_message": "Works great!",
            "configuration_status": "Active",
            "configuration_type": "AVD",
            "enabled": true,
            "quota_message": null,
            "quota_status": null,
            "integration_count": 0,
            "status": "active",
            "delete_status": "deleting",
            "synced_at": "2021-12-03 13:10:13 -0800",
            "synced_at_human": "Dec 03 2021",
            "deleted_at": "2021-12-04 13:10:13 -0800",
            "deleted_at_human": "Dec 04 2021",
            "created_at": "2021-12-03 13:10:13 -0800",
            "created_at_human": "Dec 03 2021",
            "updated_at": "2021-12-03 13:10:13 -0800",
            "updated_at_human": "Dec 03 2021",
            "sync_scheduled_at": "2021-12-03 13:10:13 -0800",
            "sync_scheduled_at_human": "Dec 03 2021"
        },
        "relationships": {
            "integration_tags": {
                "data": [
                    {
                        "type": "integration_tags",
                        "id": 20
                    }
                ]
            }
        }
    },
```

```
    "included": [
        {
            "id": 20,
            "type": "integration_tags",
            "attributes": {
                "key": "production",
                "value": "test",
                "parent_guid": "5F02DE3B-BAB1-4FDB-8065-ABB8711013E
F",
                "parent_id": 0,
                "created_at": "2021-12-03 13:10:13 -0800",
                "created_at_human": "Dec 03 2021",
                "updated_at": "2021-12-03 13:10:13 -0800",
                "updated_at_human": "Dec 03 2021"
            }
        }
    ]
}
```


- If you do not provide the required data in the request, an HTTP status code 400 is returned. For example, if you provide the tag name which is a reserved keyword, then the following response is fetched:


```
{
    "errors": [
        {
            "title": "Invalid Parameter",
            "detail": "The tag name 'azure' begins with a reserved k
eyword (e.g., 'microsoft', 'azure', or 'windows') and cannot be use
d",
            "code": "400",
            "source": {
                "parameter": "data.attributes.integration_tags"
            },
            "status": "400"
        }
    ]
}
```


Delete an App Volumes Manager and Azure Virtual Desktop Integration

Use this API to delete a specific integration between App Volumes Manager and Azure Virtual Desktop.

Note: When an integration is deleted, all the associated, synchronized App attach packages are deleted from Azure Virtual Desktop. In the App Volumes Manager admin UI, you can use the **DELETE** button to perform this action.

Parameters

`api_version` and `id` are mandatory parameters whereas `force` is an optional parameter.

- **`api_version`**

This parameter indicates the version of the REST API used in App Volumes. The value of the parameter must be `4150`.

- **`id`**

  This parameter indicates a specific integration which you choose to delete. To obtain the `id` of the integration, see List All App Volumes Manager and Azure Virtual Desktop Integrations.

- **`force`**

  This parameter deletes the integration record from the App Volumes database, but does not delete the App attach packages from Azure Virtual Desktop. After the record is deleted from the database, App attach packages must be manually deleted from Azure Virtual Desktop.

  If an integration fails to get deleted due to issues in the network or within Azure, then such an integration is in the *Delete Failed* status. In such a situation, using the `force` parameter deletes the integration record from App Volumes database.

  The values of this parameter can be `true` or `false`. By default, the value of this parameter is `false`.

Procedure

1. Send a DELETE request to delete a specific integration between App Volumes Manager and Azure Virtual Desktop.

   ```
   DELETE AV_Manager_URL/app_volumes/avd_integrations/{id}
   ```

2. Examine the response.

   - A successful request returns an HTTP status code 200 with detailed information about the deleted integration.


     ```
     {
         "data": {
             "id": "5F02DE3B-BAB1-4FDB-8065-ABB8711013EF",
             "type": "avd_integrations",
             "links": {
             "self": "http://localhost:3000/app_volumes/...?page%5Bnumbe
     r%5D=1&page%5Bsize%5D=3"
             },
             "attributes": {
             "name": "Prod West US Integration",
             "organization": "f68c64fd-870c-4d07-a9f6-b6ff41b76c38",
             "group": "ResourceGroup",
             "tenant": "e59c64fd-870c-4d07-a9f6-b6ff41b76d11",
             "client_id": "d79c64fd-870c-4d07-a9f6-b6ff41b76d18",
             "credential_id": "118",
             "guid": "1b38f44f-cb39-46fa-bcac-d72af508ddee",
             "configuration_id": null,
             "configuration_message": "Works great!",
             "configuration_status": "Active",
             "configuration_type": "AVD",
             "enabled": true,
             "quota_message": null,
             "quota_status": null,
             "integration_count": 0,
             "status": "active",
     ```


```
        "delete_status": "deleting",
        "synced_at": "2021-12-03 13:10:13 -0800",
        "synced_at_human": "Dec 03 2021",
        "deleted_at": "2021-12-04 13:10:13 -0800",
        "deleted_at_human": "Dec 04 2021",
        "created_at": "2021-12-03 13:10:13 -0800",
        "created_at_human": "Dec 03 2021",
        "updated_at": "2021-12-03 13:10:13 -0800",
        "updated_at_human": "Dec 03 2021",
        "sync_scheduled_at": "2021-12-03 13:10:13 -0800",
        "sync_scheduled_at_human": "Dec 03 2021"
        },
        "relationships": {
        "integration_tags": {
            "data": [
            {
                "type": "integration_tags",
                "id": 20
            }
            ]
        }
        }
    },
    "included": [
        {
        "id": 20,
        "type": "integration_tags",
        "attributes": {
            "key": "environment",
            "value": "test",
            "parent_guid": "5F02DE3B-BAB1-4FDB-8065-ABB8711013EF",
            "parent_id": 0,
            "created_at": "2021-12-03 13:10:13 -0800",
            "created_at_human": "Dec 03 2021",
            "updated_at": "2021-12-03 13:10:13 -0800",
            "updated_at_human": "Dec 03 2021"
        }
        }
    ]
}
```


- If an invalid parameter is provided in the API request, then HTTP status code 400 is returned and the integration is not deleted.
  For example, if `api_version` is provided incorrectly as `4240`, then the following response is fetched:


```
{
    "errors": "Invalid or unsupported API version requested: 4240"
}
```


List All App Volumes Manager and Azure Virtual Desktop Integrations

Use this API to list all configured integrations and the information associated with each integration such as name,

organization (subscription ID), resource group, tenant ID, client ID, tag value and key, and so on.

Parameters

All parameters except `api_version` are optional. The `api_version` parameter is mandatory. You can choose to list the integrations on a specific page using the pagination parameters such as `page[number]` and `page[size]`.

- **`page[number]`**

  This parameter indicates the page number for which the integration records must be listed. The default value is `1`.

  Note: At a time, only one page number can be provided as the parameter value.

- **`page[size]`**

  This parameter indicates the number of integration records that must be listed on a page. The default value is `10`.

- **`api_version`**

  This parameter indicates the version of the REST API used in App Volumes. The value of the parameter must be `4160`.

- **`include`**

  This parameter fetches relationships of each integraton tag (`integration_tags`) added to the integration.

- **`filter[name]`, `filter[organization]`, `filter[type]`, `filter[status]` `filter[integration_count]`**

  The `filter` parameters allow retrieving integration records based on the name of the integration, organization (subscription ID), type of integration (AVD), status of the integration, and integration count (number of application packages that are available for synchronization in each integration).

- **`links`**

  Displays the links to the first, next, and last pages of the integration records.

  - `first` - displays the link to the first page of the integration records
  - `next` - displays the link to the page next to the page number specified in the GET request (`page[num]`)
  - `last` - displays the link to the last page of the integration records.

Procedure

1. Make a GET request to obtain all the integration records that exist between App Volumes Manager and Azure Virtual Desktop.

   ```
   GET AV_Manager_URL/app_volumes/integrations
   ```

   Note: If no pagination parameters are specified in the GET request, the first ten integration records are fetched in the response.

2. Examine the response.

    - A successful response returns available data for each integration record with HTTP status 200. If the `include` parameter is provided in the GET request, the response displays the `included` object.


```
{
    "data": [
        {
        "id": "2d3c3de2-dc8e-4cf9-8e26-ec36cdc79b26",
        "type": "avd_integrations",
        "links": {
            "self": "http://localhost:3000/app_volumes/avd_integrations/2d
3c3de2-dc8e-4cf9-8e26-ec36cdc79b26"
        },
        "attributes": {
            "name": "ANUJ_TEST_001",
            "guid": "2d3c3de2-dc8e-4cf9-8e26-ec36cdc79b26",
            "tenant": "43fd6a96-86b1-4515-becc-a03541265cea",
            "organization": "d6e21103-6f9f-4ef7-9e0b-a94bfb946fc4",
            "group": "ANUJ_TEST_001",
            "configuration_id": null,
            "configuration_type": null,
            "configuration_status": "Active",
            "configuration_message": "",
            "quota_status": null,
            "quota_message": null,
            "integration_count": 6,
            "enabled": true,
            "status": "Active",
            "credential_id": 7,
            "synced_at": "2024-11-26 09:02:08 -0800",
            "synced_at_human": "Nov 26 2024",
            "sync_scheduled_at": null,
            "sync_scheduled_at_human": "",
            "created_at": "2024-11-26 09:01:29 -0800",
            "created_at_human": "Nov 26 2024",
            "updated_at": "2024-11-26 09:02:08 -0800",
            "updated_at_human": "Nov 26 2024",
            "delete_status": null,
            "deleted_at": null,
            "deleted_at_human": "",
            "client_id": "609b7870-f8ce-47cf-aaf7-1d6f87f2e481"
        },
        "relationships": {
            "integration_tags": {
            "data": [
                {
                "type": "integration_tags",
                "id": "280"
                }
            ]
            }
        }
        },
        {
        "id": "f5057946-aee7-44d7-8fc2-0d5c7a542c09",
        "type": "avd_integrations",
        "links": {
```

```
            "self": "http://localhost:3000/app_volumes/avd_integrations/f5
057946-aee7-44d7-8fc2-0d5c7a542c09"
        },
        "attributes": {
            "name": "DIGEUCVMW.onmicrosoft.com",
            "guid": "f5057946-aee7-44d7-8fc2-0d5c7a542c09",
            "tenant": "f0d0eb57-325a-4a72-8340-75708d7b69a9",
            "organization": "f5760cc9-f9b9-4429-afa6-83c6adae7904",
            "group": "App_Volumes_RG",
            "configuration_id": null,
            "configuration_type": null,
            "configuration_status": "Active",
            "configuration_message": "",
            "quota_status": null,
            "quota_message": null,
            "integration_count": 6,
            "enabled": true,
            "status": "Active",
            "credential_id": 8,
            "synced_at": "2024-11-26 09:02:49 -0800",
            "synced_at_human": "Nov 26 2024",
            "sync_scheduled_at": null,
            "sync_scheduled_at_human": "",
            "created_at": "2024-11-26 09:02:22 -0800",
            "created_at_human": "Nov 26 2024",
            "updated_at": "2024-11-26 09:02:49 -0800",
            "updated_at_human": "Nov 26 2024",
            "delete_status": null,
            "deleted_at": null,
            "deleted_at_human": "",
            "client_id": "2159cf93-42f5-4161-95fc-a67184e9320b"
        },
        "relationships": {
            "integration_tags": {
            "data": [
                {
                "type": "integration_tags",
                "id": "281"
                },
                {
                "type": "integration_tags",
                "id": "282"
                }
            ]
            }
        }
        }
    ],
    "included": [
        {
        "id": "280",
        "type": "integration_tags",
        "attributes": {
            "key": "owner",
            "value": "anujp",
            "parent_guid": "2d3c3de2-dc8e-4cf9-8e26-ec36cdc79b26",
            "parent_id": null,
            "created_at": "2024-11-26 09:01:29 -0800",
```

```
            "created_at_human": "Nov 26 2024",
            "updated_at": "2024-11-26 09:01:29 -0800",
            "updated_at_human": "Nov 26 2024"
        }
        },
        {
        "id": "281",
        "type": "integration_tags",
        "attributes": {
            "key": "owner",
            "value": "anujp",
            "parent_guid": "f5057946-aee7-44d7-8fc2-0d5c7a542c09",
            "parent_id": null,
            "created_at": "2024-11-26 09:02:22 -0800",
            "created_at_human": "Nov 26 2024",
            "updated_at": "2024-11-26 09:02:22 -0800",
            "updated_at_human": "Nov 26 2024"
        }
        },
        {
        "id": "282",
        "type": "integration_tags",
        "attributes": {
            "key": "testing",
            "value": "123",
            "parent_guid": "f5057946-aee7-44d7-8fc2-0d5c7a542c09",
            "parent_id": null,
            "created_at": "2024-11-26 09:02:22 -0800",
            "created_at_human": "Nov 26 2024",
            "updated_at": "2024-11-26 09:02:22 -0800",
            "updated_at_human": "Nov 26 2024"
        }
        }
    ],
    "meta": {
        "request_id": 1,
        "total": 2,
        "filtered": 2,
        "page_count": 1
    },
    "links": {
        "first": "http://localhost:3000/app_volumes/integrations?include=i
ntegration_tags&page%5Bnumber%5D=1&page%5Bsize%5D=10",
        "last": "http://localhost:3000/app_volumes/integrations?include=in
tegration_tags&page%5Bnumber%5D=1&page%5Bsize%5D=10"
    }
    }
```


- If you do not provide the required data in the request, an HTTP status code 400 is returned and the integration details are not fetched. For example, if `api_version` is incorrectly provided as 4240 in the request, then the following response is fetched:


```
{
    "errors": "Invalid or unsupported API version requested: 4240"
}
```

List Details of a Specific Integration

Use this API to list a specific integration and the information associated with that integration.

Parameters

`api_version` and `id` are mandatory parameters whereas `include` is an optional parameter.

- **`api_version`**

  This parameter indicates the version of the REST API used in App Volumes. The value of the parameter must be `4160`.

- **`id`**

  This parameter indicates a specific integration which you choose to list. To obtain the `id` of the integration, see List All App Volumes Manager and Azure Virtual Desktop Integrations.

- **`include`**

  This parameter fetches relationships of each integraton tag (`integration_tags`) added to this integration.

Procedure

1. Make a GET request to obtain the details of a specific integration.

   ```
   GET AV_Manager_URL/app_volumes/integrations/{id}
   ```

2. Examine the response.

   - A successful response returns details of that specific integration with HTTP status 200.

     In this example, `"id": "5F02DE3B-BAB1-4FDB-8065-ABB8711013EF"` is provided in the request and the following details are fetched for the integration, `Prod West US Integration`.

     ```
     {
         "data": {
             "id": "5F02DE3B-BAB1-4FDB-8065-ABB8711013EF",
             "type": "avd_integrations",
             "links": {
             "self": "http://localhost:3000/app_volumes/...?page%5Bnumbe
     r%5D=1&page%5Bsize%5D=3"
             },
             "attributes": {
                 "name": "Prod West US Integration",
                 "organization": "f68c64fd-870c-4d07-a9f6-b6ff41b76c38",
                 "group": "ResourceGroup",
                 "tenant": "e59c64fd-870c-4d07-a9f6-b6ff41b76d11",
                 "client_id": "d79c64fd-870c-4d07-a9f6-b6ff41b76d18",
                 "credential_id": "118",
                 "guid": "1b38f44f-cb39-46fa-bcac-d72af508ddee",
     ```

```
            "configuration_id": null,
            "configuration_message": "Works great!",
            "configuration_status": "Active",
            "configuration_type": "AVD",
            "enabled": true,
            "quota_message": null,
            "quota_status": null,
            "integration_count": 0,
            "status": "active",
            "delete_status": "deleting",
            "synced_at": "2021-12-03 13:10:13 -0800",
            "synced_at_human": "Dec 03 2021",
            "deleted_at": "2021-12-04 13:10:13 -0800",
            "deleted_at_human": "Dec 04 2021",
            "created_at": "2021-12-03 13:10:13 -0800",
            "created_at_human": "Dec 03 2021",
            "updated_at": "2021-12-03 13:10:13 -0800",
            "updated_at_human": "Dec 03 2021",
            "sync_scheduled_at": "2021-12-03 13:10:13 -0800",
            "sync_scheduled_at_human": "Dec 03 2021"
        },
        "relationships": {
            "integration_tags": {
                "data": [
                    {
                        "type": "integration_tags",
                        "id": 20
                    }
                ]
            }
        }
    },
    "included": [
        {
        "id": 20,
        "type": "integration_tags",
        "attributes": {
                "key": "environment",
                "value": "test",
                "parent_guid": "5F02DE3B-BAB1-4FDB-8065-ABB8711013E
F",
                "parent_id": 0,
                "created_at": "2021-12-03 13:10:13 -0800",
                "created_at_human": "Dec 03 2021",
                "updated_at": "2021-12-03 13:10:13 -0800",
                "updated_at_human": "Dec 03 2021"
            }
        }
    ]
}
```


- If you do not provide the required data in the request, an HTTP status code 400 is returned and the integration details are not fetched. For example, if `api_version` is incorrectly provided as 4240 in the request, then the following response is fetched:


```
{
    "errors": "Invalid or unsupported API version requested: 4240"
```

```
}
```

Synchronize All App Volumes Manager and Azure Virtual Desktop Integrations

Use this API to initiate synchronization for all configured integrations between App Volumes Manager and Azure Virtual Desktop. This synchronization process ensures that all `CURRENT` packages across all integrations are synchronized with Azure Virtual Desktop.

Parameters

- **`api_version`**

  This parameter indicates the version of the REST API used in App Volumes. The value of the parameter must be `4150`.

- **`action`**

  This parameter can be used to trigger synchronization for either a specific integration or for all configured integrations.

  The values of this parameter are as follows:

  - `sync_all` - triggers synchronization for all configured integrations
  - `sync` - triggers synchronization for a specific integration

Prerequisites

For information about synchronization, see *Synchronization Triggers between App Volumes Manager and Azure Virtual Desktop Integration* and *Manually Synchronize App Volumes Manager and Azure Virtual Desktop Integrations* topics in the App Volumes Administration Guide.

Procedure

1. Make a POST request to synchronize `CURRENT` application packages of all integrations with Azure Virtual Desktop.

   ```
   POST AV_Manager_URL/app_volumes/integrations
   ```

2. Examine the response.

   - A successful request returns an HTTP status code 200 and `scheduled` as true in the response. In the following example, the `scheduled` parameter indicates that synchronization is successfully scheduled for all enabled (active) integrations. As a result, all `CURRENT` packages across all integrations are synchronized with Azure Virtual Desktop.

     ```
     {
         "meta": {
             "message": "Scheduled synchronization for all enabled integr
     ```

```
ations",
        "scheduled": true
    }
}
```

When synchronization is in progress and if you repeatedly try to synchronize, the following response is fetched:

```
{
    "meta": {
        "message": "Synchronization for all enabled integrations is
already scheduled",
        "scheduled": false
    }
}
```

- If an invalid parameter is provided in the API request, then HTTP status code 400 is returned and synchronization is not performed.
  For example, if `api_version` is provided incorrectly as `4240`, then the following response is fetched:

```
{
    "errors": "Invalid or unsupported API version requested: 4240"
}
```

Synchronize a Specific App Volumes Manager and Azure Virtual Desktop Integration

Use this API to initiate synchronization for a specific configured integration by providing an integration ID. This synchronization process ensures that all the `CURRENT` packages for that integration is synchronized with Azure Virtual Desktop.

Parameters

All parameters are mandatory when using this API.

- **`api_version`**

  This parameter indicates the version of the REST API used in App Volumes. The value of the parameter must be `4150`.

- **`action`**

  This parameter can be used to trigger synchronization for either a specific integration or for all configured integrations.

  The values of this parameter are as follows:

  - `sync_all` - triggers synchronization for all configured integrations
  - `sync` - triggers synchronization for a specific integration

- **id**

  This parameter indicates the specific integration which needs to be synchronized.

Prerequisites

For information about synchronization, see *Synchronization Triggers between App Volumes Manager and Azure Virtual Desktop Integration* and *Manually Synchronize App Volumes Manager and Azure Virtual Desktop Integrations* topics in the App Volumes Administration Guide.

Procedure

1. Make a POST request to synchronize `CURRENT` packages of this specific integration with Azure Virtual Desktop.

   ```
   POST AV_Manager_URL/app_volumes/integrations/{id}
   ```

2. Examine the response.

   - A successful request returns an HTTP status code 200 and scheduled as true. In the following example, synchronization is successfully done for TestAvdIntegration. As a result, all `CURRENT` packages of this integration are synchronized with Azure Virtual Desktop.

     ```
     {
         "meta": {
             "message": "Scheduled synchronization for integration TestAv
     dIntegration",
             "scheduled": true
         }
     }
     ```

   - If an invalid parameter is provided in the API request, then HTTP status code 400 is returned and the synchronization is not performed.
     For example, if `api_version` is provided incorrectly as `4240`, then the following response is fetched:

     ```
     {
         "errors": "Invalid or unsupported API version requested: 4240"
     }
     ```

Working with Writable Volumes APIs

You can perform various operations on the Writable Volumes using the APIs.

You can get detailed information about the Writable Volume, and expand, update, or delete a volume.

List Writable Volumes and Datastore Information

Retrieve a list of Writable Volumes and the associated datastore information such the volume GUID, the date and time it was created, whether the volume is attached, and so on.

Few attributes in this API are used for indicating the space on Writable Volumes. The unit of measurement for these attributes is in MB. They are as follows:

- **total_mb**

  Current total space present on the Writable Volume

  This value can differ from `size_mb` and `requested_mb` due to datastore file formatting and expansion operations.

- **free_mb**

  Currently available free space on the Writable Volume

- **size_mb**

  Currently used space on the Writable Volume

- **requested_mb**

  Requested size of the Writable Volume as mentioned in the API used for expanding the Writable Volume.

  The value of this attribute is `null` after the volume is expanded or if no request for expansion has been made. This value can differ from `total_mb` due to datastore file formatting and expansion operations.

Procedure

1. Send the following GET request to retrieve the list of Writable Volumes.

   ```
   GET AV_Manager_URL/app_volumes/writables
   ```

2. Examine the response.

   A list of volumes and associated information is returned. For example:

   ```
   {
     "data": [
       {
         "id": 0,
         "name": "SNAPVOLUMES\test_user",
         "title": "SNAPVOLUMES\test_user",
         "owner_id": 2,
   ```

```
      "owner_name": "TestUser",
      "owner_type": "User",
      "owner_upn": "SNAPVOLUMES\test_user",
      "owner_object_guid": "8da04bfa-bbbe-45c3-9cbf-2d3d03897b84",
      "created_at": "2021-12-03 13:10:13 -0800",
      "created_at_human": "Dec 03 2021",
      "updated_at": "2021-12-03 13:10:13 -0800",
      "updated_at_human": "Dec 03 2021",
      "mounted_at": "2021-12-03 13:10:13 -0800",
      "mounted_at_human": "Dec 03 2021",
      "attached": "Attached",
      "status": "missing",
      "mount_count": 0,
      "total_mb": 10237,
      "free_mb": 5237,
      "size_mb": 5000,
      "requested_mb": 20480,
      "percent_available": "Unknown",
      "template_version": "4.0.0",
      "version_count": 0,
      "type": "DataDisk",
      "display_type": "Writable Volume",
      "error_action": "string",
      "busy": false,
      "state": "Moving",
      "filename": "snapvolumes_test_user.vmdk",
      "path": "appvolumes/writable",
      "datastore_name": "SSD Disk Array RAID8",
      "volume_guid": "volumeguid-8-volumeguid",
      "datastore_host": "null",
      "can_expand": true
    }
  ],
  "counts": {
    "total": 1,
    "warning": 0,
    "critical": 0
  }
}
```


Retrieve Writable Volume Information

Retrieve information about a Writable Volume such as the name, owner, size, path to the Writable Volume, and so on.

Few attributes in this API are used for indicating the space on Writable Volumes. The unit of measurement for these attributes is in MB. For information about these attributes, see List Writable Volumes and Datastore Information.

Prerequisites

You must know the ID of the Writable Volume for which you want to retrieve the information.

Procedure

1. Send the following GET request to retrieve the list of volumes.

   ```
   GET AV_Manager_URL/app_volumes/writables/{id}
   ```

   If the request is completed successfully, an HTTP status code of 200 is returned with information about the volume. For example:


```
{
  "writable": {
    "id": 0,
    "name": "SNAPVOLUMES\test_user",
    "name_html": "SNAPVOLUMES\test_user",
    "title": "SNAPVOLUMES\test_user",
    "title_html": "SNAPVOLUMES\test_user",
    "owner": "<a href=\\\"/directory#/Users/2\\\" title=\\\"test_use
r\\\">TESTDOMAIN\\\\test_user</a>",
    "link": "/directory#/Users/2",
    "owner_id": 3,
    "owner_name": "TestUser",
    "owner_type": "User",
    "owner_upn": "SNAPVOLUMES\test_user",
    "owner_upn_html": "SNAPVOLUMES\test_user",
    "owner_object_guid": "8da04bfa-bbbe-45c3-9cbf-2d3d03897b84",
    "description": "My writable volume",
    "created_at": "2021-12-03 13:10:13 -0800",
    "created_at_human": "Dec 03 2021",
    "updated_at": "2021-12-03 13:10:13 -0800",
    "updated_at_human": "Dec 03 2021",
    "mounted_at": "2021-12-03 13:10:13 -0800",
    "mounted_at_human": "Dec 03 2021",
    "mounted_on": "Machine <DESKTOP-VM123> (#{1234567812345678})",
    "mount_count": 0,
    "attached": "Attached",
    "status": "enabled",
    "version_count": 0,
    "version_tag": "0",
    "error_action": "",
    "block_login": true,
    "enable_version": false,
    "mount_prefix": "DESKTOP",
    "defer_create": true,
    "total_mb": 10237,
    "free_mb": 5237,
    "size_mb": 5000,
    "requested_mb": 20480,
    "template_version": "4.0.0",
    "busy": false,
    "state": "Moving",
    "volume_guid": "volumeguid-8-volumeguid",
    "datastore_name": "SSD Disk Array RAID8",
    "machine_manager_host": "0.0.0.0",
    "machine_manager_type": "Machine Manager",
    "path": "appvolumes/writable",
    "filename": "snapvolumes_test_user_15.vmdk",
    "file_location": "[SSD Disk Array RAID8] appvolumes/writable/snapvolum
es_test_user_15.vmdk",
```

```
    "type": "DataDisk",
    "display_type": "Writable Volume",
    "files_count": 1,
    "template_file_name": "[SSD Disk Array RAID8] appvolumes/writable_temp
lates/template_uia_only_persistent.vmdk",
    "protected": true,
    "percent_available": "Unknown",
    "can_expand": true,
    "datastore_host": "null",
    "primordial_os_id": 15,
    "primordial_os_name": "Windows 8.1 (x64)",
    "oses": [
      {
        "id": 0,
        "name": "Windows 8.1 (x64)"
      }
    ]
  }
}
```

If Omnissa App Volumes is unable to locate a Writable Volume, an HTTP error code 404 is returned with the following error message: `Writable Volume was not found`.

Expand a Writable Volume

Increase the size of a Writable Volume.

When expanding a Writable Volume, `size_mb` attribute is used to provide the requested size for the Writable Volume. When using the APIs for listing multiple Writable Volumes or a specific Writable Volume, the value of the `requested_mb` attribute is the same as the `size_mb` attribute used in this API. All units of measurement are in MB. For more information about `requested_mb`, see List Writable Volumes and Datastore Information.

Prerequisites

Before you expand the Writable Volume, ensure the following:

- The Writable Volume is not attached.
- You have logged out from the Writable Volume.
- The App Volumes Manager is licensed and configured correctly.

Procedure

1. Make a POST request to expand the Writable Volume by including the volume ID and the size you want to expand the volume to in the POST request body.

   ```
   POST AV_Manager_URL/app_volumes/writables/grow
   ```

   The following example shows a request where a Writable Volume id `0` is expanded to `20480 MB`:

   ```
   {
     "size_mb": 20480,
   ```

```
  "volumes": [
    0
  ]
}
```

2. Examine the response.

   A successful request returns 200 with a message.

   For example:

```
{
    "success": [
        "Successfully expanded the Writable VOlume to 20480 MB"
    ]
}
```

   You might also get an HTTP response of 200 when there are warnings or errors.

   For example: The following warning is displayed if you have not detached the volume or logged out from the Writable Volume

```
 "warnings": [ "Writable Volume XYZ is attached. Make sure you shut down/l
ogoff XYZ." ]  }
```

   For example: The following warning is displayed when there is a mismatch between the provided size of the Writable Volume and the current size.

```
"errors": [ "Error expanding Writable Volume XYZ" ]
```

   If the request is missing parameters or if App Volumes Manager is not configured correctly, an HTTP response of 400 is returned.

   If App Volumes is unable to locate a Writable Volume, an HTTP error code 404 is returned with the following error message: `Writable Volume was not found`.

Update a Writable Volume

You can update Writable Volume settings such as description, whether users can log in if there is a failure, prefix value of the computer the volume must be mounted to, and the operating system.

Prerequisites

Ensure that the Writable Volume you want to modify is not attached and that you have logged off from the given volume.

Procedure

1. Make a PUT request to update the volume. Include the properties of the volume that you want to update in the request body.

```
PUT AV_Manager_URL/app_volumes/writables/{id}
```

You can update the following properties of the Writable Volume:

Option	Description
Description (string)	Description of the Writable Volume.
error-action (string)	◦ continue_silently - skip attachments and do not alert the user. ◦ **continue_alert** - alert the end user that the writable was not attached. ◦ **disable_and_alert** - disable all virtualization and display an alert. ◦ **disable_and_alert_on_error** - disable all virtualization and display an alert only for error cases (conflicts due to multiple logins are not considered an error). Must set as “” if a) owner_type is “Computer” or b) block_login parameter is set to “1”.
block_login (integer)	◦ Set to ‘0’ to prevent users from logging in if there are failures related to this Writable Volume. ◦ Set to ‘1’ to allow users to log in even if there are failures related to this Writable Volume.
mount_prefix	Prefix used to filter which computer the Writable Volume can be mounted to.
OS	The IDs of OSes the Writable Volume can be mounted to.

For example:

```
{
  "description": "Writable Volume for Administrator",
  "error_action": "",
  "block_login": 0,
  "mount_prefix": "DESKTOP",
  "oses": [
    0
  ]
}
```

2. Examine the response.

   A successful request returns HTTP response 200 with the following message:

   ```
   Saved Writable changes.
   ```

   If Omnissa App Volumes is unable to locate a Writable Volume, an HTTP error code 404 is returned with the following error message: `Writable Volume was not found`.

Delete a Writable Volume

When a Writable Volume is deleted, the volume is immediately detached from all computers. All associated data and settings are also permanently deleted.

Prerequisites

- Ensure that the Writable Volume you want to delete is not in use by any user or computer.
- You must know the ID of the Writable Volume that you want to delete.

Procedure

1. Send the following DELETE request to delete a volume. You can also set the volume to be deleted in the background.

   ```
   DELETE /app_volumes/writables/{id}
   ```

2. Examine the response. If the volume is deleted successfully, you will receive an HTTP status code 200 with detailed information about the deleted volume.

   For example:

   ```
   {
     "success": "Deleted 1 volume",
     "error": "Unable to delete 1 volume",
     "snapvols": {
       "not_found": [
         {
           "id": 0,
           "name": "SNAPVOLUMES\test_user",
   ```

```
        "name_html": "SNAPVOLUMES\test_user",
        "title": "SNAPVOLUMES\test_user",
        "title_html": "SNAPVOLUMES\test_user",
        "owner": "<a href=\\\"/directory#/Users/2\\\" title=\\\"test_use
r\\\">TESTDOMAIN\\\\test_user</a>",
        "link": "/directory#/Users/2",
        "owner_id": 3,
        "owner_name": "TestUser",
        "owner_type": "User",
        "owner_upn": "SNAPVOLUMES\test_user",
        "owner_upn_html": "SNAPVOLUMES\test_user",
        "owner_object_guid": "8da04bfa-bbbe-45c3-9cbf-2d3d03897b84",
        "description": "My writable volume",
        "created_at": "2021-12-03 13:10:13 -0800",
        "created_at_human": "Dec 03 2021",
        "updated_at": "2021-12-03 13:10:13 -0800",
        "updated_at_human": "Dec 03 2021",
        "mounted_at": "2021-12-03 13:10:13 -0800",
        "mounted_at_human": "Dec 03 2021",
        "mounted_on": "Machine <DESKTOP-VM123> (#{1234567812345678})",
        "mount_count": 0,
        "attached": "Attached",
        "status": "enabled",
        "version_count": 0,
        "version_tag": "0",
        "error_action": "",
        "block_login": true,
        "enable_version": false,
        "mount_prefix": "DESKTOP",
        "defer_create": true,
        "size_mb": 0,
        "template_version": "4.0.0",
        "busy": false,
        "state": "Moving",
        "volume_guid": "volumeguid-8-volumeguid",
        "datastore_name": "SSD Disk Array RAID8",
        "machine_manager_host": "0.0.0.0",
        "machine_manager_type": "Machine Manager",
        "path": "appvolumes/writable",
        "filename": "snapvolumes_test_user_15.vmdk",
        "file_location": "[SSD Disk Array RAID8] appvolumes/writable/snapv
olumes_test_user_15.vmdk",
        "type": "DataDisk",
        "display_type": "Writable Volume",
        "files_count": 1,
        "template_file_name": "[SSD Disk Array RAID8] appvolumes/writabl
e_templates/template_uia_only_persistent.vmdk",
        "protected": true,
        "free_mb": 0,
        "total_mb": 0,
        "percent_available": "Unknown",
        "can_expand": true,
        "datastore_host": "null",
        "primordial_os_id": 15,
        "primordial_os_name": "Windows 8.1 (x64)",
        "oses": [
          {
            "id": 0,
```

```
            "name": "Windows 8.1 (x64)"
          }
        ]
      }
    ],
    "success": [
      {
        "id": 0,
        "name": "SNAPVOLUMES\test_user",
        "name_html": "SNAPVOLUMES\test_user",
        "title": "SNAPVOLUMES\test_user",
        "title_html": "SNAPVOLUMES\test_user",
        "owner": "<a href=\\\"/directory#/Users/2\\\" title=\\\"test_use
r\\\">TESTDOMAIN\\\\test_user</a>",
        "link": "/directory#/Users/2",
        "owner_id": 3,
        "owner_name": "TestUser",
        "owner_type": "User",
        "owner_upn": "SNAPVOLUMES\test_user",
        "owner_upn_html": "SNAPVOLUMES\test_user",
        "owner_object_guid": "8da04bfa-bbbe-45c3-9cbf-2d3d03897b84",
        "description": "My writable volume",
        "created_at": "2021-12-03 13:10:13 -0800",
        "created_at_human": "Dec 03 2021",
        "updated_at": "2021-12-03 13:10:13 -0800",
        "updated_at_human": "Dec 03 2021",
        "mounted_at": "2021-12-03 13:10:13 -0800",
        "mounted_at_human": "Dec 03 2021",
        "mounted_on": "Machine <DESKTOP-VM123> (#{1234567812345678})",
        "mount_count": 0,
        "attached": "Attached",
        "status": "enabled",
        "version_count": 0,
        "version_tag": "0",
        "error_action": "",
        "block_login": true,
        "enable_version": false,
        "mount_prefix": "DESKTOP",
        "defer_create": true,
        "size_mb": 0,
        "template_version": "4.0.0",
        "busy": false,
        "state": "Moving",
        "volume_guid": "volumeguid-8-volumeguid",
        "datastore_name": "SSD Disk Array RAID8",
        "machine_manager_host": "0.0.0.0",
        "machine_manager_type": "Machine Manager",
        "path": "appvolumes/writable",
        "filename": "snapvolumes_test_user_15.vmdk",
        "file_location": "[SSD Disk Array RAID8] appvolumes/writable/snapv
olumes_test_user_15.vmdk",
        "type": "DataDisk",
        "display_type": "Writable Volume",
        "files_count": 1,
        "template_file_name": "[SSD Disk Array RAID8] appvolumes/writabl
e_templates/template_uia_only_persistent.vmdk",
        "protected": true,
        "free_mb": 0,
```

```
        "total_mb": 0,
        "percent_available": "Unknown",
        "can_expand": true,
        "datastore_host": "null",
        "primordial_os_id": 15,
        "primordial_os_name": "Windows 8.1 (x64)",
        "oses": [
          {
            "id": 0,
            "name": "Windows 8.1 (x64)"
          }
        ]
      }
    ],
    "error": [
      {
        "id": 0,
        "name": "SNAPVOLUMES\test_user",
        "name_html": "SNAPVOLUMES\test_user",
        "title": "SNAPVOLUMES\test_user",
        "title_html": "SNAPVOLUMES\test_user",
        "owner": "<a href=\\\"/directory#/Users/2\\\" title=\\\"test_use
r\\\">TESTDOMAIN\\\\test_user</a>",
        "link": "/directory#/Users/2",
        "owner_id": 3,
        "owner_name": "TestUser",
        "owner_type": "User",
        "owner_upn": "SNAPVOLUMES\test_user",
        "owner_upn_html": "SNAPVOLUMES\test_user",
        "owner_object_guid": "8da04bfa-bbbe-45c3-9cbf-2d3d03897b84",
        "description": "My writable volume",
        "created_at": "2021-12-03 13:10:13 -0800",
        "created_at_human": "Dec 03 2021",
        "updated_at": "2021-12-03 13:10:13 -0800",
        "updated_at_human": "Dec 03 2021",
        "mounted_at": "2021-12-03 13:10:13 -0800",
        "mounted_at_human": "Dec 03 2021",
        "mounted_on": "Machine <DESKTOP-VM123> (#{1234567812345678})",
        "mount_count": 0,
        "attached": "Attached",
        "status": "enabled",
        "version_count": 0,
        "version_tag": "0",
        "error_action": "",
        "block_login": true,
        "enable_version": false,
        "mount_prefix": "DESKTOP",
        "defer_create": true,
        "size_mb": 0,
        "template_version": "4.0.0",
        "busy": false,
        "state": "Moving",
        "volume_guid": "volumeguid-8-volumeguid",
        "datastore_name": "SSD Disk Array RAID8",
        "machine_manager_host": "0.0.0.0",
        "machine_manager_type": "Machine Manager",
        "path": "appvolumes/writable",
        "filename": "snapvolumes_test_user_15.vmdk",
```

```
        "file_location": "[SSD Disk Array RAID8] appvolumes/writable/snapv
olumes_test_user_15.vmdk",
        "type": "DataDisk",
        "display_type": "Writable Volume",
        "files_count": 1,
        "template_file_name": "[SSD Disk Array RAID8] appvolumes/writabl
e_templates/template_uia_only_persistent.vmdk",
        "protected": true,
        "free_mb": 0,
        "total_mb": 0,
        "percent_available": "Unknown",
        "can_expand": true,
        "datastore_host": "null",
        "primordial_os_id": 15,
        "primordial_os_name": "Windows 8.1 (x64)",
        "oses": [
          {
            "id": 0,
            "name": "Windows 8.1 (x64)"
          }
        ]
      }
    ],
    "scheduled": [
      {
        "id": 0,
        "name": "SNAPVOLUMES\test_user",
        "name_html": "SNAPVOLUMES\test_user",
        "title": "SNAPVOLUMES\test_user",
        "title_html": "SNAPVOLUMES\test_user",
        "owner": "<a href=\\\"/directory#/Users/2\\\" title=\\\"test_use
r\\\">TESTDOMAIN\\\\test_user</a>",
        "link": "/directory#/Users/2",
        "owner_id": 3,
        "owner_name": "TestUser",
        "owner_type": "User",
        "owner_upn": "SNAPVOLUMES\test_user",
        "owner_upn_html": "SNAPVOLUMES\test_user",
        "owner_object_guid": "8da04bfa-bbbe-45c3-9cbf-2d3d03897b84",
        "description": "My writable volume",
        "created_at": "2021-12-03 13:10:13 -0800",
        "created_at_human": "Dec 03 2021",
        "updated_at": "2021-12-03 13:10:13 -0800",
        "updated_at_human": "Dec 03 2021",
        "mounted_at": "2021-12-03 13:10:13 -0800",
        "mounted_at_human": "Dec 03 2021",
        "mounted_on": "Machine <DESKTOP-VM123> (#{1234567812345678})",
        "mount_count": 0,
        "attached": "Attached",
        "status": "enabled",
        "version_count": 0,
        "version_tag": "0",
        "error_action": "",
        "block_login": true,
        "enable_version": false,
        "mount_prefix": "DESKTOP",
        "defer_create": true,
        "size_mb": 0,
```

```
        "template_version": "4.0.0",
        "busy": false,
        "state": "Moving",
        "volume_guid": "volumeguid-8-volumeguid",
        "datastore_name": "SSD Disk Array RAID8",
        "machine_manager_host": "0.0.0.0",
        "machine_manager_type": "Machine Manager",
        "path": "appvolumes/writable",
        "filename": "snapvolumes_test_user_15.vmdk",
        "file_location": "[SSD Disk Array RAID8] appvolumes/writable/snapv
olumes_test_user_15.vmdk",
        "type": "DataDisk",
        "display_type": "Writable Volume",
        "files_count": 1,
        "template_file_name": "[SSD Disk Array RAID8] appvolumes/writabl
e_templates/template_uia_only_persistent.vmdk",
        "protected": true,
        "free_mb": 0,
        "total_mb": 0,
        "percent_available": "Unknown",
        "can_expand": true,
        "datastore_host": "null",
        "primordial_os_id": 15,
        "primordial_os_name": "Windows 8.1 (x64)",
        "oses": [
          {
            "id": 0,
            "name": "Windows 8.1 (x64)"
          }
        ]
      }
    ]
  }
}
```


For example: If the operation has errors or the volume is missing, you receive a status 200.


```
{
  "error": "Unable to delete 1 volume",
   {
        "id": 0,
        "name": "SNAPVOLUMES\test_user",
        "name_html": "string",
        "title": "SNAPVOLUMES\test_user",
        "title_html": "string",
        "owner": "string",
        "link": "/directory#/Users/2",
        "owner_name": "TestUser",
        "owner_type": "User",
        "owner_upn": "SNAPVOLUMES\test_user",
        "owner_upn_html": "SNAPVOLUMES\test_user",
        "description": "string",
        "created_at": "string",
        "created_at_human": "string",
        "updated_at": "string",
        "updated_at_human": "string",
        "mounted_at": "string",
```

```
        "mounted_at_human": "string",
        "mounted_on": "Machine <DESKTOP-VM123> (#{1234567812345678})",
        "attached": "Attached",
        "status": "missing",
        "version_count": 0,
        "version_tag": 0,
        "block_login": true,
        "enable_version": true,
        "mount_prefix": "string",
        "defer_create": true,
        "size_mb": 0,
        "template_version": "string",
        "datastore_name": "SSD Disk Array RAID8",
        "machine_manager_host": "string",
        "machine_manager_type": "string",
        "path": "cloudvolumes/writable",
        "filename": "snapvolumes_test_user_15.vmdk",
        "file_location": "string",
        "mount_count": 0,
        "type": "string",
        "display_type": "string",
        "template_file_name": "string",
        "protected": true,
        "free_mb": 0,
        "total_mb": 0,
        "percent_available": "string",
        "can_expand": true,
        "storage_group": "string",
        "storage_group_members": "string",
        "primordial_os_id": 0,
        "primordial_os_name": "string",
        "oses": [
          {}
        ]
      }
    ],
     }
```


For example: If the operation has errors or the volume is missing, you receive a status 200.


```
{
  "error": "Unable to delete 1 volume",
   {
        "id": 0,
        "name": "SNAPVOLUMES\test_user",
        "name_html": "string",
        "title": "SNAPVOLUMES\test_user",
        "title_html": "string",
        "owner": "string",
        "link": "/directory#/Users/2",
        "owner_name": "TestUser",
        "owner_type": "User",
        "owner_upn": "SNAPVOLUMES\test_user",
        "owner_upn_html": "SNAPVOLUMES\test_user",
        "description": "string",
        "created_at": "string",
        "created_at_human": "string",
```

```
        "updated_at": "string",
        "updated_at_human": "string",
        "mounted_at": "string",
        "mounted_at_human": "string",
        "mounted_on": "Machine <DESKTOP-VM123> (#{1234567812345678})",
        "attached": "Attached",
        "status": "missing",
        "version_count": 0,
        "version_tag": 0,
        "block_login": true,
        "enable_version": true,
        "mount_prefix": "string",
        "defer_create": true,
        "size_mb": 0,
        "template_version": "string",
        "datastore_name": "SSD Disk Array RAID8",
        "machine_manager_host": "string",
        "machine_manager_type": "string",
        "path": "cloudvolumes/writable",
        "filename": "snapvolumes_test_user_15.vmdk",
        "file_location": "string",
        "mount_count": 0,
        "type": "string",
        "display_type": "string",
        "template_file_name": "string",
        "protected": true,
        "free_mb": 0,
        "total_mb": 0,
        "percent_available": "string",
        "can_expand": true,
        "storage_group": "string",
        "storage_group_members": "string",
        "primordial_os_id": 0,
        "primordial_os_name": "string",
        "oses": [
          {}
        ]
      }
    ],
     }
```

For example: When a Writable Volume deletion is scheduled:

```
"scheduled": [
  {
         "id": 0,
       "name": "SNAPVOLUMES\test_user",
      "name_html": "SNAPVOLUMES\test_user",
      "title": "SNAPVOLUMES\test_user",
      "title_html": "SNAPVOLUMES\test_user",
        "owner": "<a href=\\\"/directory#/Users/2\\\" title=\\\"test_use
r\\\">TESTDOMAIN\\\\test_user</a>",
      "link": "/directory#/Users/2",
      "owner_name": "TestUser",
      "owner_type": "User",
      "owner_upn": "SNAPVOLUMES\test_user",
      "owner_upn_html": "SNAPVOLUMES\test_user",
```

```
      "description": "My writable volume", "created_at": "2016-11-23 13:1
0:13 -0800",
      "created_at_human":
      "Nov 23 2016",
       "updated_at": "2016-11-23 13:10:13 -0800",
      "updated_at_human": "Nov 23 2016",
      "mounted_at": "2016-11-23 13:10:13 -0800",
      "mounted_at_human": "Nov 23 2016",
     "mounted_on": "Machine <DESKTOP-VM123>
      (#{1234567812345678})",
      "mount_count": 0,
      "attached": "Attached",
      "status": "enabled",
      "version_count": 0,
      "version_tag": 0,
      "block_login": true,
       "error_action": "",
       "enable_version": false,
      "mount_prefix": "DESKTOP",
      "defer_create": true,
      "size_mb": 0,
      "template_version": "2.10.0.709",
      "datastore_name": "SSD Disk Array RAID8",
      "machine_manager_host": "0.0.0.0",
      "machine_manager_type": "Machine Manager",
      "path": "cloudvolumes/writable",
      "filename": "snapvolumes_test_user_15.vmdk",
      "file_location": "[SSD Disk Array RAID8]
      cloudvolumes/writable/snapvolumes_test_user_15.vmdk",
      "template_file_name": "[SSD Disk Array RAID8] cloudvolumes/writabl
e_templates/template_uia_only_persistent.vmdk",
      "protected": true,
      "free_mb": 0,
      "total_mb": 0,
      "percent_available": "Unknown",
       "can_expand": true,
       "storage_group":
      "Storage group 1",
      "storage_group_members": 2,
      "primordial_os_id": 15,
      "primordial_os_name": "Windows 8.1 (x64)",
      "oses": [
      {
      "id": 0,
      "name": "Windows 8.1 (x64)"
      }
    ],
   "busy": false,
   "state": "Moving"
 }
]
}
 }
```


For example: If a Writable Volume does not exist, then an HTTP error code 404 is returned with an error message:

```
{
    "error": "Unable to delete 1 volume because record does not exist",
    "snapvols": {
        "not_found": [
            220
        ],
        "success": [],
        "error": [],
        "scheduled": []
    }
}
```