VHQ API
Reference Guide

Verifone®

Verifone Part Number DOC888-006-EN-E.2
VHQ API Reference Guide

This document is confidential to Verifone and is provided for informational purposes only and may only be used by the Customer in connection with its use of Verifone's VHQ Estate Manager product. Nothing herein shall create or be construed as constituting any warranty for any Verifone products or services. Verifone shall not be liable for technical or editorial errors or omissions contained herein. The information contained herein is subject to change without notice.

Revision History

Date	Version	Author	Description of change
03-June-2020	Revision A	VPD	Document created.
30-Nov-2021	Revision B.1	VPD	Document amendments for V1.4 and internal review.
03-Dec-2021	Revision B.2	VPD	Document amendments for V1.4 and internal review.
12-Dec-2021	Revision B.3	VPD	Document amendments for V1.4 and internal review.
15-Dec-2021	Revision B	VPD	Released for V1.4.
26-Feb-2022	Revision C	VPD	Document amendments for V1.4 and internal review.
17-May-2022	Revision D	VPD	Document amendments for V1.5, V1.4 and V1.3, and internal review.
18-Oct-2022	Revision E	VPD	Document amendments for NEW API Releases/Changes for 3.20.01.

TABLE OF CONTENTS

PREFACE

VHQ is a powerful and dynamic estate management solution that transforms estate challenges into dynamic opportunities.

The device establishment, or “registration”, is the first step in establishing the device information in the VHQ Server. While boarding the devices onto the VHQ, you can place the devices under the required hierarchy. Hierarchies are the basis of device organization. You can configure applications that run on a device. The model that a device belongs to determines the type of application that needs to configure for the device. Applications on the device comprise a set of executables and configuration files. Configuration files store the application parameters. The Parameter Management feature makes it easier for the user to configure, edit, download, and manage the values of these application parameters. The download feature allows you to schedule application and parameter downloads.

This guide is your primary source of information for configuring the devices and hierarchies on the VHQ using function calls that are listed in the Swagger tool. It also covers the steps to get the downloads, get parameters to download, and the application downloads using the Swagger tool.

AUDIENCE

This guide is useful for the technical team, which uses the VHQ APIs.

ORGANIZATION

This guide is organized as follows:

1. Chapter 1, **Introduction to the Swagger Tool and Usage**. Provides the steps to get started with the Swagger tool.
2. Chapter 2, **Configuration**. Provides the steps to configure devices, hierarchies, and packages for a device on the VHQ through APIs using the Swagger tool.
3. Chapter 3, **Verifone Support**. Provides contact information of the VCS/Support team.

INTRODUCTION TO SWAGGER TOOL AND USAGE

Swagger is primarily an API documentation tool that helps to design, build, document, and consume RESTful web services. While most users identify Swagger by the Swagger UI tool, the Swagger toolset includes support for automated documentation, code generation, and test-case generation.

VHQ APIs allow you to perform the following tasks using the Swagger tool:

- configure devices and hierarchies.
- fetch the details like models, applications assigned for downloads, application parameters, and assigned downloads.

This chapter provides the steps to get started with the Swagger tool and VHQ APIs.

AUTHENTICATION/AUTHORIZATION

This section details two ways of authentication:

1. Authorizing the Swagger Tool to authenticate with VHQ APIs.
2. Programmatically authenticating with VHQ APIs.

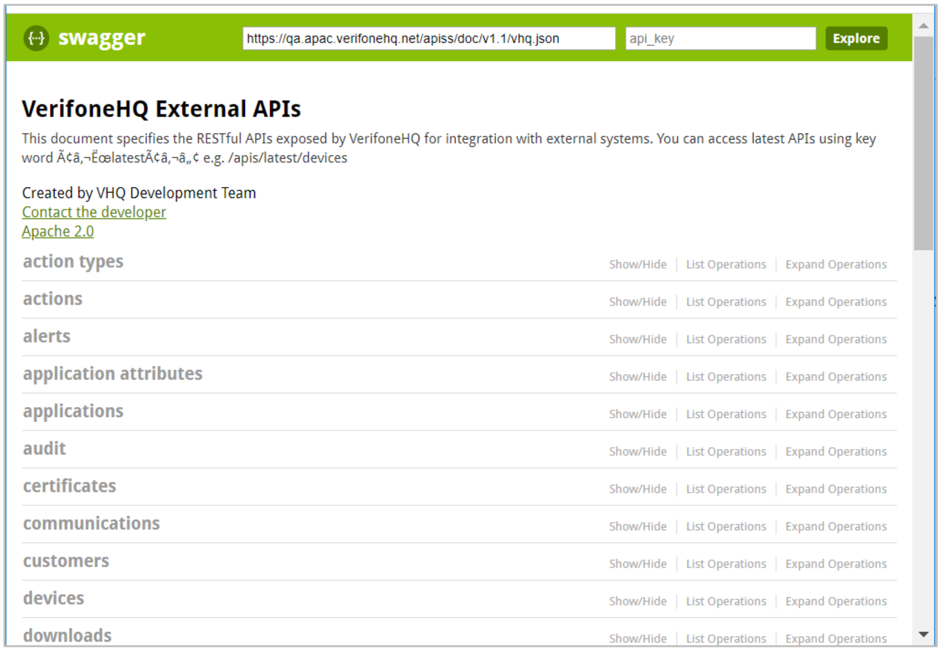
AUTHORIZING SWAGGER TO AUTHENTICATE WITH VHQ APIS

Following are the steps to authorize the Swagger tool using the OAuth2.0 protocol to access the VHQ APIs:

1. Click the link to open the Swagger tool: <https://qa.apac.verifonehq.net/Swagger/>

NOTE	Contact the **Verifone Support team** to get the user-specific URL.

2. The Swagger UI is displayed.



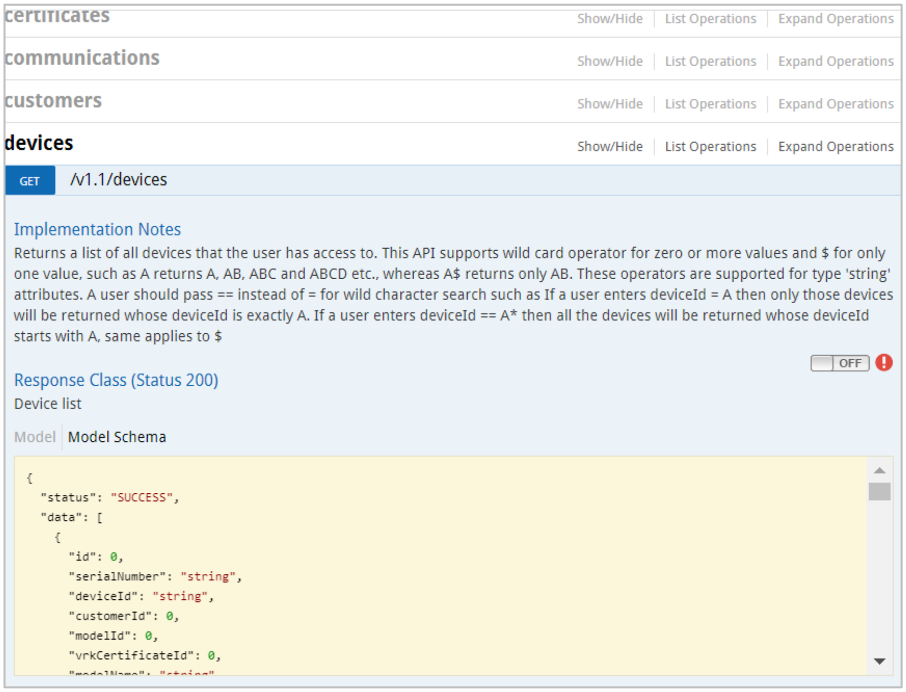
VerifoneHQ External APIs

This document specifies the RESTful APIs exposed by VerifoneHQ for integration with external systems. You can access latest APIs using key word Ã¢â‚¬ËœlatestÃ¢â‚¬â„¢ e.g. /apis/latest/devices

Created by VHQ Development Team
Contact the developer
Apache 2.0

action types — Show/Hide | List Operations | Expand Operations
actions — Show/Hide | List Operations | Expand Operations
alerts — Show/Hide | List Operations | Expand Operations
application attributes — Show/Hide | List Operations | Expand Operations
applications — Show/Hide | List Operations | Expand Operations
audit — Show/Hide | List Operations | Expand Operations
certificates — Show/Hide | List Operations | Expand Operations
communications — Show/Hide | List Operations | Expand Operations
customers — Show/Hide | List Operations | Expand Operations
devices — Show/Hide | List Operations | Expand Operations
downloads — Show/Hide | List Operations | Expand Operations

3. Select the option sequence **Devices-> Get()** API.


certificates
Show/Hide | List Operations | Expand Operations
communications
Show/Hide | List Operations | Expand Operations
customers
Show/Hide | List Operations | Expand Operations
devices
Show/Hide | List Operations | Expand Operations
GET
/v1.1/devices
OFF

Implementation Notes

Returns a list of all devices that the user has access to. This API supports wild card operator for zero or more values and $ for only one value, such as A returns A, AB, ABC and ABCD etc., whereas A$ returns only AB. These operators are supported for type 'string' attributes. A user should pass == instead of = for wild character search such as If a user enters deviceId = A then only those devices will be returned whose deviceId is exactly A. If a user enters deviceId == A* then all the devices will be returned whose deviceId starts with A, same applies to $

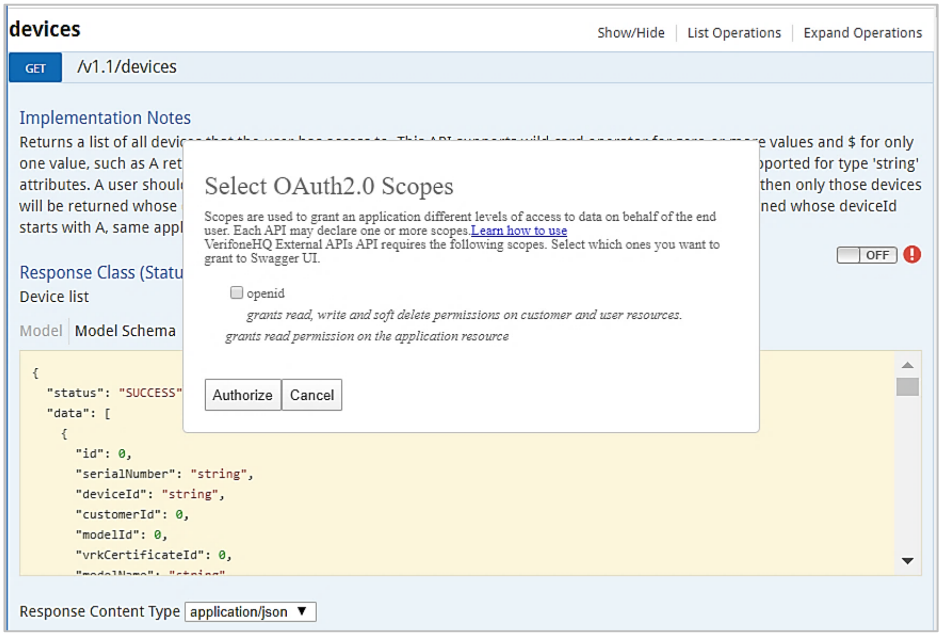
Response Class (Status 200)

Device list

Model | Model Schema

```
{
  "status": "SUCCESS",
  "data": [
    {
      "id": 0,
      "serialNumber": "string",
      "deviceId": "string",
      "customerId": 0,
      "modelId": 0,
      "vrkCertificateId": 0,
```

4. Select **ON** option from the **ON/OFF** toggle button, which is located on the right-hand side of the pane.



Implementation Notes

Returns a list of all devic… values and $ for only one value, such as A ret… pported for type 'string' attributes. A user shoul… then only those devices will be returned whose … ned whose deviceId starts with A, same appl…

Select OAuth2.0 Scopes

Scopes are used to grant an application different levels of access to data on behalf of the end user. Each API may declare one or more scopes.Learn how to use

VerifoneHQ External APIs API requires the following scopes. Select which ones you want to grant to Swagger UI.

□ openid

grants read, write and soft delete permissions on customer and user resources.

grants read permission on the application resource

Response Class (Statu…

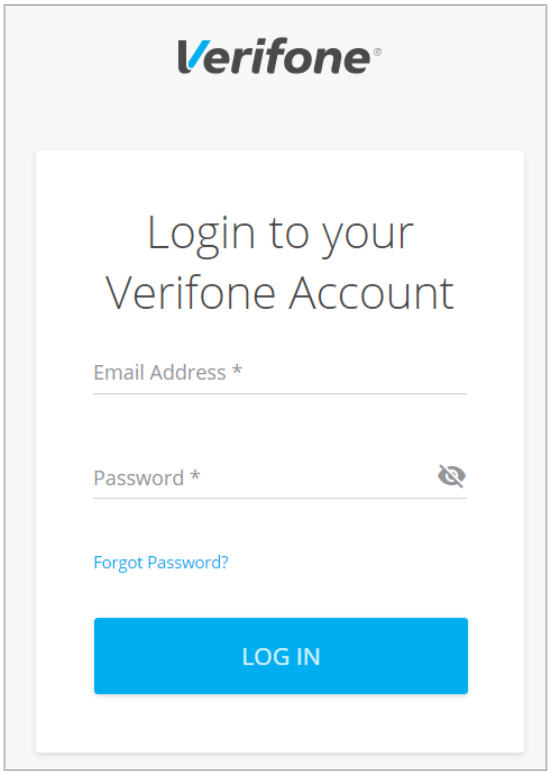
Device list

Model | Model Schema

```
{
  "status": "SUCCESS"
  "data": [
    {
      "id": 0,
      "serialNumber": "string",
      "deviceId": "string",
      "customerId": 0,
      "modelId": 0,
      "vrkCertificateId": 0,
```

Response Content Type application/json ▼

5. Select **OpenID** checkbox and click **Authorize** button. The Verifone SSO (Single Sign-On) screen is displayed.



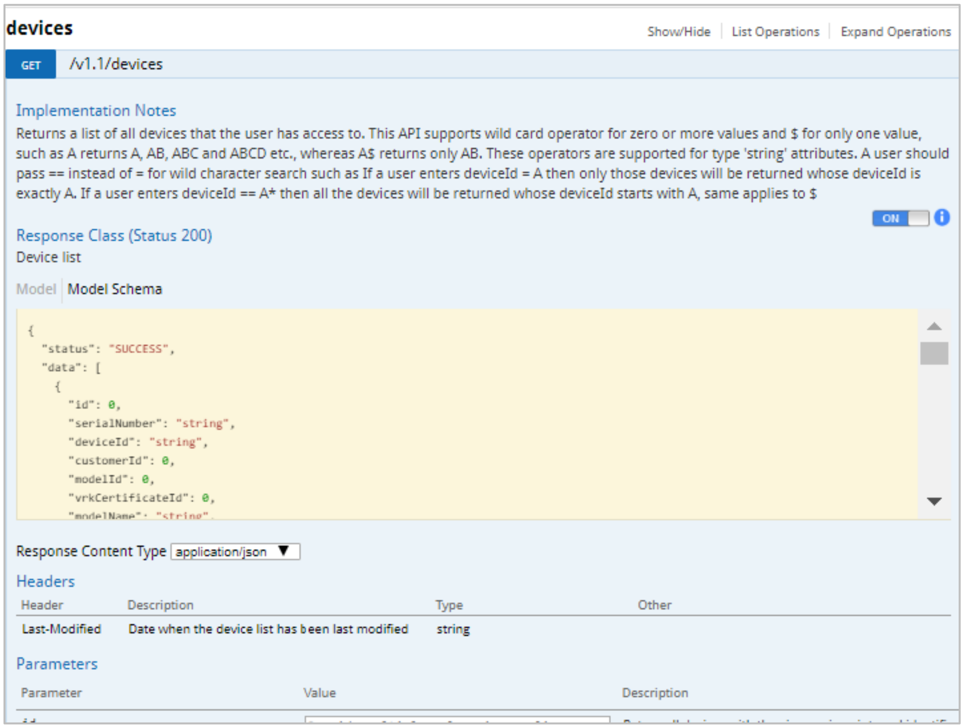
Verifone

Login to your Verifone Account

Enter the username and password provided by the Verifone Support Team.

6. The main screen is displayed to view the function calls.

NOTE	The login credentials (username and password) are provided by the Verifone Support Team.



Implementation Notes

Returns a list of all devices that the user has access to. This API supports wild card operator for zero or more values and $ for only one value, such as A returns A, AB, ABC and ABCD etc., whereas A$ returns only AB. These operators are supported for type 'string' attributes. A user should pass == instead of = for wild character search such as If a user enters deviceId = A then only those devices will be returned whose deviceId is exactly A. If a user enters deviceId == A* then all the devices will be returned whose deviceId starts with A, same applies to $

Response Class (Status 200)

Device list

```
{
  "status": "SUCCESS",
  "data": [
    {
      "id": 0,
      "serialNumber": "string",
      "deviceId": "string",
      "customerId": 0,
      "modelId": 0,
      "vrkCertificateId": 0,
      "modelName": "string",
```

Response Content Type application/json ▼

Headers

Header	Description	Type	Other
Last-Modified	Date when the device list has been last modified	string	

Parameters

Parameter	Value	Description

PROGRAMMATICALLY AUTHENTICATING THE VHQ APIS

The user can programmatically authenticate VHQ APIs using the Client ID and the Secret key provided by the **Verifone Support Team**. The Client ID and the Secret key should be encoded using base64decode.org.

NOTE	The Client ID and the Secret key are provided by the Verifone Support Team.

Sample Code

SAMPLE CODE SNIPPET

```
public class TokenGenerate {

    private static String url = "https://qa.account.verifonecp.com/oauth2/token";
    private static String base64EncodedClientIDAndSecret = "Basic
cktUT3ZZYjhXNjZGbEFzOWZDZEtmVDZBM0hBYTpxVndnb3FQSloxV3RDVVBSR1dJZEZ3SjJxNVlhOg==";

/*to get base64EncodedClientIDAndSecret, ClientID and Secret key should be encoded using
https://www.base64decode.org/ .
*/
    private static String postParameter = "grant_type=client_credentials&scope=openId";
    private int statusCode;

    HttpURLConnection getConnection(URL completeURL, String method) throws IOException{
        HttpURLConnection httpCon = (HttpURLConnection) completeURL.openConnection();
        httpCon.setDoOutput(true);
        httpCon.setRequestProperty("Content-Type", "application/x-www-form-urlencoded");
        httpCon.setRequestProperty("Authorization", base64EncodedClientIDAndSecret);
        return httpCon;
    }
    String getToken(){
        StringBuffer response = null;
        try{
            URL completeURL = new URL(url);
            HttpURLConnection httpCon = getConnection(completeURL, "POST");
            OutputStream os = httpCon.getOutputStream();
            os.write(postParameter.getBytes());
            os.flush();
            if (httpCon.getResponseCode() == HttpURLConnection.HTTP_OK) {
                BufferedReader in = new BufferedReader(new
InputStreamReader(httpCon.getInputStream()));
                String inputLine;
                response = new StringBuffer();
                while ((inputLine = in.readLine()) != null) {
                    response.append(inputLine);
                }
                System.out.println("Response  :::  " +response);
                System.out.println(httpCon.getResponseCode());
                System.out.println(httpCon.getResponseMessage());
            }
        }
        catch (Exception ex) {
            ex.printStackTrace();
            if (ex.getMessage().equalsIgnoreCase("404 Not Found")) {
                statusCode = 404;
                throw new RuntimeException("Not Found " + statusCode);
            }
            else if (ex.getMessage().equalsIgnoreCase("400 Bad Request")) {
                statusCode = 400;
                throw new RuntimeException("Bad Request " + statusCode);
            }
        }
        return response.toString();
    }
    public static void main(String[] args) {
        TokenGenerate =new TokenGenerate();
        tokenGenerate.getToken();
    }

}
        return id;
    }
    public static void main(String[] args) {
        //Post method will accept two-parameter, Request Body and API
        System.out.println(":::::::: Create Device calling start ::::::::");
        CreateDeviceSample createDeviceSample = new CreateDeviceSample();
        createDeviceSample.post(postparameter, "devices");
        System.out.println(":::::::: Create Device calling End ::::::::");
    }
}
```

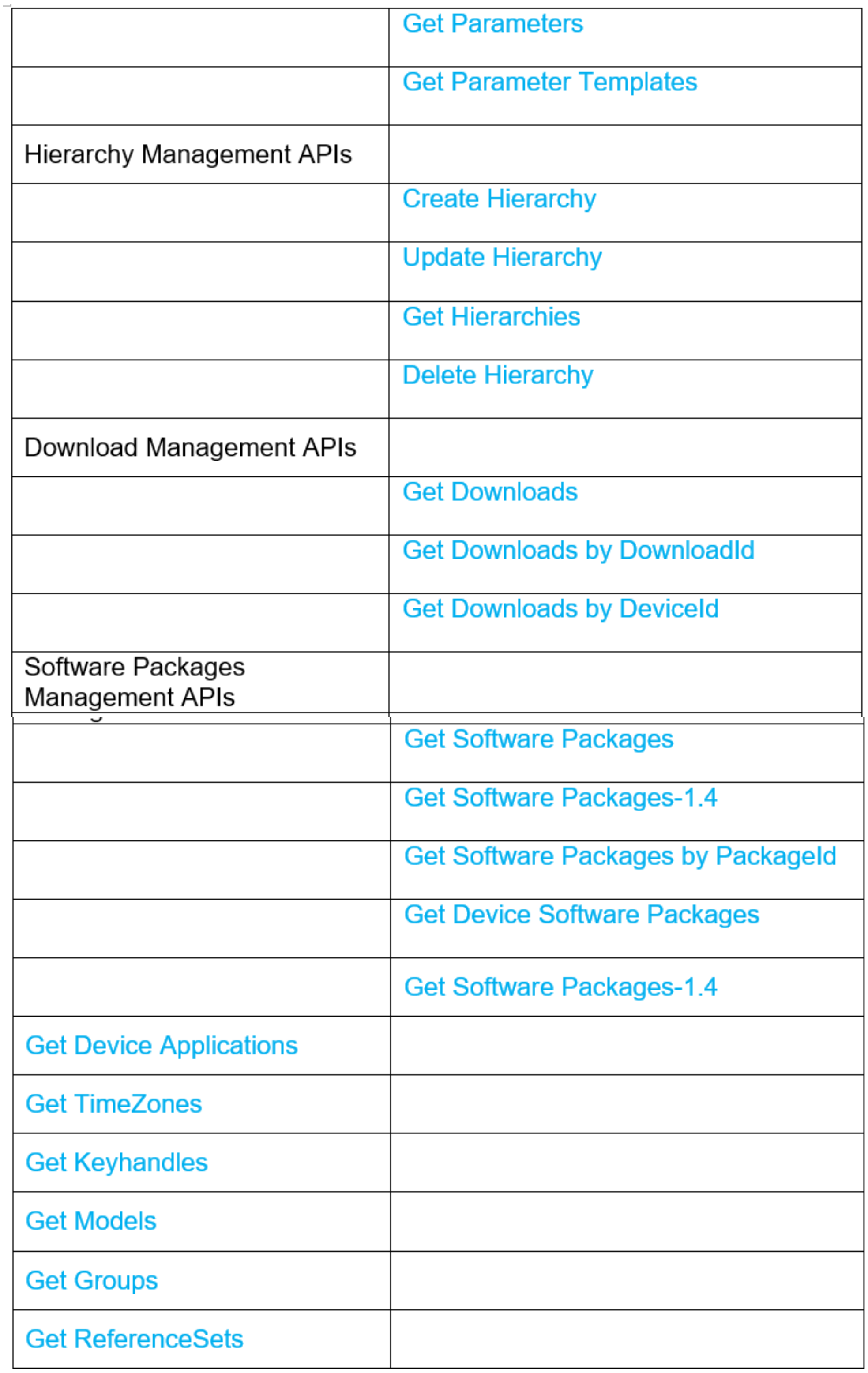
VHQ APIs

This chapter explains the steps to perform the following activities using the VHQ APIs with the help of the Swagger tool:

- configure devices and hierarchies.
- fetch the details like models, applications assigned for downloads, application parameters, and assigned downloads.

Following are the VHQ APIs:

Device Management APIs	
	Create Device
	Update Device
	Update Devices-1.4
	Get Devices
	Get Devices-1.4
	Delete Device
Parameter Management APIs	
	Create Parameter
	Update Parameter
	Delete Parameter



	Get Parameters
	Get Parameter Templates
Hierarchy Management APIs	
	Create Hierarchy
	Update Hierarchy
	Get Hierarchies
	Delete Hierarchy
Download Management APIs	
	Get Downloads
	Get Downloads by DownloadId
	Get Downloads by DeviceId
Software Packages Management APIs	
	Get Software Packages
	Get Software Packages-1.4
	Get Software Packages by PackageId
	Get Device Software Packages
	Get Software Packages-1.4
Get Device Applications	
Get TimeZones	
Get Keyhandles	
Get Models	
Get Groups	
Get ReferenceSets	

The severity level in the table indicates the severity of the error based on the impact the error has on the end-user operation. Following are the severity codes for VHQ APIs:

0	Emergency	The system is unusable.
1	Alert	Action must be taken immediately.
2	Critical	Critical conditions.
3	Error	Error conditions.
4	Warning	Warning conditions.
5	Notice	Normal but significant condition.
6	Information	Information.
7	Debug	Debug-level messages.

WHAT'S NEW

The following APIs are updated.

GET DEVICES-1.5

This API is used to get the list of all devices except deleted and blacklisted devices under the root hierarchy that are accessible to the logged-in users. This API supports Pagination.

This API is supported for the following versions: V1.1, V1.2, V1.3 and V1.5.

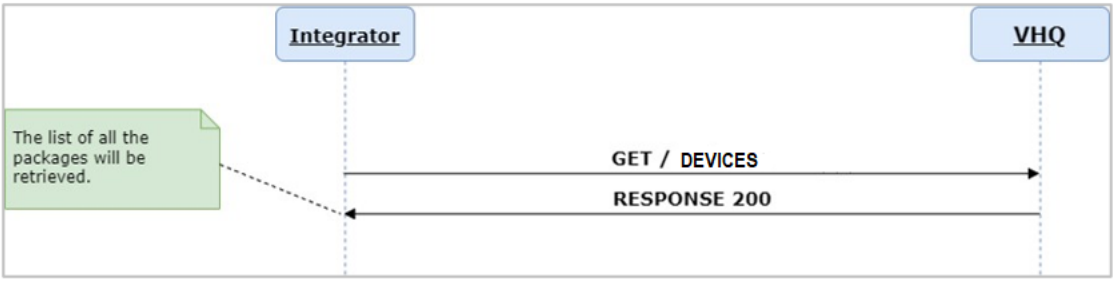
Get Devices API

To get the list of all device details such as: Device ID, Serial Number, Model ID, Model Name, Status, Sub-status, IP Address, MAC Address, and Location ID.

Go to the following path:

```
API - GET/devices

URL - https://<API Server>/apis/v1.5/devices

https://vhqint.verifonehq.net/apis/v1.5/devices/?customerId=3&paymentAppName==a*

https://vhqint.verifonehq.net/apis/v1.5/devices/?customerId=3&paymentAppVersion==7*

https://vhqint.verifonehq.net/apis/v1.5/devices/?customerId=3&paymentAppVersion==7.1
*&paymentAppName==com.*

e.g.  https://vhqtest.verifone.com/apis/v1.5/devices
```

Sequence Diagram



Input Parameters

Input parameters must be sent in the query string.

Element	Optional/Mandatory	Data Type	Description
serialNumber	Optional	String	The serial number of the device.
modelId	Optional	Int	The model ID of the device.
deviceId	Optional	String	Device Identification.
hierarchyId/hierarchyname	Optional	Int	Hierarchy Id.
groupId/groupName	Optional	Int	Group Id.
applications	Optional	String	The array of applications. Value format with Application = {Application Name}: {Version}.
reportedApplications	Optional	String	The array of reported applications. Value format with Application = {Application Name}: {Version}.
packageName	Optional	String	The array of packages.
limit	Optional	Int	Rows per request.
offset	Optional	Int	Offset required to calculate the page number.

Output Parameters

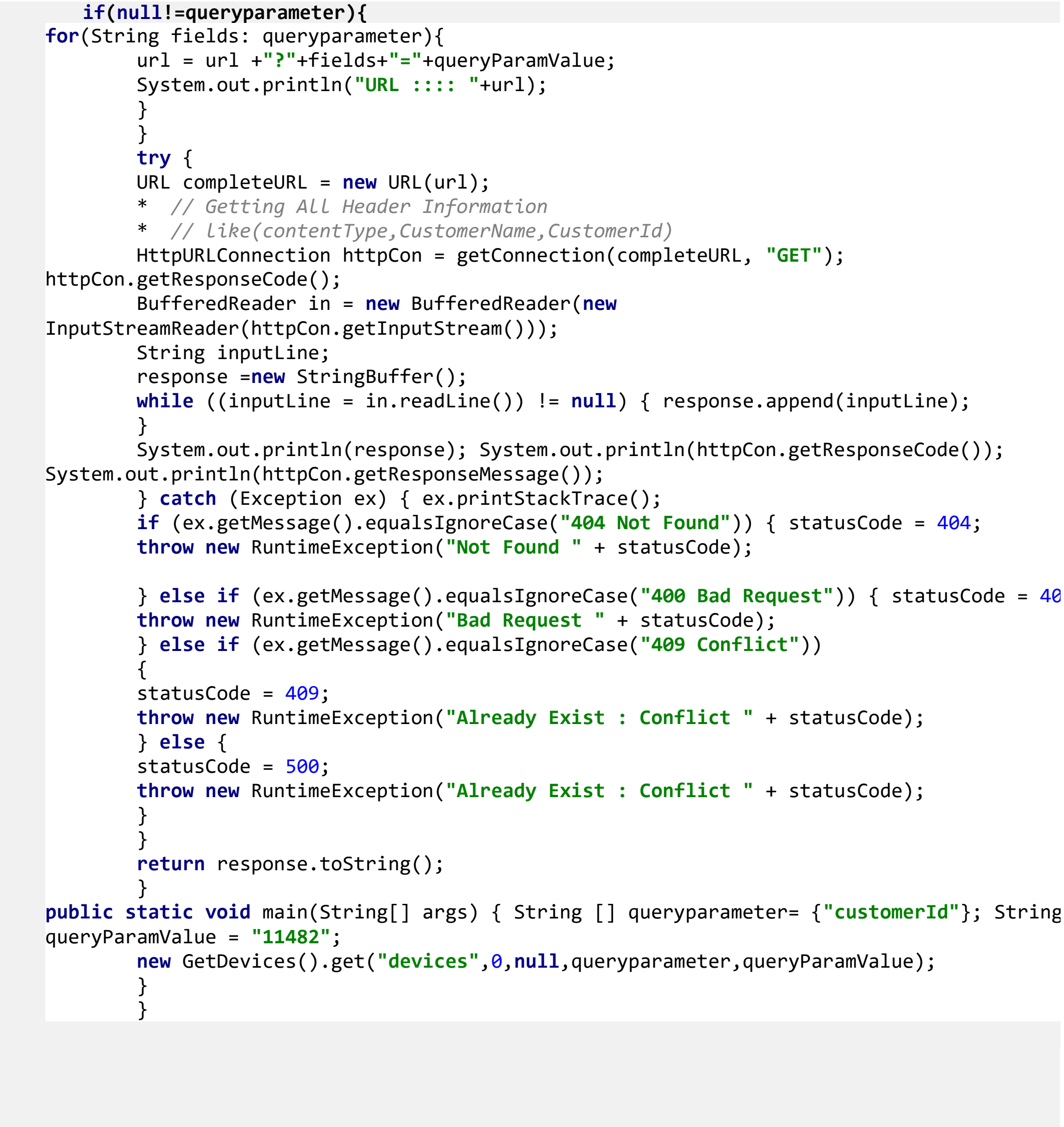
Element	Attribute	Data Type
devices (collection of devices)	uniqueDeviceId	Int
	serialNumber	String
	deviceId	String

	status	String
	model	String
	hierarchyFullPath	String
	locationId	String

Get Devices API - Sample Code

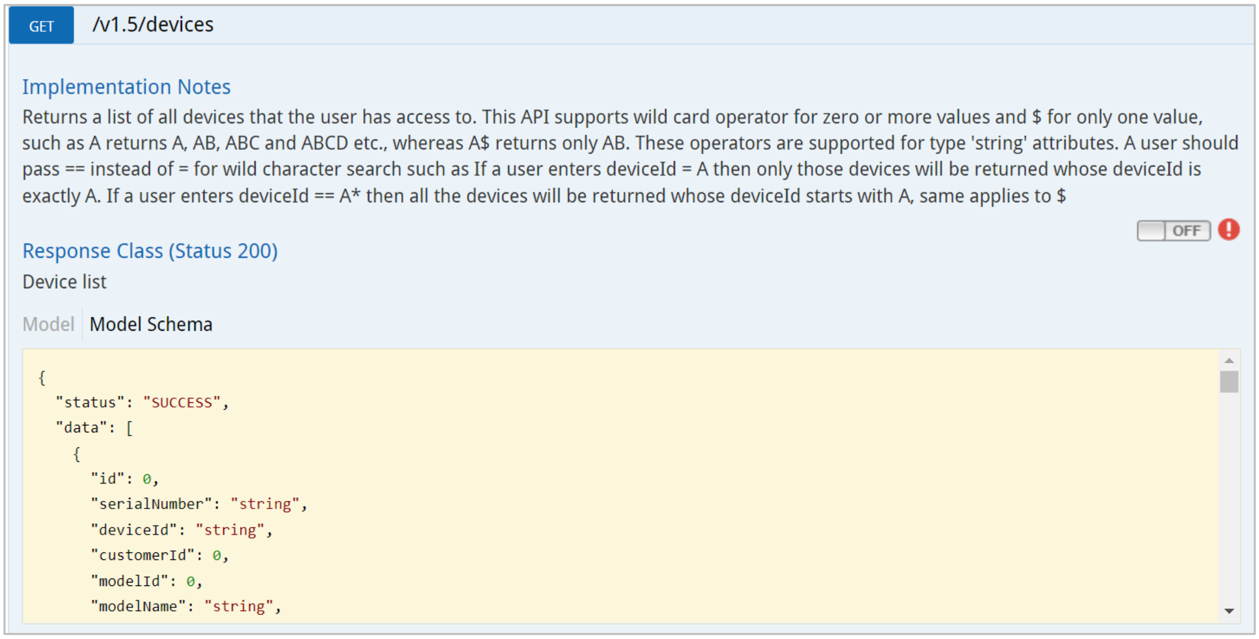
SAMPLE CODE SNIPPET

```
public class GetDevices {

   private static String url = "http://<servername:port>/apis/v1.5/";
   private int statusCode;
   // This is the base URL, it will be appended as per the entity which is invoked

   /**
    * @param completeURL -> This is the URL which appended as per the
   entity which is invoked
    * @param method -> This refer to which method we are going to
   send e.g. get,post,put
    * @return -> httpCon with Content Type and Authorization token
    * @throws IOException
    */
   HttpURLConnection getConnection(URL completeURL, String method)
         throws IOException {
      HttpURLConnection httpCon = (HttpURLConnection)
            completeURL.openConnection();
      httpCon.setDoOutput(true);
      httpCon.setRequestProperty("Authorization", "Bearer 29797eae-3879-3827-8ec3-
5295796c5af1");
      httpCon.setRequestMethod(method);
      httpCon.setRequestProperty("Content-Type", "application/json");
      return httpCon;
   }
   /**
    * This method get all data of a entity
    * @param API -> This refers to the endpoint of the URL for the
   particular entity e.g. for Device entity it is devices
    * @param id -> This refer to show specific data from Entity based
   on Id.
    * @param fields -> This refers to show specific fields, which we
   pass as a parameter.
    * @param queryparameter
    * @return response from entity
    */
   String get(String Api, int id,String fullPath,String[] queryparameter,String
queryParamValue) {
      StringBuffer response=null;
      if(id!=0) {
         url = url.concat(Api) + "/" + id;
      }else {
         url = url.concat(Api);
      }

      if(fullPath!=null){ url=url+"/".concat(fullPath
         );
      }
```



```
    if(null!=queryparameter){
for(String fields: queryparameter){
        url = url +"?"+fields+"="+queryParamValue;
        System.out.println("URL :::: "+url);
        }
        }
        try {
        URL completeURL = new URL(url);
        *  // Getting All Header Information
        *  // like(contentType,CustomerName,CustomerId)
        HttpURLConnection httpCon = getConnection(completeURL, "GET");
httpCon.getResponseCode();
        BufferedReader in = new BufferedReader(new
InputStreamReader(httpCon.getInputStream()));
        String inputLine;
        response =new StringBuffer();
        while ((inputLine = in.readLine()) != null) { response.append(inputLine);
        }
        System.out.println(response); System.out.println(httpCon.getResponseCode());
System.out.println(httpCon.getResponseMessage());
        } catch (Exception ex) { ex.printStackTrace();
        if (ex.getMessage().equalsIgnoreCase("404 Not Found")) { statusCode = 404;
        throw new RuntimeException("Not Found " + statusCode);

        } else if (ex.getMessage().equalsIgnoreCase("400 Bad Request")) { statusCode = 40
        throw new RuntimeException("Bad Request " + statusCode);
        } else if (ex.getMessage().equalsIgnoreCase("409 Conflict"))
        {
        statusCode = 409;
        throw new RuntimeException("Already Exist : Conflict " + statusCode);
        } else {
        statusCode = 500;
        throw new RuntimeException("Already Exist : Conflict " + statusCode);
        }
        }
        return response.toString();
        }
public static void main(String[] args) { String [] queryparameter= {"customerId"}; String
queryParamValue = "11482";
        new GetDevices().get("devices",0,null,queryparameter,queryParamValue);
        }
        }
```

Sample using Swagger – Request



/v1.5/devices

Implementation Notes

Returns a list of all devices that the user has access to. This API supports wild card operator for zero or more values and $ for only one value, such as A returns A, AB, ABC and ABCD etc., whereas A$ returns only AB. These operators are supported for type 'string' attributes. A user should pass == instead of = for wild character search such as If a user enters deviceId = A then only those devices will be returned whose deviceId is exactly A. If a user enters deviceId == A* then all the devices will be returned whose deviceId starts with A, same applies to $

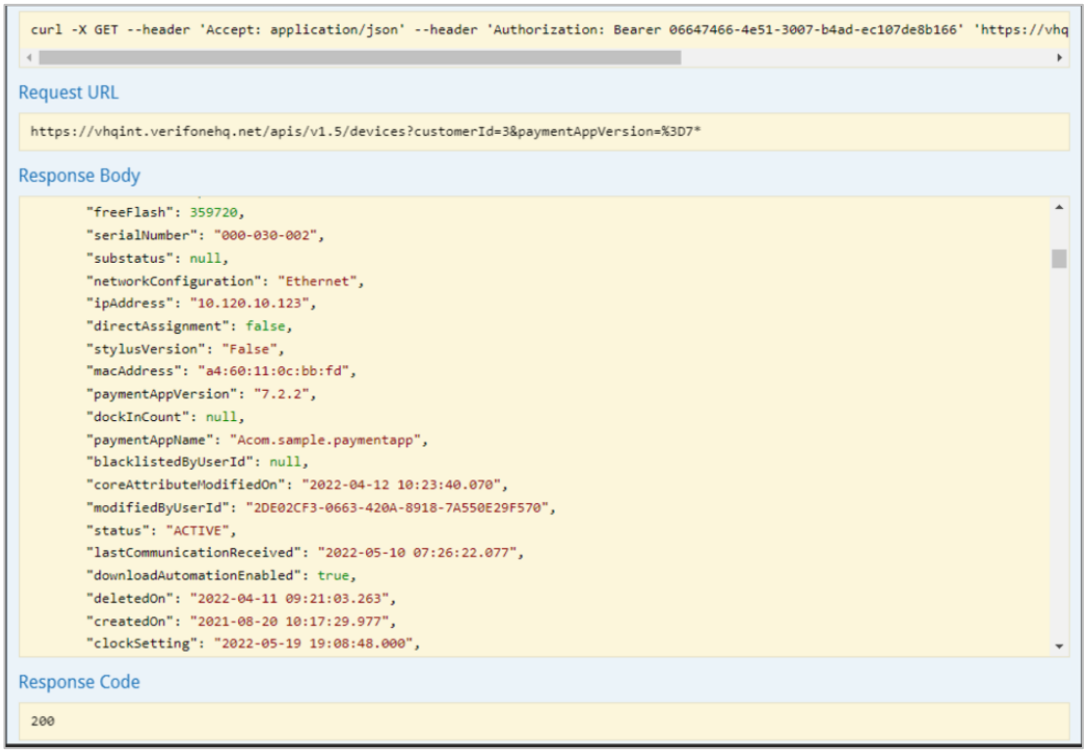
Response Class (Status 200)

Device list

```
{
  "status": "SUCCESS",
  "data": [
    {
      "id": 0,
      "serialNumber": "string",
      "deviceId": "string",
      "customerId": 0,
      "modelId": 0,
      "modelName": "string",
```

Click *Try it out* button to get the device details.

Sample using Swagger – Response



```
curl -X GET --header 'Accept: application/json' --header 'Authorization: Bearer 06647466-4e51-3007-b4ad-ec107de8b166' 'https://vhq
```

Request URL

```
https://vhqint.verifonehq.net/apis/v1.5/devices?customerId=3&paymentAppVersion=%3D7*
```

Response Body

```
        "freeFlash": 359720,
        "serialNumber": "000-030-002",
        "substatus": null,
        "networkConfiguration": "Ethernet",
        "ipAddress": "10.120.10.123",
        "directAssignment": false,
        "stylusVersion": "False",
        "macAddress": "a4:60:11:0c:bb:fd",
        "paymentAppVersion": "7.2.2",
        "dockInCount": null,
        "paymentAppName": "Acom.sample.paymentapp",
        "blacklistedByUserId": null,
        "coreAttributeModifiedOn": "2022-04-12 10:23:40.070",
        "modifiedByUserId": "2DE02CF3-0663-420A-8918-7A550E29F570",
        "status": "ACTIVE",
        "lastCommunicationReceived": "2022-05-10 07:26:22.077",
        "downloadAutomationEnabled": true,
        "deletedOn": "2022-04-11 09:21:03.263",
        "createdOn": "2021-08-20 10:17:29.977",
        "clockSetting": "2022-05-19 19:08:48.000",
```

Response Code

```
200
```

Error Codes

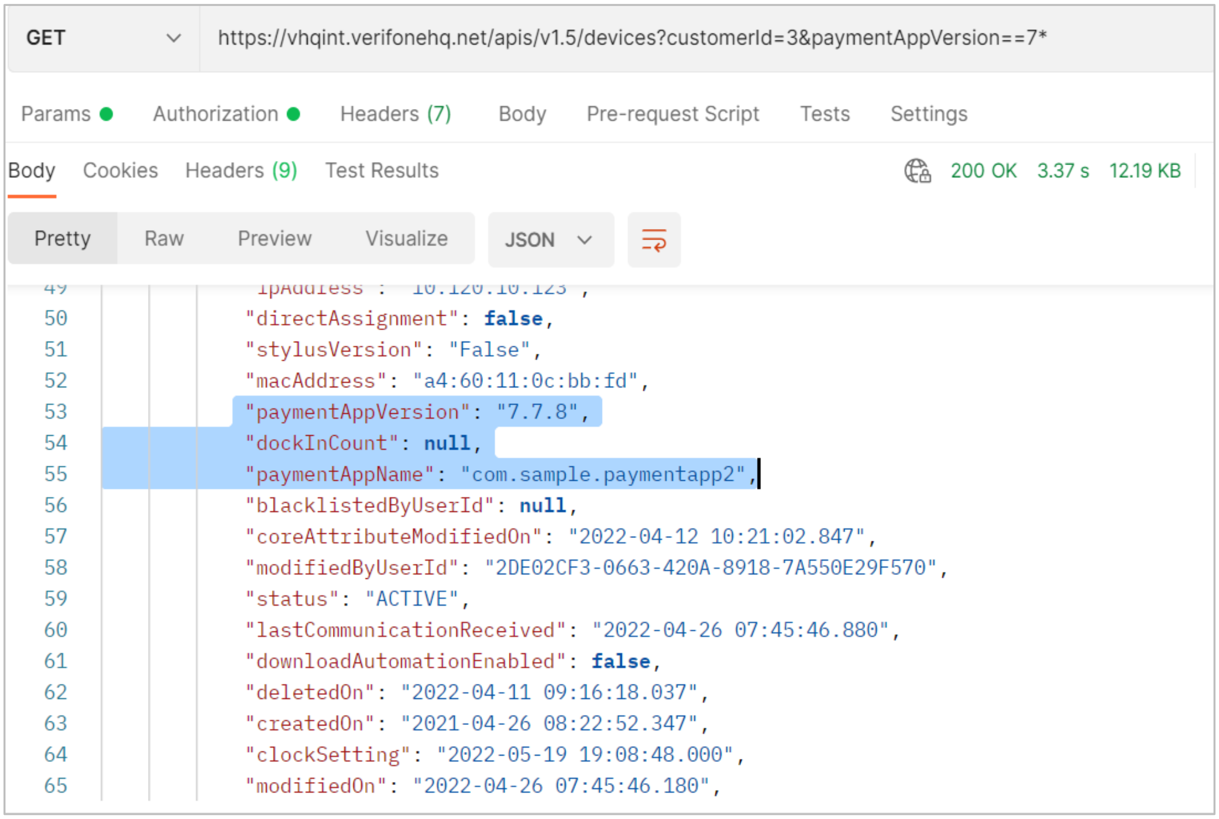
HTTP Status Code	Application Code	Name	Severity	Message
200	S100	S_OK	-	Success.
401	E101	E_TOKEN_INVALID_OR_EXPIRED	2	Token is expired or invalid.
401	E102	E_UNAUTHORIZED_ACCESS	2	The user does not have the right to operate.
500	E103	E_GENERAL_ERROR	3	API failed.

GET Device API also supports **Payment App Name** and **version** as filters.

1. The **GET Device API** with the two parameters **Payment app name** and **Payment appversion** supports V1.5.
2. Wildcard is supported for both the parameters.

paymentAppName and **paymentAppVersion** are the fields introduced in the response fields of GET device API.

Sample Response

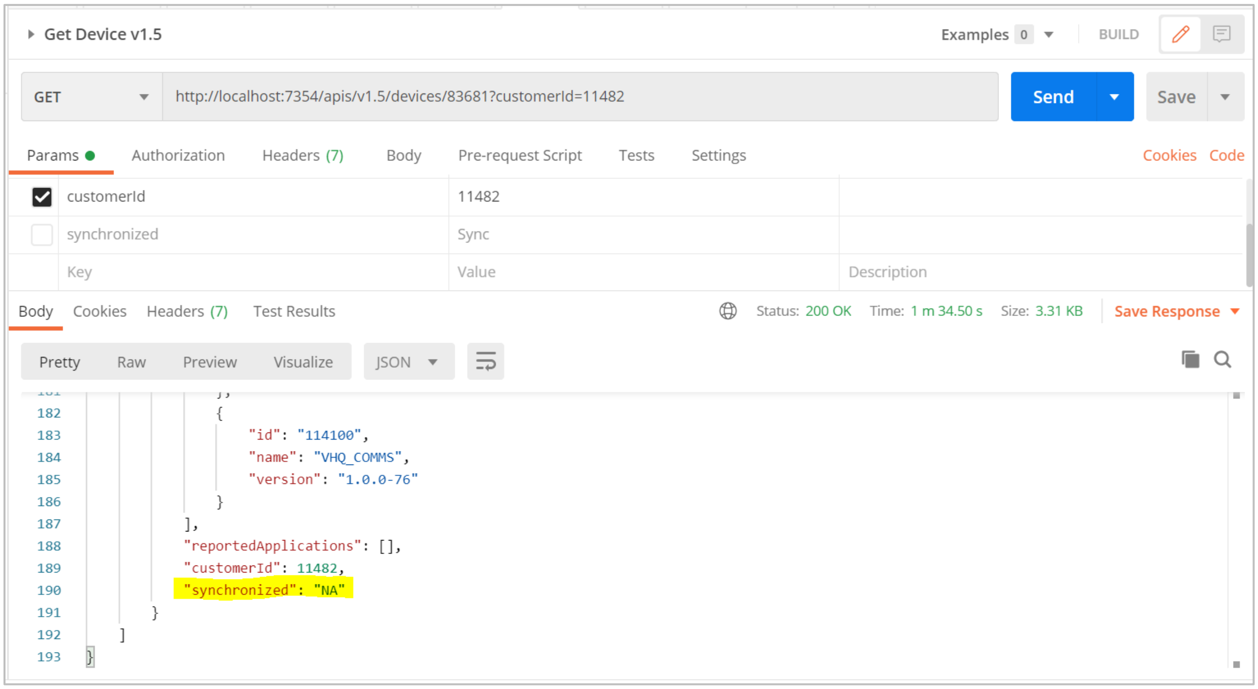


https://vhqint.verifonehq.net/apis/v1.5/devices?customerId=3&paymentAppVersion==7*

```
            "ipAddress": "10.120.10.123",
            "directAssignment": false,
            "stylusVersion": "False",
            "macAddress": "a4:60:11:0c:bb:fd",
            "paymentAppVersion": "7.7.8",
            "dockInCount": null,
            "paymentAppName": "com.sample.paymentapp2",
            "blacklistedByUserId": null,
            "coreAttributeModifiedOn": "2022-04-12 10:21:02.847",
            "modifiedByUserId": "2DE02CF3-0663-420A-8918-7A550E29F570",
            "status": "ACTIVE",
            "lastCommunicationReceived": "2022-04-26 07:45:46.880",
            "downloadAutomationEnabled": false,
            "deletedOn": "2022-04-11 09:16:18.037",
            "createdOn": "2021-04-26 08:22:52.347",
            "clockSetting": "2022-05-19 19:08:48.000",
            "modifiedOn": "2022-04-26 07:45:46.180",
```

Synchronization status: New field, synchronized is introduced to show the synchronization status of software and parameters on VHQ and the device. Valid synchronization statuses are Sync, Not in Sync, NA.

Sample Response



GET http://localhost:7354/apis/v1.5/devices/83681?customerId=11482

	Key	Value	Description
☑	customerId	11482	
□	synchronized	Sync	

Status: 200 OK Time: 1 m 34.50 s Size: 3.31 KB

```
                },
                {
                    "id": "114100",
                    "name": "VHQ_COMMS",
                    "version": "1.0.0-76"
                }
            ],
            "reportedApplications": [],
            "customerId": 11482,
            "synchronized": "NA"
        }
    ]
}
```

CREATE HIERARCHY-1.4

This API is used to add a new hierarchy.

Pre-requisites

Following are the pre-requisites for creating a new hierarchy:

Mandatory Fields:

1. Parent Hierarchy
2. Name of the Hierarchy (User Defined)

Non-Mandatory Fields: Reference Set, Entity Id and Entity type

NOTE: The user can use Reference set name instead of Reference set ID.

Request Body

```
{
  "data": {
    "id": 0,
    "customerId": 0,
    "hierarchyFullPath": "string",
    "parentHierarchyId": 0,
    "parentHierarchyName": 0,
    "name": "string",
    "description": "string",
    "ipStartingAddress": "string",
    "ipEndingAddress": "string",
    "locationIdentifier": "string",
    "timezoneId": 0,
    "entityId": "string",
    “entityType”: “MERCHANT_SITE”,
    "childHierarchies": [
        {
         "id": 0,
         "name": "string"
        }
    ],
    "downloadAutomationEnabled": true,
    "inheritReferenceSet":  true,
    "downloadOn": "NEXT_CONTACT",
    "referenceSets": [
      {
         "id": 0,
         "name": "string"
      }
    ]
  }
}
```

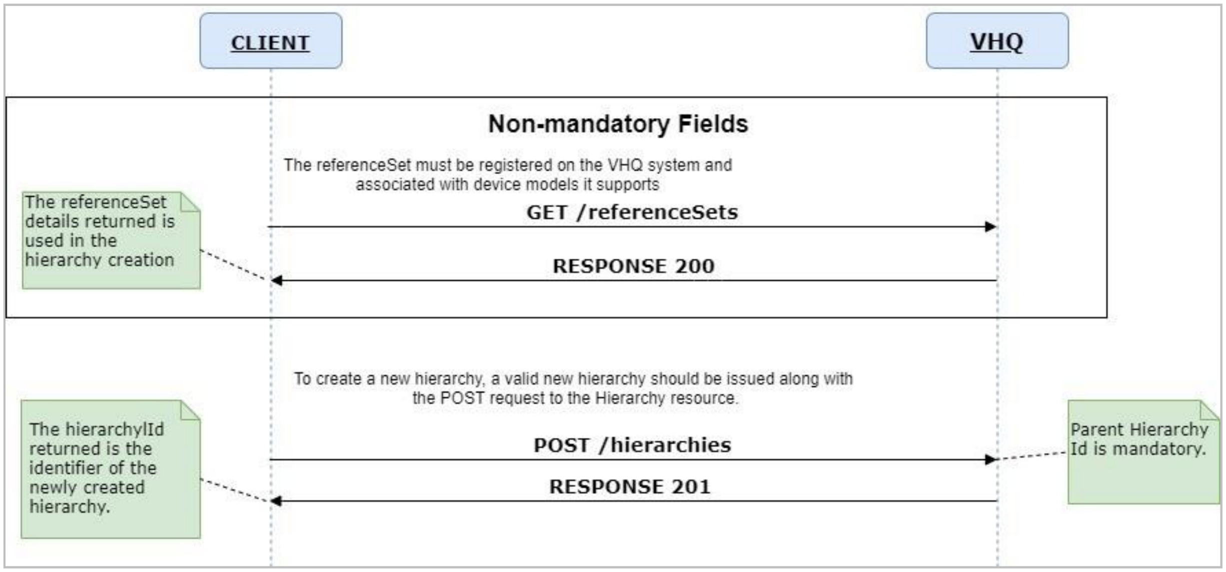
Go to the following path:

API - POST/hierarchies

URL - https://<API Server>/apis/v1.4/hierarchies

e.g. https://vhqtest.verifone.com/apis/v1.4/hierarchies

Sequence Diagram


CLIENT
VHQ
Non-mandatory Fields
The referenceSet must be registered on the VHQ system and associated with device models it supports
GET /referenceSets
RESPONSE 200
The referenceSet details returned is used in the hierarchy creation
To create a new hierarchy, a valid new hierarchy should be issued along with the POST request to the Hierarchy resource.
POST /hierarchies
RESPONSE 201
The hierarchyId returned is the identifier of the newly created hierarchy.
Parent Hierarchy Id is mandatory.

Input Parameters

Element	Optional/Mandatory	Data Type	Description
hierarchyFullPath	Optional	String	The full path of the hierarchy.
parentHierarchyId	Mandatory	Integer	Parent hierarchy identifier.
parentHierarchyName	Optional	String	Parent hierarchy name.
name	Mandatory	String	Name of the hierarchy.
description	Optional	String	Description of the hierarchy.
childHierarchies	Optional	Array	Identifier(s) of immediate child hierarchies.
referenceSets	Optional	Array	Identifier(s) of the reference sets associated with the hierarchy.
downloadOn	Optional	String	Option to indicate when to download.
ipStartingAddress	Optional	String	Starting IP Address of the hierarchy. Optional to add the entity.
ipEndingAddress	Optional	String	Ending IP Address of the hierarchy. Optional to add the entity.
downloadAutomationEnabled	Optional	Boolean	Indicates, if automated downloads should be performed for the devices under this hierarchy.

entityId	Optional	String	Uuid is linked with the hierarchy. Optional to add the entity. Value can be modified after the entity is created.
entityType	Optional	String	Optional to add the entity. Value can be modified after the entity is created. [‘MERCHANT_SITE’, ‘MERCHANT_COMPANY’, ‘UNDEFINED’]
inheritReferenceSet	Optional	Boolean	Indicates, if this hierarchy inherits Reference Set from its parent hierarchy. By default, this is enabled.
locationIdentifier	Optional	String	Location identifier of the hierarchy.

Output Parameters

Element	Data Type	Description
hierarchyId	int	hierarchy Id of the newly created hierarchy.

Create Hierarchy API - Sample Code

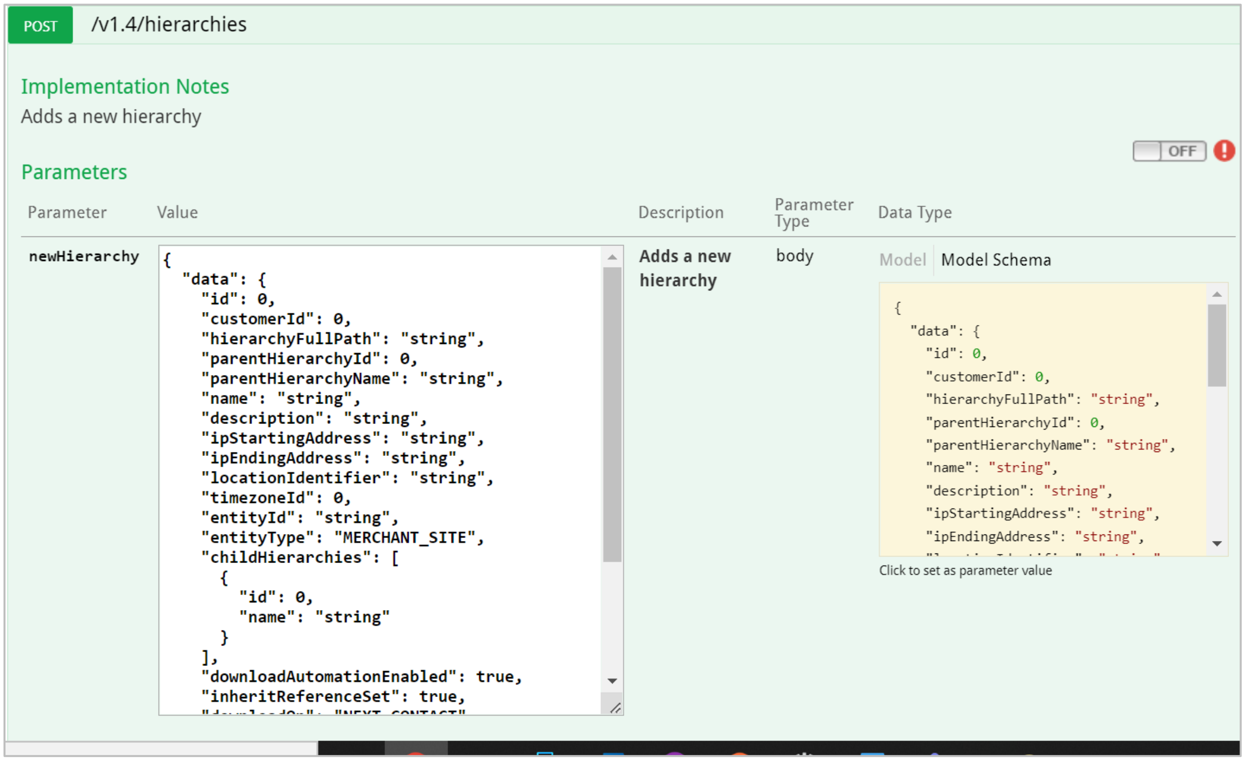
SAMPLE CODE SNIPPET

```
package com.VhqApiSampleCodeTest;
public class CreateHierarchy {
   //Common API URL
   private static String url =
   "http://<servername:port>/apis/v1.3/";
   private int statusCode;
   // Request body
   private static String postHierarchy= "{\r\n" +
         "  \"data\": {\r\n" +
         "     \"customerId\": 11482,\r\n" +
         "     \"name\": \"TestHierarchy_123\",\r\n" +
         "     \"hierarchyFullPath\": \"RATNERCO >> RATNERCO\",\r\n" +
         "     \"parentHierarchyName\": \"AutomationRootHierarchy\",\r\n" +
         "     \"downloadAutomationEnabled\": false,\r\n" +
         "     \"inheritReferenceSet\": true,\r\n" +
         “     \”entityId\”: 1234,\r\n” +
         “     \”entityType\”: “MERCHANT_SITE”,\r\n” +
         "     \"downloadOn\": \"NEXT_CONTACT\",\r\n" +
         "     \"referenceSets\": [{\r\n" +
         "           \"id\": 107082,\r\n" +
         "           \"name\": \"\"\r\n" +
         "        },\r\n" +
         "        {\r\n" +
         "           \"id\": 106463\r\n" +
         "        }\r\n" +
         "     ]\r\n" +
         "  }\r\n" +
         "}";
   /**
    * @param completeURL -> This is the URL which appended as per the entity which is invoked
    * @param method   -> This refer to which method we are going to send e.g. get,post,put
    * @return   -> httpCon with Content Type and Authorization token
    * @throws IOException
    */
   HttpURLConnection getConnection(URL completeURL, String method) throws
      IOException {HttpURLConnection httpCon = (HttpURLConnection)
      completeURL.openConnection(); httpCon.setDoOutput(true);
```

```
          httpCon.setRequestProperty("Authorization", "Bearer 17fee712-031d-3dcd-abd6-
          236a28393210");httpCon.setRequestMethod(method);
          httpCon.setRequestProperty("Content-Type",
          "application/json");
          return httpCon;
       }
         /**
        * This method creates a new entity
        * @param body -> The request body which is required to be posted, This changes as per the entity
        * @param Api -> This refers to the endpoint of the URL for the particular entity e.g. for Hierarchy
    entity it ishierarchies

        */
       String post(String body, String apiEndPoint,String [] queryparameter, String queryParamValue) {
          url =
          url.concat(apiEndPoin
          t);String id = "";
          if(null!=queryparamet
          er){
             for(String fields: queryparameter){
                url = url
                +"?"+fields+"="+queryParamValue;
                System.out.println("URL ::::
                "+url);
             }
          }
          try {
             URL completeURL = new URL(url);


         // Getting All Header Information
         // like(contentType,CustomerName,CustomerId)
         HttpURLConnection httpCon = getConnection(completeURL,
         "POST");OutputStream os = httpCon.getOutputStream();
         os.write(body.getBytes());
         os.flush();


         if (httpCon.getResponseCode() != HttpURLConnection.HTTP_CREATED) {
            if (httpCon.getResponseCode() == 409) {
               throw new RuntimeException(apiEndPoint + "Already Exist : Conflict " + httpCon.getResponseCode());
            }
         }
         for(Entry<String, List<String>> entry: httpCon.getHeaderFields().entrySet()){
            if(null !=entry.getKey()
               &&entry.getKey().toLowerCase().contains("new")){id =
               entry.getValue().get(0);
               id = id.substring(id.lastIndexOf("/")+1);
            }
         }
         System.out.println(id);
         System.out.println(httpCon.getHeaderFields());
         System.out.println("Response Code for POST method-
         >"+httpCon.getResponseCode());System.out.println(httpCon.getResponseMessage());
      } catch (Exception ex)
         {ex.printStackTrace(
         );
         if (ex.getMessage().equalsIgnoreCase("404 Not Found")) {
            statusCode = 404;
            throw new RuntimeException("Not Found " + statusCode);
         } else if (ex.getMessage().equalsIgnoreCase("400 Bad Request")) {
            statusCode = 400;
            throw new RuntimeException("Bad Request " + statusCode);
         } else if (ex.getMessage().equalsIgnoreCase("409 Conflict")) {
            statusCode = 409;
            throw new RuntimeException("Already Exist : Conflict " + statusCode);
         } else {
            statusCode = 500;
            throw new RuntimeException("Already Exist : Conflict " + statusCode);
         }
      }
      System.out.println("Added
      Successfully");return id;
   }
   public static void main(String[] args) {
      System.out.println(":::::::: Create Hierarchy calling start
      ::::::::");
      String [] queryparameter= {"customerId"};
```

```
        String queryParamValue = "11482";
        //Post method will accept two parameter, Request Body and API (Hierarchies)
        CreateHierarchy createHierarchies = new CreateHierarchy();
        createHierarchies.post(postHierarchy,
        "hierarchies",queryparameter,queryParamValue);
    }
}
```

Sample using Swagger - Request



Implementation Notes

Adds a new hierarchy

Parameters

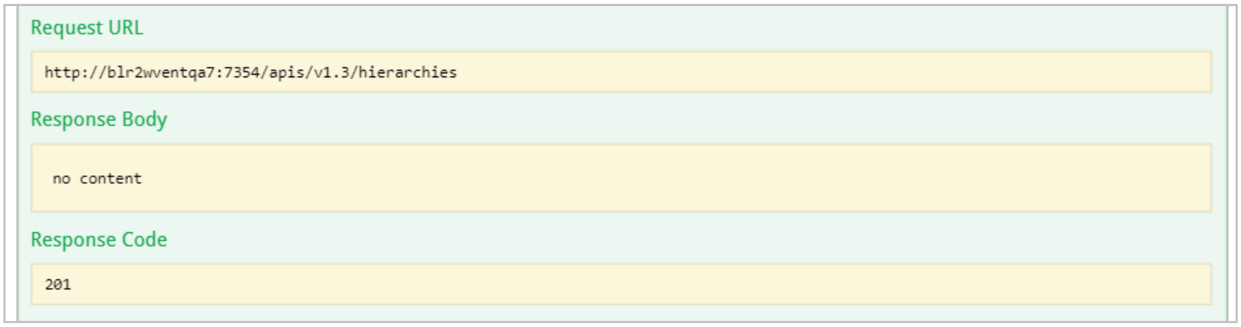
Parameter	Value	Description	Parameter Type	Data Type
newHierarchy		Adds a new hierarchy	body	Model | Model Schema

```
{
  "data": {
    "id": 0,
    "customerId": 0,
    "hierarchyFullPath": "string",
    "parentHierarchyId": 0,
    "parentHierarchyName": "string",
    "name": "string",
    "description": "string",
    "ipStartingAddress": "string",
    "ipEndingAddress": "string",
    "locationIdentifier": "string",
    "timezoneId": 0,
    "entityId": "string",
    "entityType": "MERCHANT_SITE",
    "childHierarchies": [
      {
        "id": 0,
        "name": "string"
      }
    ],
    "downloadAutomationEnabled": true,
    "inheritReferenceSet": true,
    "downloadOn": "NEXT_CONTACT"
```

```
{
  "data": {
    "id": 0,
    "customerId": 0,
    "hierarchyFullPath": "string",
    "parentHierarchyId": 0,
    "parentHierarchyName": "string",
    "name": "string",
    "description": "string",
    "ipStartingAddress": "string",
    "ipEndingAddress": "string",
    "locationIdentifier": "string",
```

Click to set as parameter value

Sample using Swagger - Response



Request URL

```
http://blr2wventqa7:7354/apis/v1.3/hierarchies
```

Response Body

```
no content
```

Response Code

```
201
```

Error Codes

HTTP Status Code	Application Code	Name	Severity	Message
201	S201	S_CREATED	-	Success.

401	E101	E_TOKEN_INVALID_OR_EXPIRED	2	Token expired or invalid.
401	E102	E_UNAUTHORIZED_ACCESS	2	The user does not have the right to perform the operation.
409	E116	E_HIERARCHY_ALREADY_EXISTS	2	Hierarchy already exists
500	E103	E_GENERAL_ERROR	3	API failed.

UPDATE HIERARCHY-1.4

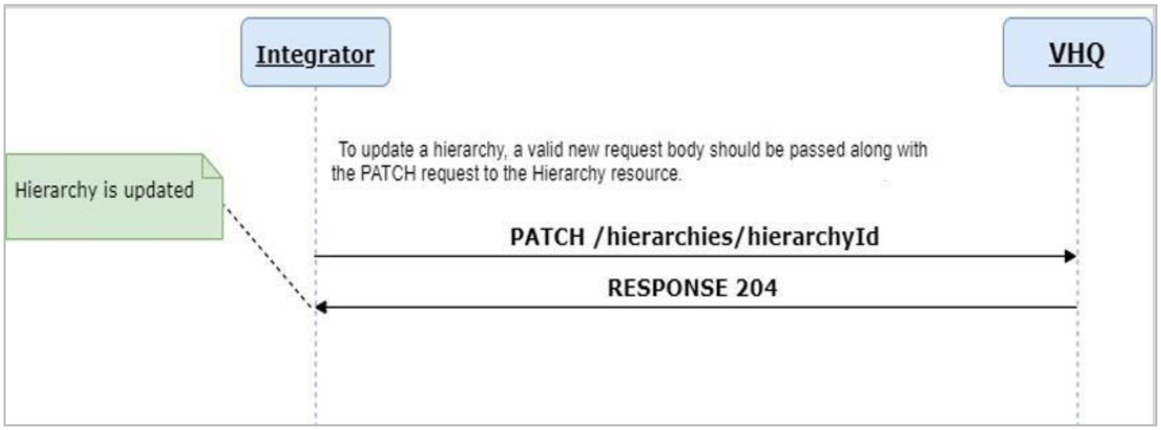
This API is used to update the details of the Hierarchy.

```
{
  "data": {
    "id": 0,
    "customerId": 0,
    "hierarchyFullPath": "string",
    "parentHierarchyId": 0,
    "parentHierarchyName": 0,
    "name": "string",
    "description": "string",
    "ipStartingAddress": "string",
    "ipEndingAddress": "string",
    "locationIdentifier": "string",
    "timezoneId": 0,
    “entityId”: “string”,
    “entityType”: “MERCHANT_SITE”,
    "childHierarchies": [
        {
         "id": 0,
         "name": "string"
        }
    ],
    "downloadAutomationEnabled": true,
    "inheritReferenceSet": true,
    "downloadOn": "NEXT_CONTACT",
    "referenceSets": [
      {
         "id": 0,
         "name": "string"
      }
    ]
  }
}
```

Go to the following path:

API – PATCH/hierarchies/{hierarchyId}
URL – https://<API Server>/apis/v1.4/hierarchies/{hierarchyId}
e.g. https://vhqtest.verifone.com/apis/v1.4/hierarchies/{hierarchyId}

Sequence Diagram


Integrator
VHQ
Hierarchy is updated
To update a hierarchy, a valid new request body should be passed along with the PATCH request to the Hierarchy resource.
PATCH /hierarchies/hierarchyId
RESPONSE 204

Input Parameters

Element	Optional/Mandatory	Data Type	Description
hierarchyId	Mandatory	Integer	Identifier of the hierarchy.
ParentHierarchyId	Optional	Integer	Parent hierarchy identifier.
ParentHierarchyName	Optional	String	Parent hierarchy name.
name	Optional	String	Name of the hierarchy.
Description	Optional	String	Description of the hierarchy.
ChildHierarchies	Optional	Array	Identifier(s) of immediate child hierarchies.
ReferenceSets	Optional	Array	Identifier(s) of the reference sets associated with the hierarchy.
DownloadOn	Optional	String	Option to indicate when to download.
IpStartingAddress	Optional	String	Starting IP Address of the hierarchy. Optional to add the entity.
IpEndingAddress	Optional	String	Ending IP Address of the hierarchy. Optional to add the entity.
DownloadAutomationEnabled	Optional	Boolean	Indicates, if automated downloads should be performed for the devices under this hierarchy.

EntityId	Optional	String	Uuid is linked with the hierarchy. Optional to add the entity. Value can be modified after the entity is created.
EntityType	Optional	String	Optional to add the entity. Value can be modified after the entity is created. [‘MERCHANT_SITE’, ‘MERCHANT_COMPANY’, ‘UNDEFINED’]
inheritReferenceSet	Optional	Boolean	Indicates, if this hierarchy inherits Reference Set from its parent hierarchy. By default, this is enabled.
LocationIdentifier	Optional	String	Location identifier of the hierarchy.

Output Parameters

None

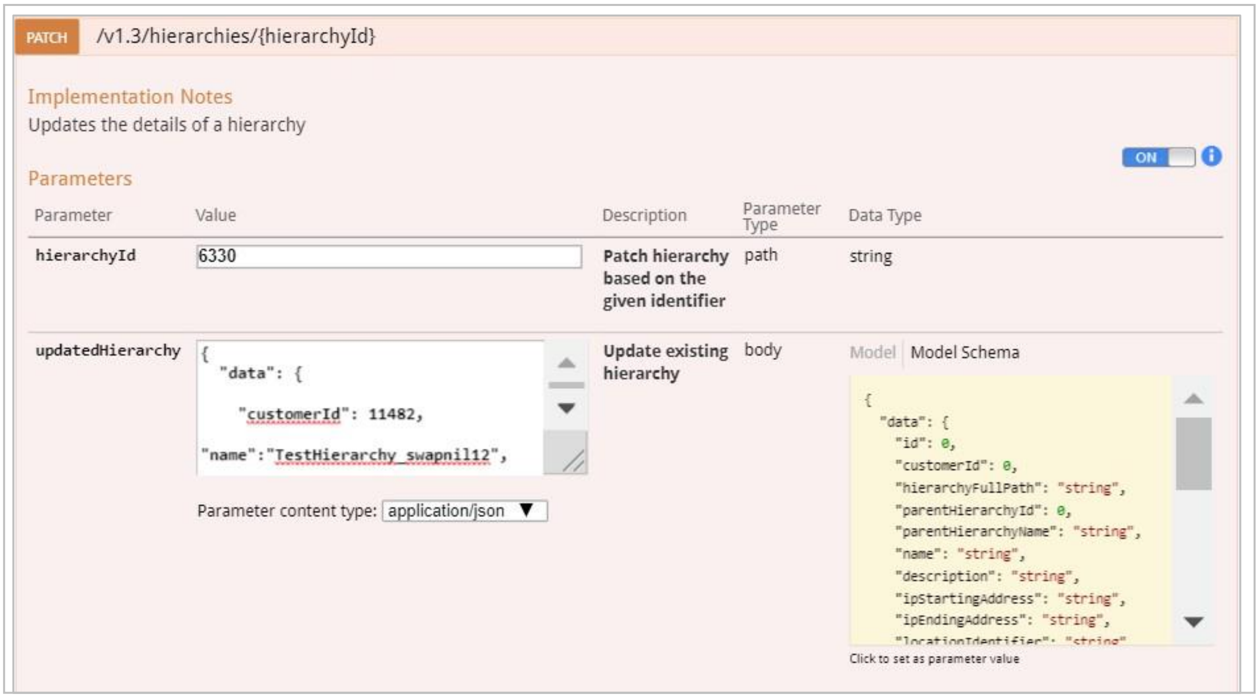
Update Hierarchy API – Sample Code

SAMPLE CODE SNIPPET

```
public class UpdateHierarchy {
   private static String url =
   “http://<servername:port>/apis/v1.3/”;private static String
   requestBody = “{\r\n” +
         “  \”data\”: {\r\n” +
         “
         \
         r
         \
         n
         ”
         +
         “
         \
         r
         \
         n
         ”
         +
         “
         \
         r
         \
         n
         ”
         +
         “    \”customerId\”: 11482,\r\n” +
         “    \”name\”:\”TestHierarchy1\”,\r\n” +
         “    \r\n” +
         “        \”referenceSets\”:[\r\n” +
         “        {\r\n” +
         “            \”id\”:106470,\r\n” +
         “          \”name\”:\”CM5-N-2.0.10\”\r\n” +
         “\r\n” +
         “        }\r\n” +
         “        ]\r\n” +
         “  }\r\n” +
         “}”; // Patch Request body in JSON format
   /**
    * @param Api -> This refers to the endpoint of the URL for the particular entity e.g. for Hierarchy
entity it ishierarchies
```

```
                 *   @return
        */
       HttpHeaders getHeader(String Api)
          { HttpHeaders headers = new
          HttpHeaders();
          headers.set(“Authorization”, “Bearer 93638536-242a-370a-9d50-
          71a4959b6390”);headers.setContentType(MediaType.APPLICATION_JSON);
          return headers;
       }
       /**
        * @param Api -> This refers to the endpoint of the URL for the particular entity e.g. for Hierarchy
    entity it ishierarchies
        * @param body -> The request body which is required to be patched, This changes as per the entity
        * @param id   -> This refer to which specific Row need to change
        */
       void patch(String api, String body, int id,String [] queryparameter,String
          queryParamValue) {String BASE_PATH = url.concat(api) + “/” + id;
          // Getting All Header Information
          // like(contentType,CustomerName,CustomerId)

          if(null!=queryparameter){
             for(String fields: queryparameter){
                BASE_PATH = BASE_PATH
                +”?”+fields+”=”+queryParamValue;
                System.out.println(“URL :::: “+BASE_PATH);
             }
          }
          HttpHeaders headers = getHeader(api);

          int statusCode;
          try {
             HttpEntity<String> entity = new HttpEntity<String>(body,
             headers);RestTemplate restTemplate = new RestTemplate();
             HttpComponentsClientHttpRequestFactory requestFactory = new
             HttpComponentsClientHttpRequestFactory();requestFactory.setConnectTimeout(18000);
             requestFactory.setReadTimeout(18000);
             restTemplate.setRequestFactory(requestFactory);

             //HttpEntity<String> result = restTemplate.exchange(BASE_PATH, HttpMethod.PATCH, entity, String.class);
             ResponseEntity<String> responseEntity = restTemplate.exchange(BASE_PATH,
    HttpMethod.PATCH, entity,String.class);
             statusCode = responseEntity.getStatusCode().value();
             System.out.println(statusCode);
          } catch
             (Exception
             ex)
             {ex.printSta
             ckTrace();
         if (ex.getMessage().equalsIgnoreCase(“404 Not Found”))
            {statusCode = 404;
            throw new RuntimeException(“Not Found “ + statusCode);
         } else if (ex.getMessage().equalsIgnoreCase(“400 Bad Request”))
            {statusCode = 400;
            throw new RuntimeException(“Bad Request “ + statusCode);
         } else if (ex.getMessage().equalsIgnoreCase(“409 Conflict”))
            {statusCode = 409;
            throw new RuntimeException(“Already Exist : Conflict “ + statusCode);
         } else {
            statusCode = 500;
            throw new RuntimeException(“Already Exist : Conflict “ + statusCode);
         }
      }
      System.out.println(“Updated Successfully”);
   }
   public static void main(String[] args) {
      String [] queryparameter=
      {“customerId”};String queryParamValue =
      “11482”;
      new UpdateHierarchy().patch(“hierarchies”, requestBody, 6332,queryparameter,queryParamValue);
   }
}
```

Sample using Swagger – Request



/v1.3/hierarchies/{hierarchyId}

Implementation Notes

Updates the details of a hierarchy

Parameters

Parameter	Value	Description	Parameter Type	Data Type
hierarchyId	6330	Patch hierarchy based on the given identifier	path	string
updatedHierarchy		Update existing hierarchy	body	Model | Model Schema

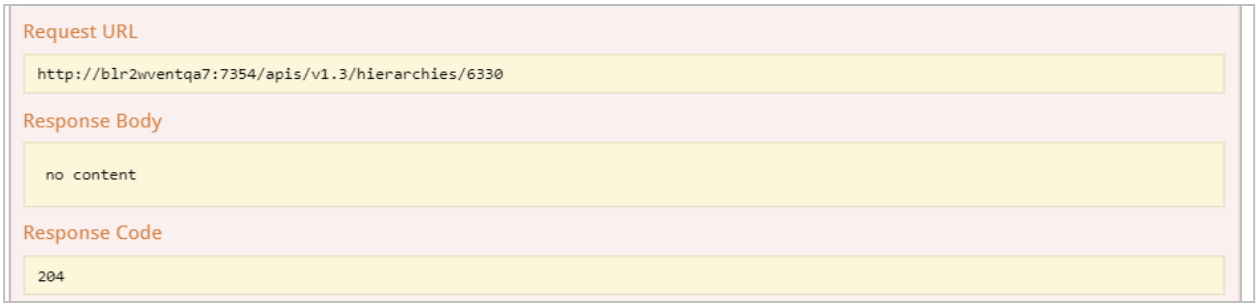
```
{
  "data": {

    "customerId": 11482,

"name":"TestHierarchy_swapnil12",
```

Parameter content type: application/json

```
{
  "data": {
    "id": 0,
    "customerId": 0,
    "hierarchyFullPath": "string",
    "parentHierarchyId": 0,
    "parentHierarchyName": "string",
    "name": "string",
    "description": "string",
    "ipStartingAddress": "string",
    "ipEndingAddress": "string",
    "locationIdentifier": "string"
```

Click to set as parameter value

Sample using Swagger – Response



Request URL

```
http://blr2wventqa7:7354/apis/v1.3/hierarchies/6330
```

Response Body

```
no content
```

Response Code

```
204
```

Error Codes

HTTP Status Code	Application Code	Name	Severity	Message
201	S201	S_CREATED	-	Success.
401	E101	E_TOKEN_INVALID_OR_EXPIRED	2	Token expired or invalid.
401	E102	E_UNAUTHORIZED_ACCESS	2	The user does not have the right to perform the operation.
500	E103	E_GENERAL_ERROR	3	API failed.

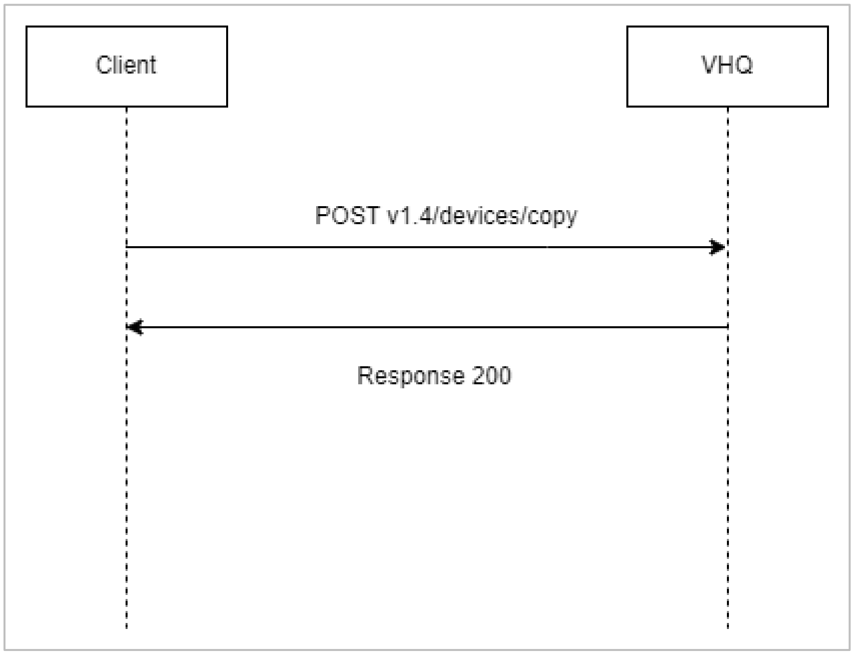
COPY DEVICE – 1.4

This API is used for cloning a device’s detail as a template while creating a new device entry with the same configurations as the source device.
This API creates a template for device ID, target serial number, and target device ID along with copy options as shown in the sample request.
A new device is added with the status as PENDING REGISTRATION, cloning the configurations of the source device.
By calling the above API, a new device is added with the below details:

SerialNumber	= targetSerialNumber
DeviceId	= targetDeviceId
Model	= sourceModelId (or sourceModelName)
Status	= PENDING REGISTRATION

API - Copy Device
URL : POST v1.4/devices/copy

Sequence Diagram

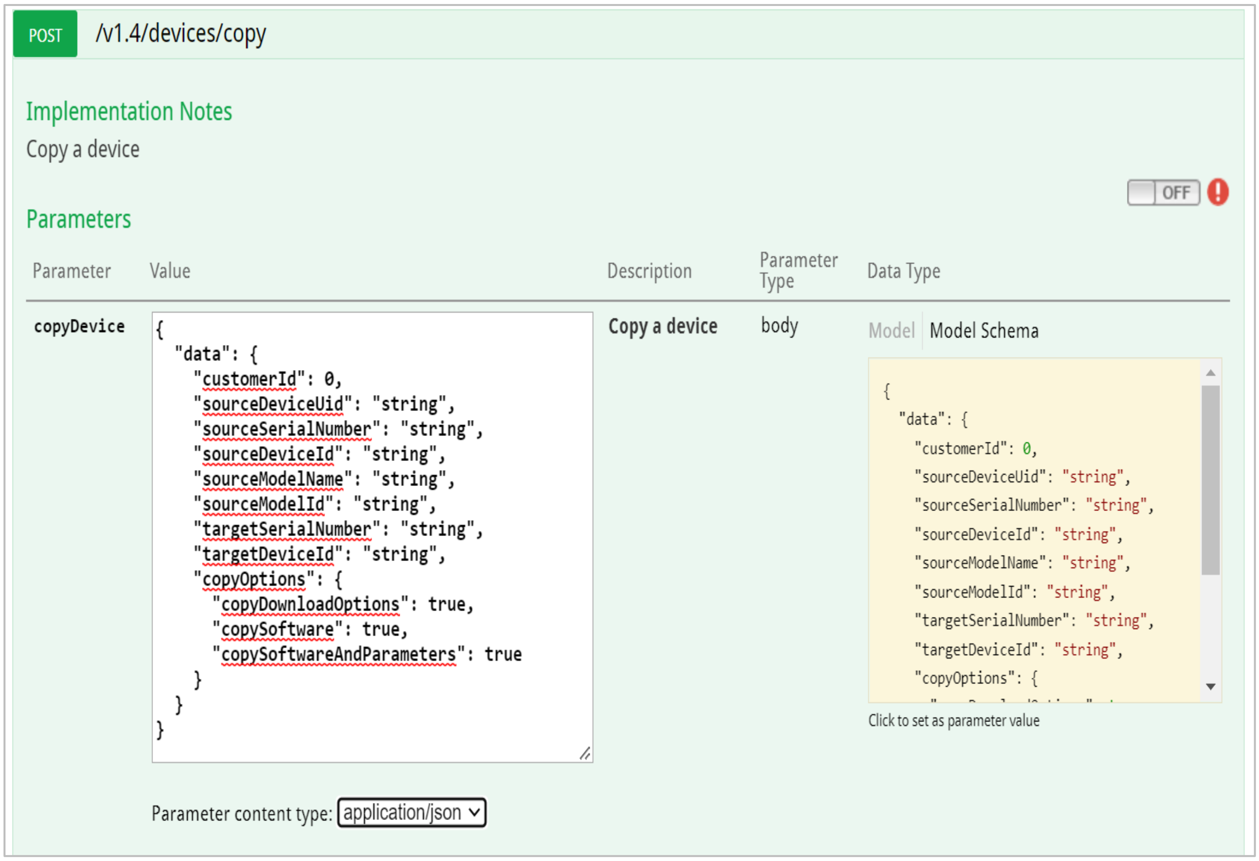


Copy Device API – Sample Code

SAMPLE CODE SNIPPET

```
package com.verifone ;

import java.awt.PageAttributes.MediaType ;
import java.net.http.HttpHeaders ;

public class CopyDevices {
    private static String url = « http ://<servername:port>/apis/v1.4/devices/copy » ;
    //This is the base URL, 35iérarc be appended as per the entity which is invoked
  //along with the corresponding deviceUid
    private static String postparameter = « {\r\n » +
        « \ »data\ » :\r\n » +
        « {\r\n » +
        « \ »customerId\ » :1382,\r\n » +

   "\"sourceDeviceUid\":\"4044\",\r\n" +

  "\"targetSerialNumber\":\"030-203-100\",\r\n" +
  "\"targetDeviceId\":\"New-01\",\r\n" +
  "\"copyOptions\":\r\n" +
  "{\r\n" + "\"copyDownloadOptions\":true,\r\n"
  + "\"copySoftware\":true,\r\n" +
  "\"copySoftwareAndParameters\":true \r\n"+
  "}\r\n" +
  "}\r\n" +
  "}\r\n" +
  " ";

/**

 * @param Api -> This refers to the endpoint of the URL for the particular entity e.g. for Device entity
it is devices
 * @return
 */

HttpHeaders getHeader(String Api)
  {
    HttpHeaders headers = new HttpHeaders();
    headers.set("Authorization", "Bearer e22abce8-c938-3b47-87ac-8edafca2f12d");
headers.setContentType(MediaType.APPLICATION_JSON);
    return headers;
  }

  /**
   * @param Api -> This refers to the endpoint of the URL for the particular entity e.g. for Device
entity it is devices
   * @param body -> The request body which is required to be patched, This changes as per the entity
   * @param id -> This refer to which specific Row need to change
   */

  void post(String api, String body, int id)
  {
    int statusCode;
    String BASE_PATH = api;
// Getting All Header Information
// like(contentType,CustomerName,CustomerId)
    HttpHeaders headers = getHeader(api);
    try {
      HttpEntity<String> entity = new HttpEntity<String>(body, headers);

RestTemplate restTemplate = new RestTemplate();

HttpComponentsClientHttpRequestFactory requestFactory = new
          HttpComponentsClientHttpRequestFactory(); requestFactory.setConnectTimeout(18000);
requestFactory.setReadTimeout(18000);

restTemplate.setRequestFactory(requestFactory);
//HttpEntity<String> result = restTemplate.exchange(BASE_PATH, HttpMethod.PATCH, entity, String.class);
      ResponseEntity<String> responseEntity = restTemplate.exchange(BASE_PATH, HttpMethod.PATCH,
entity, String.class);
      statusCode = responseEntity.getStatusCode().value();

System.out.println(statusCode);
    } catch (Exception ex)
 { ex.printStackTrace();
      if (ex.getMessage().equalsIgnoreCase(“404 Not Found”)) {
          statusCode = 404; throw new RuntimeException(“Not Found “ + statusCode);
      }
      else if (ex.getMessage().equalsIgnoreCase(“400 Bad Request”)) {
          statusCode = 400; throw new RuntimeException(“Bad Request “ + statusCode);
      }
      else if (ex.getMessage().equalsIgnoreCase(“409 Conflict”)) {
          statusCode = 409; throw new RuntimeException(“Already Exist : Conflict “ + statusCode);
      }
      else {
          statusCode = 500; throw new RuntimeException(“Already Exist : Conflict “ + statusCode);
      }
    }
  }

  public static void main(String[] args) {
    new CopyDevices().post(“devices”, postparameter, 78500);

System.out.println(“Device Cloned Successfully”);
  }
}
```

Sample using Swagger – Request



Implementation Notes

Copy a device

Parameters

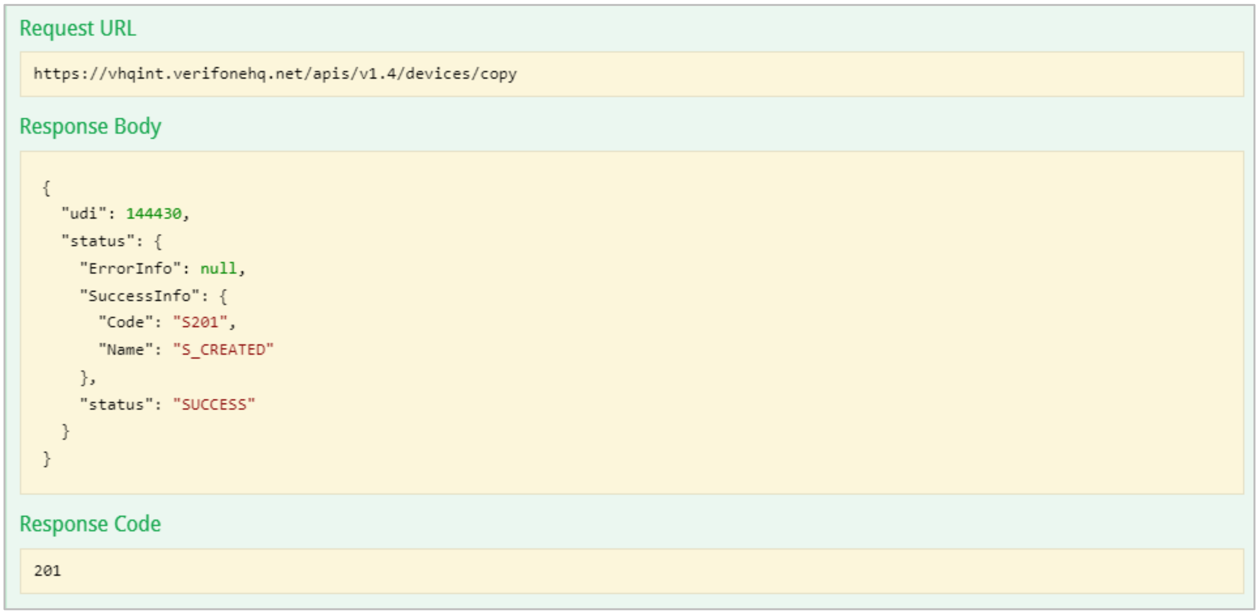
Parameter	Value	Description	Parameter Type	Data Type
copyDevice	(see below)	Copy a device	body	Model Model Schema

```
{
  "data": {
    "customerId": 0,
    "sourceDeviceUid": "string",
    "sourceSerialNumber": "string",
    "sourceDeviceId": "string",
    "sourceModelName": "string",
    "sourceModelId": "string",
    "targetSerialNumber": "string",
    "targetDeviceId": "string",
    "copyOptions": {
      "copyDownloadOptions": true,
      "copySoftware": true,
      "copySoftwareAndParameters": true
    }
  }
}
```

```
{
  "data": {
    "customerId": 0,
    "sourceDeviceUid": "string",
    "sourceSerialNumber": "string",
    "sourceDeviceId": "string",
    "sourceModelName": "string",
    "sourceModelId": "string",
    "targetSerialNumber": "string",
    "targetDeviceId": "string",
    "copyOptions": {
```

Parameter content type: application/json

Sample using Swagger – Response



Request URL

```
https://vhqint.verifonehq.net/apis/v1.4/devices/copy
```

Response Body

```
{
  "udi": 144430,
  "status": {
    "ErrorInfo": null,
    "SuccessInfo": {
      "Code": "S201",
      "Name": "S_CREATED"
    },
    "status": "SUCCESS"
  }
}
```

Response Code

```
201
```

Error Codes

HTTP Status code	Application code	Name	Severity	Message
201	S200	S_SUCCESS	-	Success.
401	E101	E_TOKEN_INVALID_OR_EXPIRED	2	Token is expired or invalid.
401	E102	E_UNAUTHORIZED_ACCESS	2	The user does not have the right to operate.
500	E103	E_GENERAL_ERROR	3	API failed.
400	5006	Argument Is Invalid	3	Invalid argument.

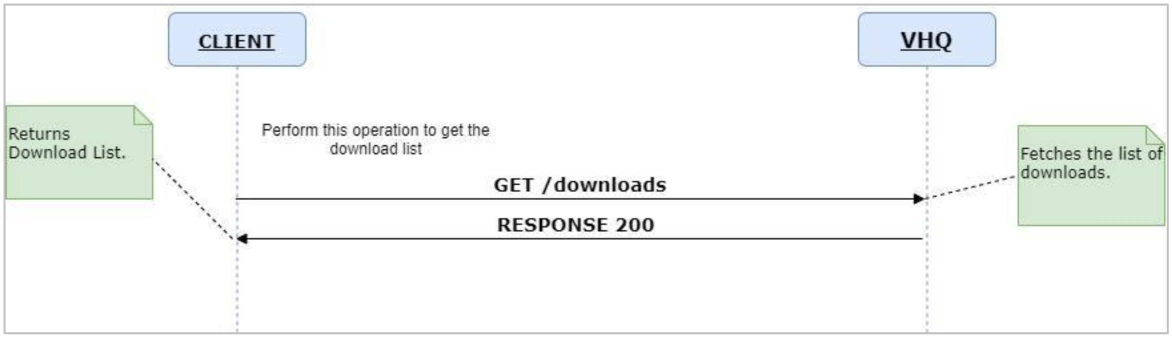
GET DOWNLOADS – 1.4

This API is used to get the status of all the downloads. Scheduling the download of multiple packages onto multiple devices results in multiple download tasks. This API returns the download information for each download task that indicates the status of the download by device and package.
This API is enhanced to capture the latest modification date of the download task or job.

Go to the following path:

```
API - GET/downloads
URL- https://<APIServer>/apis/v1.4/downloads
e.g. https://vhqtest.verifone.com/apis/v1.4/downloads
```

Sequence Diagram


CLIENT
VHQ
Returns Download List.
Perform this operation to get the download list
GET /downloads
RESPONSE 200
Fetches the list of downloads.

Input Parameters

Element	Optional/Mandatory	Data Type	Description
packageId/packageName	Optional	Integer/String	Package Identifier/package name.

Output Parameters

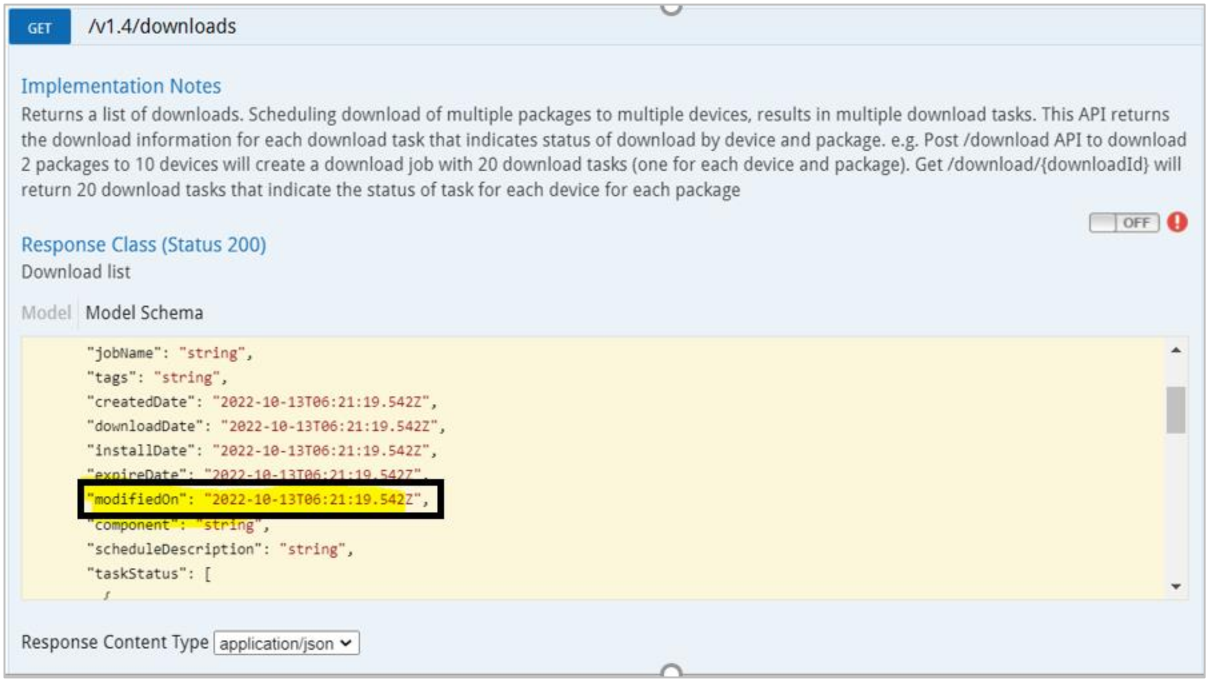
Element	Attribute	Data Type	Description
downloads (collection of downloads)	id	Integer	Download Identifier.
	taskStatus	Array	Details of the task.

Get Downloads API – Sample Code

SAMPLE CODE SNIPPET

```
public class GetDownloads {
    private static String url = http://<servername:port>/apis/v1.4/;
    private int statusCode; // This is the base URL, it will be appended as per the entity which is invoked /**
    * @param completeURL -> This is the URL which appended as per the entity which is invoked
    * @param method -> This refer to which method we are going to send e.g. get,post,put
    * @return -> httpCon with Content Type and Authorization token
    * @throws IOException
    */
HttpURLConnection getConnection(URL completeURL, String method) throws IOException {
HttpURLConnection httpCon = (HttpURLConnection) completeURL.openConnection();
httpCon.setDoOutput(true);
httpCon.setRequestProperty("Authorization", "Bearer 8c10910f-9228-37f7-a1d3-d73883ea2615");
httpCon.setRequestMethod(method); httpCon.setRequestProperty("Content-Type", "application/json"); return httpCon;
} /**
* This method get all data of a entity
* @param Api -> This refers to the endpoint of the URL for the particular entity e.g. for Parameter entity it is
parameters
* @param id -> This refer to show specific data from Entity based on Id.
* @param fields -> This refers to show specific fields, which we pass as a parameter.
* @param queryparameter * @return response from entity
*/
String getDownloads(String Api, int id,String[] queryparameter,String queryParamValue) {
    StringBuffer response=null;
    if(id!=0) {
        url = url.concat(Api) + "/" + id;
    }else {
        url = url.concat(Api);
    }
    String fullPath = null;
    if(fullPath!=null){
        url=url+"/".concat(fullPath);
    }
    if(null!=queryparameter){
        for(String fields: queryparameter){ url = url +"?"+fields+"="+queryParamValue; System.out.println("URL ::::
"+url);
        }
    }
    try { URL completeURL = new URL(url);
// Getting All Header Information
// like(contentType,CustomerName,CustomerId)
        HttpURLConnection httpCon = getConnection(completeURL, "GET");
        httpCon.getResponseCode(); BufferedReader in = new BufferedReader(new
InputStreamReader(httpCon.getInputStream()));
        String inputLine; response =new StringBuffer();
        while ((inputLine = in.readLine()) != null) { response.append(inputLine); }
        System.out.println(response);
        System.out.println(httpCon.getResponseCode());
        System.out.println(httpCon.getResponseMessage());
    } catch (Exception ex) {
        ex.printStackTrace();
        if (ex.getMessage().equalsIgnoreCase("404 Not Found")) {
            statusCode = 404;
            throw new RuntimeException("Not Found " + statusCode); }
        else if (ex.getMessage().equalsIgnoreCase("400 Bad Request")) {
            statusCode = 400; throw new RuntimeException("Bad Request " + statusCode); }
        else if (ex.getMessage().equalsIgnoreCase("409 Conflict")) { statusCode = 409; throw new
RuntimeException("Already Exist : Conflict " + statusCode); }
        else { statusCode = 500; throw new RuntimeException("Already Exist : Conflict " + statusCode); } }
return response.toString(); }
public static void main(String[] args) {
    String [] queryparameter= {"customerId"}; String queryParamValue = "11482";
new GetDownloads().getDownloads("downloads",0,queryparameter,queryParamValue);
}
}
```

Sample using Swagger – Response



Implementation Notes

Returns a list of downloads. Scheduling download of multiple packages to multiple devices, results in multiple download tasks. This API returns the download information for each download task that indicates status of download by device and package. e.g. Post /download API to download 2 packages to 10 devices will create a download job with 20 download tasks (one for each device and package). Get /download/{downloadId} will return 20 download tasks that indicate the status of task for each device for each package

Response Class (Status 200)

Download list

```
"jobName": "string",
"tags": "string",
"createdDate": "2022-10-13T06:21:19.542Z",
"downloadDate": "2022-10-13T06:21:19.542Z",
"installDate": "2022-10-13T06:21:19.542Z",
"expireDate": "2022-10-13T06:21:19.542Z",
"modifiedOn": "2022-10-13T06:21:19.542Z",
"component": "string",
"scheduleDescription": "string",
"taskStatus": [
  {
```

Response Content Type application/json

Error Codes

HTTP Status Code	Application Code	Name	Severity	Message

200	S100	S_OK	-	Success.
401	E101	E_TOKEN_INVALID_OR_EXPIRED	2	Token expired or it is invalid.
401	E102	E_UNAUTHORIZED_ACCESS	2	The user does not have the right to perform the operation.
500	E103	E_GENERAL_ERROR	3	API failed.

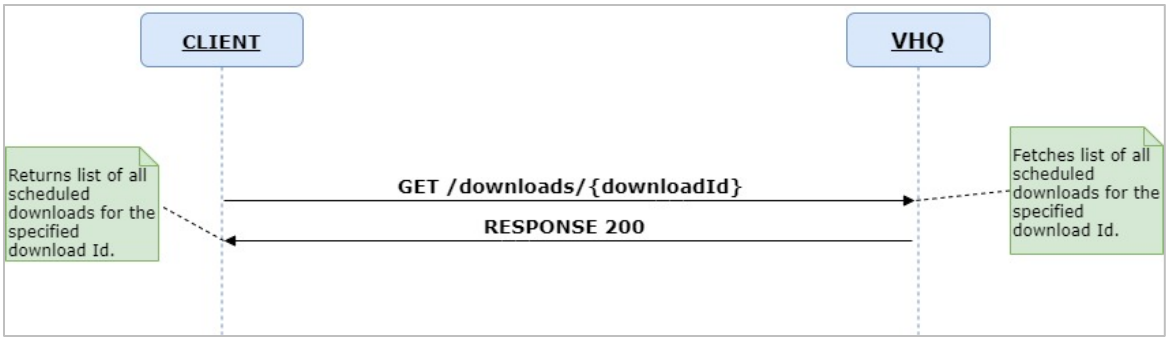
GET DOWNLOADS BY DOWNLOADID – 1.4

This API is used to get the details of a particular downloadId along with the latest modification date of the download task or job.

Once you have the download ID, go to the following path:

```
API - GET/downloads/{downloadId}
URL - https://<API Server>/apis/v1.4/downloads/{downloadId}
e.g. https://vhqtest.verifone.com/apis/v1.4/downloads/{downloadId}
```

Sequence Diagram


CLIENT
VHQ
Returns list of all scheduled downloads for the specified download Id.
GET /downloads/{downloadId}
RESPONSE 200
Fetches list of all scheduled downloads for the specified download Id.

Input Parameters

Element	Optional/Mandatory	Data Type	Description
downloadId	Mandatory	Int	Unique device identifier to be set as the path of the URI.

Output Parameters

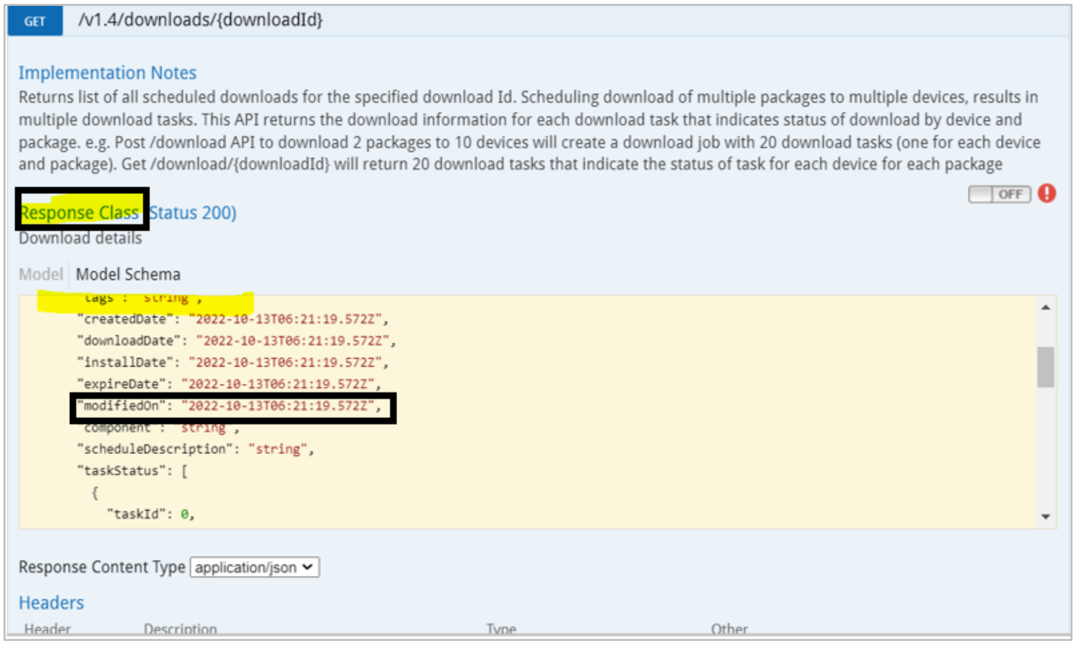
Element	Attribute	Data Type	Description
downloads (collection of downloads)	id	int	Download Identifier.
	taskStatus	array	Details of the task.

Get Downloads by DownloadId API - Sample Code

SAMPLE CODE SNIPPET

```
public class GetDownloadsById {
   private static String url = "http://<servername:port>/apis/v1.3/";
   private int statusCode;
   // This is the base URL, it will be appended as per the entity which is invoked
   /**
    * @param completeURL -> This is the URL which appended as per the
   entity which is invoked
    * @param method -> This refer to which method we are going to
   send e.g. get,post,put
    * @return -> httpCon with Content Type and Authorization token
    * @throws IOException
    */
   HttpURLConnection getConnection(URL completeURL, String method)
         throws IOException {
      HttpURLConnection httpCon = (HttpURLConnection)
            completeURL.openConnection();
      httpCon.setDoOutput(true);
      httpCon.setRequestProperty("Authorization", "Bearer dec38556-c51a-3702-82e9-b97757930023");
      httpCon.setRequestMethod(method);
      httpCon.setRequestProperty("Content-Type", "application/json");
      return httpCon;
   }
   /**
    * This method get all data of an entity
    * @param Api -> This refers to the endpoint of the URL for the
   particular entity e.g. for Download entity it is downloads
    * @param id -> This refer to show specific data from Entity based
   on Id.
    * @param fields -> This refers to show specific fields, which we
   pass as a parameter.
    * @param queryparameter
    * @return response from entity
    */
   String getById(String Api, int id,String fullPath,String[] queryparameter,String queryParamValue) {
      StringBuffer response=null;
      if(id!=0) {
         url = url.concat(Api) + "/" + id;
      }else {
         url = url.concat(Api);
      }
      if(fullPath!=null){
         url=url+"/".concat(fullPath);
      }
      if(null!=queryparameter){
         for(String fields: queryparameter){
            url = url +"?"+fields+"="+queryParamValue;
            System.out.println("URL :::: "+url);
         }
      }
      try {
         URL completeURL = new URL(url);
         // Getting All Header Information
         // like(contentType,CustomerName,CustomerId)
         HttpURLConnection httpCon = getConnection(completeURL, "GET");
         httpCon.getResponseCode();
         BufferedReader in = new BufferedReader(new InputStreamReader(httpCon.getInputStream()));
         String inputLine;
         response =new StringBuffer();
         while ((inputLine = in.readLine()) != null) {
            response.append(inputLine);
         }
         System.out.println(response);
         System.out.println(httpCon.getResponseCode());
         System.out.println(httpCon.getResponseMessage());
      } catch (Exception ex) {
         ex.printStackTrace();
         if (ex.getMessage().equalsIgnoreCase("404 Not Found")) {
            statusCode = 404;
            throw new RuntimeException("Not Found " + statusCode);

         } else if (ex.getMessage().equalsIgnoreCase("400 Bad Request")) {
            statusCode = 400;
            throw new RuntimeException("Bad Request " + statusCode);
         } else if (ex.getMessage().equalsIgnoreCase("409 Conflict"))
         {
            statusCode = 409;
            throw new RuntimeException("Already Exist : Conflict " +
                    statusCode);
         } else {
            statusCode = 500;
            throw new RuntimeException("Already Exist : Conflict " +
                    statusCode);
         }
      }
      return response.toString();
   }
   public static void main(String[] args) {
      String [] queryparameter= {"customerId"};
      String queryParamValue = "11482";
      new GetDownloadsById().getById("downloads",983363,null,queryparameter,queryParamValue);
   }
}
```

Sample using Swagger – Response



Implementation Notes

Returns list of all scheduled downloads for the specified download Id. Scheduling download of multiple packages to multiple devices, results in multiple download tasks. This API returns the download information for each download task that indicates status of download by device and package. e.g. Post /download API to download 2 packages to 10 devices will create a download job with 20 download tasks (one for each device and package). Get /download/{downloadId} will return 20 download tasks that indicate the status of task for each device for each package

Response Class (Status 200)

Download details

```
      "tags": "string",
      "createdDate": "2022-10-13T06:21:19.572Z",
      "downloadDate": "2022-10-13T06:21:19.572Z",
      "installDate": "2022-10-13T06:21:19.572Z",
      "expireDate": "2022-10-13T06:21:19.572Z",
      "modifiedOn": "2022-10-13T06:21:19.572Z",
      "component": "string",
      "scheduleDescription": "string",
      "taskStatus": [
        {
          "taskId": 0,
```

Response Content Type application/json

Error Codes

HTTP Status Code	Application Code	Name	Severity	Message
200	S100	S_OK	-	Success.
401	E101	E_TOKEN_INVALID_OR_EXPIRED	2	The token is expired, or it is invalid.
401	E102	E_UNAUTHORIZED_ACCESS	2	The user does not have the right to perform the operation.
500	E103	E_GENERAL_ERROR	3	API failed.

Create Hierarchy – 1.3

This API is used to add a new hierarchy.

Pre-requisites

Following are the pre-requisites for creating a new hierarchy:

Mandatory Fields:

1. Parent Hierarchy
2. Name of the Hierarchy (User Defined)

Non-Mandatory Fields: Reference Set

NOTE: The user can use Reference set name instead of Reference set ID.

Request Body

```
{
  “data”: {
    “id”: 0,
    “customerId”: 0,
    “hierarchyFullPath”: “string”,
    “parentHierarchyId”: 0,
    “parentHierarchyName”: 0,
    “name”: “string”,
    “description”: “string”,
    “ipStartingAddress”: “string”,
    “ipEndingAddress”: “string”,
    “locationIdentifier”: “string”,
    “timezoneId”: 0,
    “entityId”: “string”,
    “childHierarchies”: [
        {
         “id”: 0,
         “name”: “string”
        }
    ],

    “downloadAutomationEnabled”: true,
    “inheritReferenceSet”: true,
    “downloadOn”: “NEXT_CONTACT”,
    “referenceSets”: [    <← Attribute added
      {
         “id”: 0,            <← id or name mandatory
         “name”: “string”
      }
    ]
  }
}
```

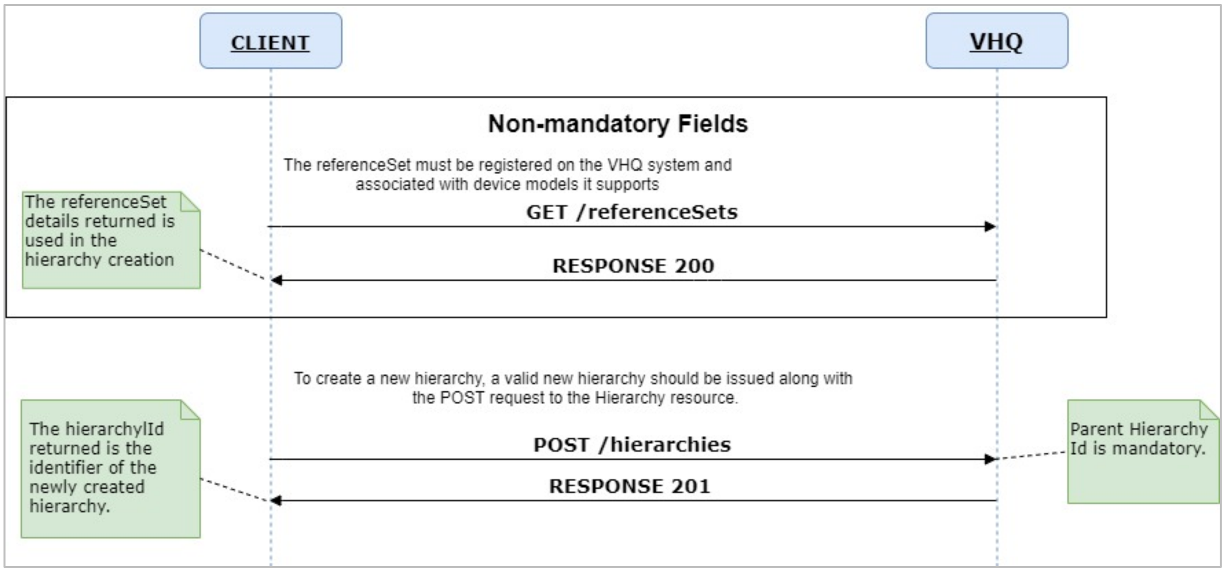
Go to the following path:

`API – POST/hierarchies`

`URL – https://<API Server>/apis/v1.3/hierarchies`

`e.g. https://vhqtest.verifone.com/apis/v1.3/hierarchies`

Sequence Diagram


CLIENT
VHQ
Non-mandatory Fields
The referenceSet must be registered on the VHQ system and associated with device models it supports
GET /referenceSets
RESPONSE 200
The referenceSet details returned is used in the hierarchy creation
To create a new hierarchy, a valid new hierarchy should be issued along with the POST request to the Hierarchy resource.
POST /hierarchies
RESPONSE 201
The hierarchyId returned is the identifier of the newly created hierarchy.
Parent Hierarchy Id is mandatory.

Input Parameters

Element	Optional/Mandatory	Data Type	Description
hierarchyFullPath	Optional	String	The full path of the hierarchy.
ParentHierarchyId	Mandatory	Integer	Parent hierarchy identifier.
ParentHierarchyName	Optional	String	Parent hierarchy name.
name	Mandatory	String	Name of the hierarchy.
Description	Optional	String	Description of the hierarchy.
ChildHierarchies	Optional	Array	Identifier(s) of immediate child hierarchies.
ReferenceSets	Optional	Array	Identifier(s) of the reference sets associated with the hierarchy.
DownloadOn	Optional	String	Option to indicate when to download.

IpStartingAddress	Optional	String	Starting IP Address of the hierarchy. Optional to add the entity.
IpEndingAddress	Optional	String	Ending IP Address of the hierarchy. Optional to add the entity.
DownloadAutomationEnabled	Optional	Boolean	Indicates, if automated downloads should be performed for the devices under this hierarchy.
EntityId	Optional	String	Uuid is linked with the hierarchy. Optional to add the entity. Value can be modified after the entity is created.
InheritReferenceSet	Optional	Boolean	Indicates, if this hierarchy inherits Reference Set from its parent hierarchy. By default, this is enabled.
LocationIdentifier	Optional	String	Location identifier of the hierarchy.

Output Parameters

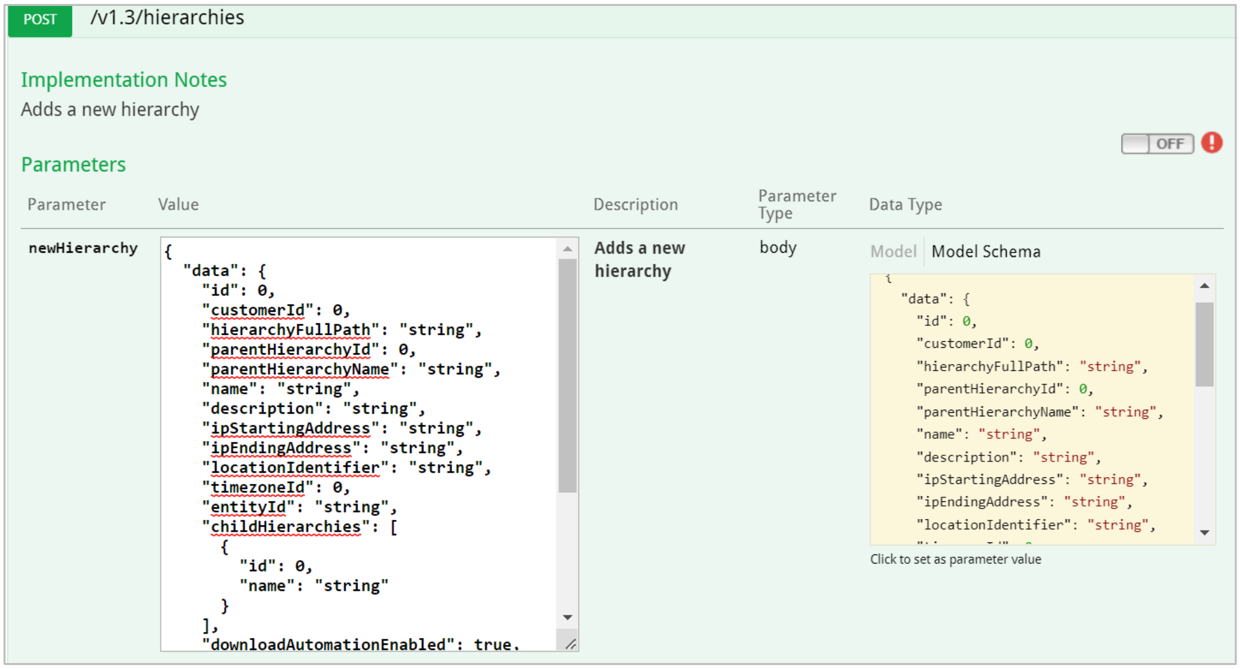
Element	Data Type	Description
hierarchyId	int	hierarchy Id of the newly created hierarchy.

Create Hierarchy API – Sample Code

SAMPLE CODE SNIPPET

```
package com.VhqApiSampleCodeTest ;
public class CreateHierarchy {
   //Common API URL
   private static String url = « http ://<servername:port>/apis/v1.3/ » ;
   private int statusCode ;
   // Request body
   private static String postHierarchy= « {\r\n » +
         «   \ »data\ » : {\r\n » +
         «      \ »customerId\ » : 11482,\r\n » +
         «      \ »name\ » : \ »TestHierarchy_123\ »,\r\n » +
         «      \ »hierarchyFullPath\ » : \ »RATNERCO >> RATNERCO\ »,\r\n » +
         «      \ »parentHierarchyName\ » : \ »AutomationRootHierarchy\ »,\r\n » +
         «      \ »downloadAutomationEnabled\ » : false,\r\n » +
         «      \ »inheritReferenceSet\ » : true,\r\n » +
         «      \ »downloadOn\ » : \ »NEXT_CONTACT\ »,\r\n » +
         «      \ »referenceSets\ » : [{\r\n » +
         «            \ »id\ » : 107082,\r\n » +
         «            \ »name\ » : \ »\ »\r\n » +
         «         },\r\n » +
         «         {\r\n » +
         «            \ »id\ » : 106463\r\n » +
         «         }\r\n » +
         «      ]\r\n » +
         «   }\r\n » +
         « } » ;
   /**
    * @param completeURL -> This is the URL which appended as per the entity which is invoked
    * @param method   -> This refer to which method we are going to send e.g. get,post,put
    * @return   -> httpCon with Content Type and Authorization token
    * @throws IOException
    */
   HttpURLConnection getConnection(URL completeURL, String method) throws IOException {
      HttpURLConnection httpCon = (HttpURLConnection) completeURL.openConnection() ;
      httpCon.setDoOutput(true) ;
      httpCon.setRequestProperty(« Authorization », « Bearer 17fee712-031d-3dcd-abd6-236a28393210 ») ;
      httpCon.setRequestMethod(method) ;
      httpCon.setRequestProperty(« Content-Type », « application/json ») ;
      return httpCon ;
   }
   /**
    * This method creates a new entity
    * @param body -> The request body which is required to be posted, This changes as per the entity
    * @param Api -> This refers to the endpoint of the URL for the particular entity e.g. for Hierarchy entity it is 
46iérarchies
    */
   String post(String body, String apiEndPoint,String [] queryparameter, String queryParamValue) {
      url = url.concat(apiEndPoint) ;
      String id = «  » ;
      if(null !=queryparameter){
         for(String fields : queryparameter){
            url = url + » ? »+fields+ »= »+queryParamValue ;
            System.out.println(« URL :::: « +url) ;
         }
      }
      try {
         URL completeURL = new URL(url) ;
         // Getting All Header Information
         // like(contentType,CustomerName,CustomerId)
         HttpURLConnection httpCon = getConnection(completeURL, « POST ») ;
         OutputStream os = httpCon.getOutputStream() ;
         os.write(body.getBytes()) ;
         os.flush() ;


         if (httpCon.getResponseCode() != HttpURLConnection.HTTP_CREATED) {
            if (httpCon.getResponseCode() == 409) {
               throw new RuntimeException(apiEndPoint + « Already Exist : Conflict «  + httpCon.getResponseCode()) ;
            }
         }
         for(Entry<String, List<String>> entry : httpCon.getHeaderFields().entrySet()){
            if(null !=entry.getKey() &&entry.getKey().toLowerCase().contains(« new »)){
               id = entry.getValue().get(0) ;
               id = id.substring(id.lastIndexOf(« / »)+1) ;
            }
         }
         System.out.println(id) ;
         System.out.println(httpCon.getHeaderFields()) ;
         System.out.println(« Response Code for POST method-> »+httpCon.getResponseCode()) ;
         System.out.println(httpCon.getResponseMessage()) ;
      } catch (Exception ex) {
         ex.printStackTrace() ;
         if (ex.getMessage().equalsIgnoreCase(« 404 Not Found »)) {
            statusCode = 404 ;
            throw new RuntimeException(« Not Found «  + statusCode) ;
         } else if (ex.getMessage().equalsIgnoreCase(« 400 Bad Request »)) {
            statusCode = 400 ;
            throw new RuntimeException(« Bad Request «  + statusCode) ;
         } else if (ex.getMessage().equalsIgnoreCase(« 409 Conflict »)) {
            statusCode = 409 ;
            throw new RuntimeException(« Already Exist : Conflict «  + statusCode) ;
         } else {
            statusCode = 500 ;
            throw new RuntimeException(« Already Exist : Conflict «  + statusCode) ;
         }
      }
      System.out.println(« Added Successfully ») ;
      return id ;
   }
   public static void main(String[] args) {
      System.out.println(«  :::::::: Create Hierarchy calling start :::::::: ») ;
      String [] queryparameter= {« customerId »} ;
      String queryParamValue = « 11482 » ;
      //Post method will accept two parameter, Request Body and API (46iérarchies)
      CreateHierarchy createHierarchies = new CreateHierarchy() ;
      createHierarchies.post(postHierarchy, « 46iérarchies »,queryparameter,queryParamValue) ;
   }
}
```

Sample using Swagger – Request



Implementation Notes

Adds a new hierarchy

Parameters

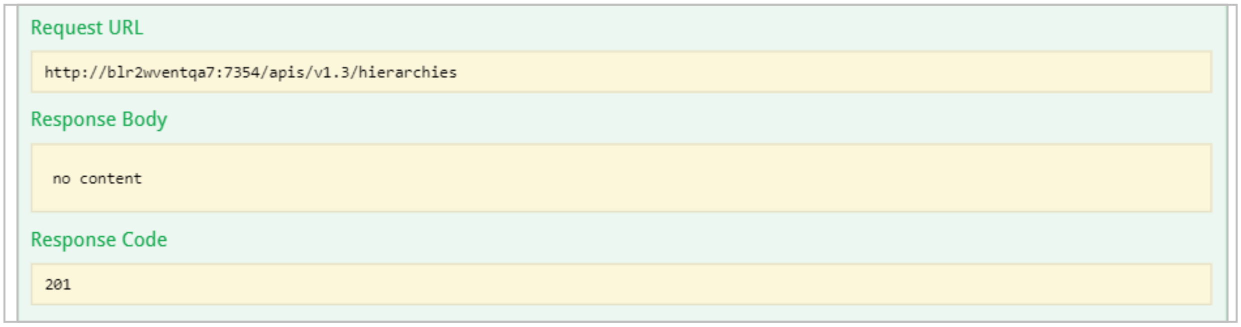
Parameter	Value	Description	Parameter Type	Data Type
newHierarchy		Adds a new hierarchy	body	Model Model Schema

```
{
  "data": {
    "id": 0,
    "customerId": 0,
    "hierarchyFullPath": "string",
    "parentHierarchyId": 0,
    "parentHierarchyName": "string",
    "name": "string",
    "description": "string",
    "ipStartingAddress": "string",
    "ipEndingAddress": "string",
    "locationIdentifier": "string",
    "timezoneId": 0,
    "entityId": "string",
    "childHierarchies": [
      {
        "id": 0,
        "name": "string"
      }
    ],
    "downloadAutomationEnabled": true,
```

```
{
  "data": {
    "id": 0,
    "customerId": 0,
    "hierarchyFullPath": "string",
    "parentHierarchyId": 0,
    "parentHierarchyName": "string",
    "name": "string",
    "description": "string",
    "ipStartingAddress": "string",
    "ipEndingAddress": "string",
    "locationIdentifier": "string",
```

Click to set as parameter value

Sample using Swagger – Response



Request URL

```
http://blr2wventqa7:7354/apis/v1.3/hierarchies
```

Response Body

```
no content
```

Response Code

```
201
```

Error Codes

HTTP Status Code	Application Code	Name	Severity	Message
201	S201	S_CREATED	-	Success.
401	E101	E_TOKEN_INVALID_OR_EXPIRED	2	Token expired or invalid.
401	E102	E_UNAUTHORIZED_ACCESS	2	The user does not have the right to perform the operation.
409	E116	E_HIERARCHY_ALREADY_EXISTS	2	Hierarchy already exists
500	E103	E_GENERAL_ERROR	3	API failed.

UPDATE HIERARCHY – 1.3

This API is used to update the details of the Hierarchy.

```
{
  "data": {
    "id": 0,
    "customerId": 0,
    "hierarchyFullPath": "string",
    "parentHierarchyId": 0,
    "parentHierarchyName": 0,
    "name": "string",
    "description": "string",
    "ipStartingAddress": "string",
    "ipEndingAddress": "string",
    "locationIdentifier": "string",
    "timezoneId": 0,
    "entityId": "string",
    "childHierarchies": [
        {
         "id": 0,
         "name": "string"
        }
    ],

    "downloadAutomationEnabled": true,
    "inheritReferenceSet": true,
    "downloadOn": "NEXT_CONTACT",
    "referenceSets": [
      {
         "id": 0,
         "name": "string"
      }
    ]
  }
}
```

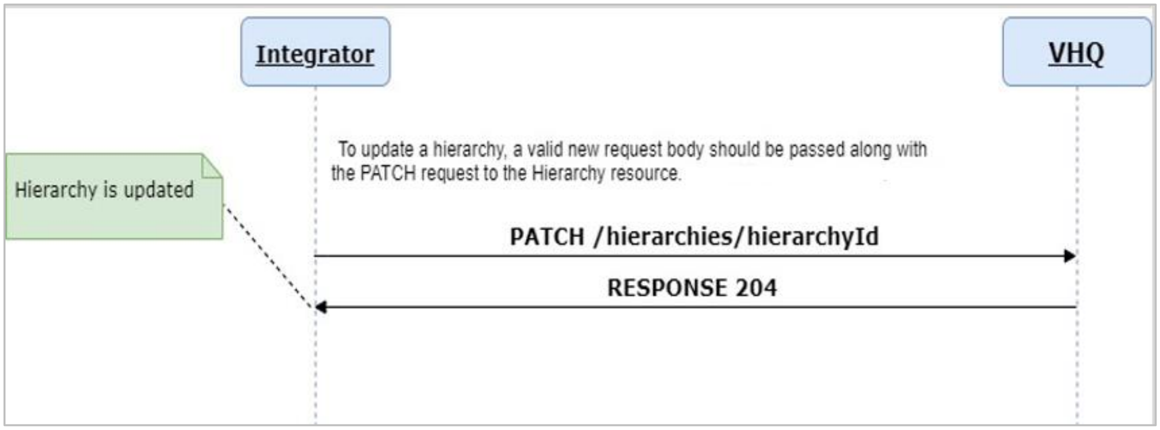
Go to the following path:

API - PATCH/hierarchies/{hierarchyId}

URL - https://<API Server>/apis/v1.3/hierarchies/{hierarchyId}

e.g. https://vhqtest.verifone.com/apis/v1.3/hierarchies/{hierarchyId}

Sequence Diagram


Integrator
VHQ
Hierarchy is updated
To update a hierarchy, a valid new request body should be passed along with the PATCH request to the Hierarchy resource.
PATCH /hierarchies/hierarchyId
RESPONSE 204

Input Parameters

Element	Optional/Mandatory	Data Type	Description
hierarchyId	Mandatory	Integer	Identifier of the hierarchy.
parentHierarchyId	Mandatory	Integer	Parent hierarchy identifier.
parentHierarchyName	Optional	String	Parent hierarchy name.
name	Optional	String	Name of the hierarchy.
description	Optional	String	Description of the hierarchy.
childHierarchies	Optional	Array	Identifier(s) of immediate child hierarchies.
referenceSets	Optional	Array	Identifier(s) of the reference sets associated with the hierarchy.
downloadOn	Optional	String	Option to indicate when to download.
ipStartingAddress	Optional	String	Starting IP Address of the hierarchy. Optional to add the entity.
ipEndingAddress	Optional	String	Ending IP Address of the hierarchy. Optional to add the entity.
downloadAutomationEnabled	Optional	Boolean	Indicates, if automated downloads should be performed for the devices under this hierarchy.
inheritReferenceSet	Optional	Boolean	Indicates, if this hierarchy inherits Reference Set from its parent hierarchy. By default, this is enabled.

entityId	Optional	String	Uuid is linked with the hierarchy. Optional to add the entity. Value can be modified after the entity is created.
locationIdentifier	Optional	String	Location identifier of the hierarchy.

Output Parameters

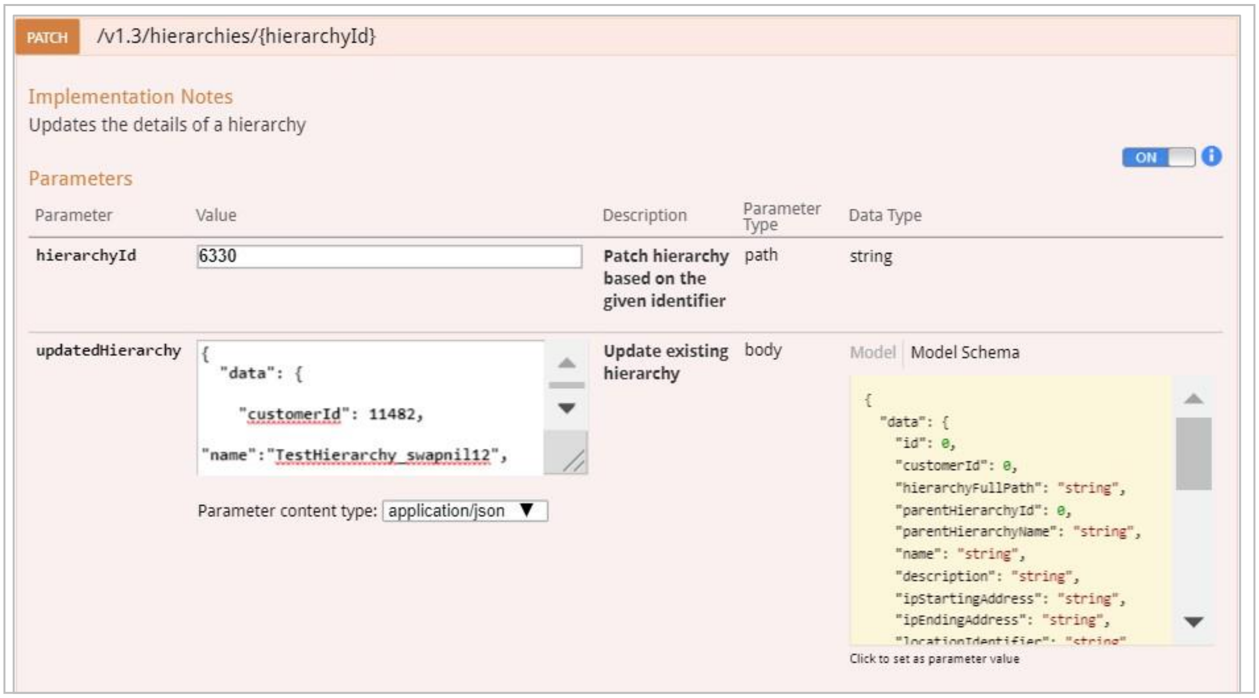
None

Update Hierarchy API - Sample Code

SAMPLE CODE SNIPPET

```
public class UpdateHierarchy {
   private static String url = "http://<servername:port>/apis/v1.3/";
   private static String requestBody = "{\r\n" +
         "  \"data\": {\r\n" +
         "\r\n" +
         " \r\n" +
         "\r\n" +
         "    \"customerId\": 11482,\r\n" +
         "    \"name\":\"TestHierarchy1\",\r\n" +
         "    \r\n" +
         "        \"referenceSets\":[\r\n" +
         "        {\r\n" +
         "            \"id\":106470,\r\n" +
         "          \"name\":\"CM5-N-2.0.10\"\r\n" +
         "\r\n" +
         "        }\r\n" +
         "        ]\r\n" +
         "  }\r\n" +
         "}"; // Patch Request body in JSON format
   /**
    * @param Api -> This refers to the endpoint of the URL for the particular entity e.g. for Hierarchy entity it is 
hierarchies
    * @return
    */
   HttpHeaders getHeader(String Api) {
      HttpHeaders headers = new HttpHeaders();
      headers.set("Authorization", "Bearer 93638536-242a-370a-9d50-71a4959b6390");
      headers.setContentType(MediaType.APPLICATION_JSON);
      return headers;
   }
   /**
    * @param Api ->  This refers to the endpoint of the URL for the particular entity e.g. for Hierarchy entity it is 
hierarchies
    * @param body -> The request body which is required to be patched, This changes as per the entity
    * @param id   -> This refer to which specific Row need to cheange
    */
   void patch(String api, String body, int id,String [] queryparameter,String queryParamValue) {
      String BASE_PATH = url.concat(api) + "/" + id;
      // Getting All Header Information
      // like(contentType,CustomerName,CustomerId)

      if(null!=queryparameter){
         for(String fields: queryparameter){
            BASE_PATH = BASE_PATH +"?"+fields+"="+queryParamValue;
            System.out.println("URL :::: "+BASE_PATH);
         }
      }
      HttpHeaders headers = getHeader(api);

      int statusCode;
      try {
         HttpEntity<String> entity = new HttpEntity<String>(body, headers);
         RestTemplate restTemplate = new RestTemplate();
         HttpComponentsClientHttpRequestFactory requestFactory = new HttpComponentsClientHttpRequestFactory();
         requestFactory.setConnectTimeout(18000);
         requestFactory.setReadTimeout(18000);
         restTemplate.setRequestFactory(requestFactory);
         //HttpEntity<String> result = restTemplate.exchange(BASE_PATH, HttpMethod.PATCH, entity, String.class);
         ResponseEntity<String> responseEntity = restTemplate.exchange(BASE_PATH, HttpMethod.PATCH, entity,
String.class);
         statusCode =  responseEntity.getStatusCode().value();
         System.out.println(statusCode);
      } catch (Exception ex) {
         ex.printStackTrace();
         if (ex.getMessage().equalsIgnoreCase("404 Not Found")) {
            statusCode = 404;
            throw new RuntimeException("Not Found " + statusCode);
         } else if (ex.getMessage().equalsIgnoreCase("400 Bad Request")) {
            statusCode = 400;
            throw new RuntimeException("Bad Request " + statusCode);
         } else if (ex.getMessage().equalsIgnoreCase("409 Conflict")) {
            statusCode = 409;
            throw new RuntimeException("Already Exist : Conflict " + statusCode);
         } else {
            statusCode = 500;
            throw new RuntimeException("Already Exist : Conflict " + statusCode);
         }
      }
      System.out.println("Updated Successfully");
   }
   public static void main(String[] args) {
      String [] queryparameter= {"customerId"};
      String queryParamValue = "11482";
      new UpdateHierarchy().patch("hierarchies", requestBody, 6332,queryparameter,queryParamValue);
   }
}
```

Sample using Swagger - Request



Implementation Notes

Updates the details of a hierarchy

Parameters

Parameter	Value	Description	Parameter Type	Data Type
hierarchyId	6330	Patch hierarchy based on the given identifier	path	string
updatedHierarchy	{ "data": { "customerId": 11482, "name":"TestHierarchy_swapnil12",	Update existing hierarchy	body	Model Model Schema

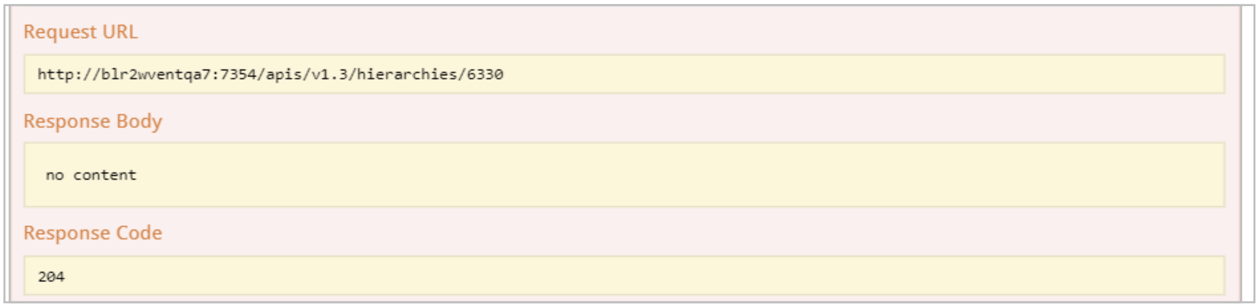
```
{
  "data": {
    "customerId": 11482,
"name":"TestHierarchy_swapnil12",
```

Parameter content type: application/json ▼

```
{
  "data": {
    "id": 0,
    "customerId": 0,
    "hierarchyFullPath": "string",
    "parentHierarchyId": 0,
    "parentHierarchyName": "string",
    "name": "string",
    "description": "string",
    "ipStartingAddress": "string",
    "ipEndingAddress": "string",
    "locationIdentifier": "string"
```

Click to set as parameter value

Sample using Swagger – Response



Request URL

```
http://blr2wventqa7:7354/apis/v1.3/hierarchies/6330
```

Response Body

```
no content
```

Response Code

```
204
```

Error Codes

HTTP Status Code	Application Code	Name	Severity	Message
201	S201	S_CREATED	-	Success.
401	E101	E_TOKEN_INVALID_OR_EXPIRED	2	Token expired or invalid.
401	E102	E_UNAUTHORIZED_ACCESS	2	The user does not have the right to perform the operation.
500	E103	E_GENERAL_ERROR	3	API failed.

DEVICE MANAGEMENT APIs

The user can manage the devices by organizing them as part of the hierarchies and managing the software applications on the device using Packages.

CREATE DEVICE

This API is used to create a new device with a specific model ID and a hierarchy ID with a set of software packages.

> **NOTE** The user should know the entity details, or they can call the get devices API for information. The user can also use the names of the entities instead of IDs while creating the device. E.g., modelName, hierarchyName etc.

Mandatory field

Device model - the user can provide either modelId or modelName.

Optional fields

1. The hierarchy to which the device should be part is specified by providing the hierarchyId or hierarchyName.
   Note: hierarchyFullPath value should be given in the hierarchyName field of the request body.
2. Group

3. SoftwareConfigurations:

- Packages (Software applications)
- KeyHandles
- ParameterTemplates:
    a. Applications – Check whether the application is associated with the package of the software configurations
    b. Parameter template of the application

4. ParameterConfigurations:

- Applications
- Parameters

Request Body

```
{
  "data": {
    "customerId": 0,
    "serialNumber": "string",
    "logicalId": "string",
    "modelId": 0,
    "modelName": "string",
    "autoDownload": "string",
    "hierarchyId": 0,
    "hierarchyName": "string",
    "groups": [
      {
        "id": 0,
        "name": "string"
      }
    ],
    "softwareConfigurations": {
            "softwareAssignmentType": "REFERENCE_SET",
            "directAssignment": true,
            "downloadOn": "NEXT_CONTACT",
            "referenceSetName": "string",
            "referenceSetId": 0,
              "software": [
                {
                  "type": "Package",
                  "id": 0,
                  "name": "string"
                }
            ],
            "parameterTemplates": [
            {
              "applicationId": 0,
              "applicationName": "string",
              "applicationVersion": "string",
              "templateName": "string",
              "templateId": 0
            }
            ]
      },
      "parameterConfigurations": [
        {
          "applicationId": 0,
          "applicationName": "string",
          "applicationVersion": "string",
          "parameters": [
            {
              "name": "string",
              "value": "string"
            }
          ]
        }
      ],
      "customField1": "string",
      "customField2": "string",
      "customField3": "string"
    }
}
```

Once you have the device details, go to the following path:

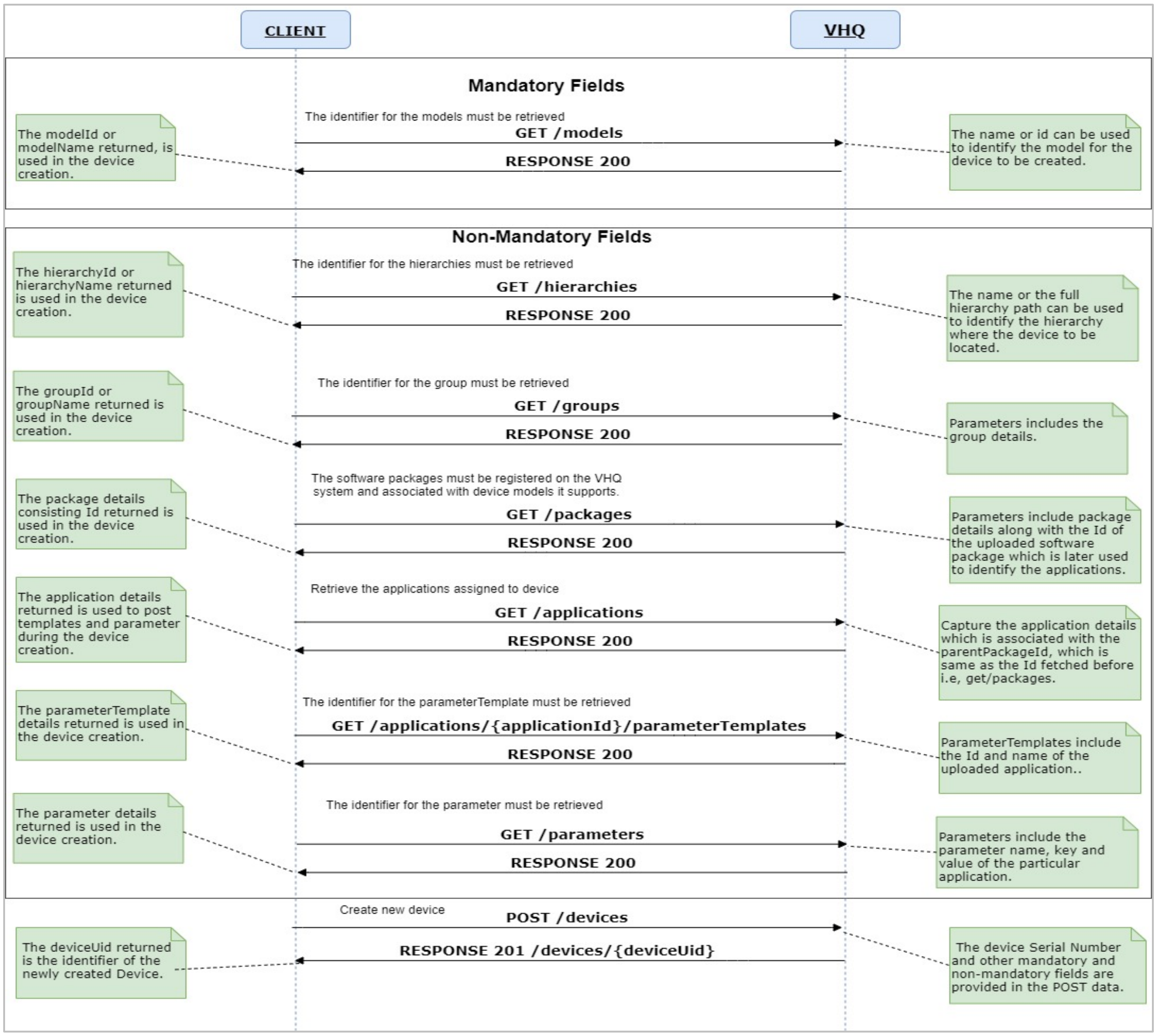
```
API - POST/devices
URL - https://<API Server>/apis/v1.3/devices
e.g. https://vhqtest.verifone.com/apis/v1.3/devices
```

NOTE

If the Hierarchy ID is not mentioned in the request body, then, by default, the device will be created and placed under the root Hierarchy.

Create a new hierarchy using **Create Hierarchy** API before placing the device under the new hierarchy.

Sequence Diagram


CLIENT
VHQ
Mandatory Fields
The identifier for the models must be retrieved
GET /models
RESPONSE 200
The modelId or modelName returned, is used in the device creation.
The name or id can be used to identify the model for the device to be created.
Non-Mandatory Fields
The identifier for the hierarchies must be retrieved
GET /hierarchies
RESPONSE 200
The hierarchyId or hierarchyName returned is used in the device creation.
The name or the full hierarchy path can be used to identify the hierarchy where the device to be located.
The identifier for the group must be retrieved
GET /groups
RESPONSE 200
The groupId or groupName returned is used in the device creation.
Parameters includes the group details.
The software packages must be registered on the VHQ system and associated with device models it supports.
GET /packages
RESPONSE 200
The package details consisting Id returned is used in the device creation.
Parameters include package details along with the Id of the uploaded software package which is later used to identify the applications.
Retrieve the applications assigned to device
GET /applications
RESPONSE 200
The application details returned is used to post templates and parameter during the device creation.
Capture the application details which is associated with the parentPackageId, which is same as the Id fetched before i.e, get/packages.
The identifier for the parameterTemplate must be retrieved
GET /applications/{applicationId}/parameterTemplates
RESPONSE 200
The parameterTemplate details returned is used in the device creation.
ParameterTemplates include the Id and name of the uploaded application..
The identifier for the parameter must be retrieved
GET /parameters
RESPONSE 200
The parameter details returned is used in the device creation.
Parameters include the parameter name, key and value of the particular application.
Create new device
POST /devices
RESPONSE 201 /devices/{deviceUid}
The deviceUid returned is the identifier of the newly created Device.
The device Serial Number and other mandatory and non-mandatory fields are provided in the POST data.

Input Parameters

Element	Optional/Mandatory	Data Type	Description
serialNumber	Mandatory	String	The serial number of the device.
modelId	Mandatory	String	Internal model identifier.
hierarchyId	Optional	Int	Hierarchy identifier.
logicalId	Optional	String	The user-defined Device Id is similar to the Terminal Id.
groups	Optional	Array	The array of Group Identifier or Group names.
downloadOn	Optional	String	Option to indicate whether the download is ON.
softwareConfigurations/software	Optional	Array	The array of Packages and KeyHandles.
softwareConfigurations/parametertemplates	Optional	Array	Parameter templates of a particular application. **NOTE**: Application is mandatory while providing the parameter templates.
parameterConfigurations/parameters	Optional	Array	The array of parameter names and the parameter values. **NOTE**: Application is mandatory while providing the parameters.

Output Parameters

Element	Data Type	Description
udi	int	Unique device Id of the inserted device.

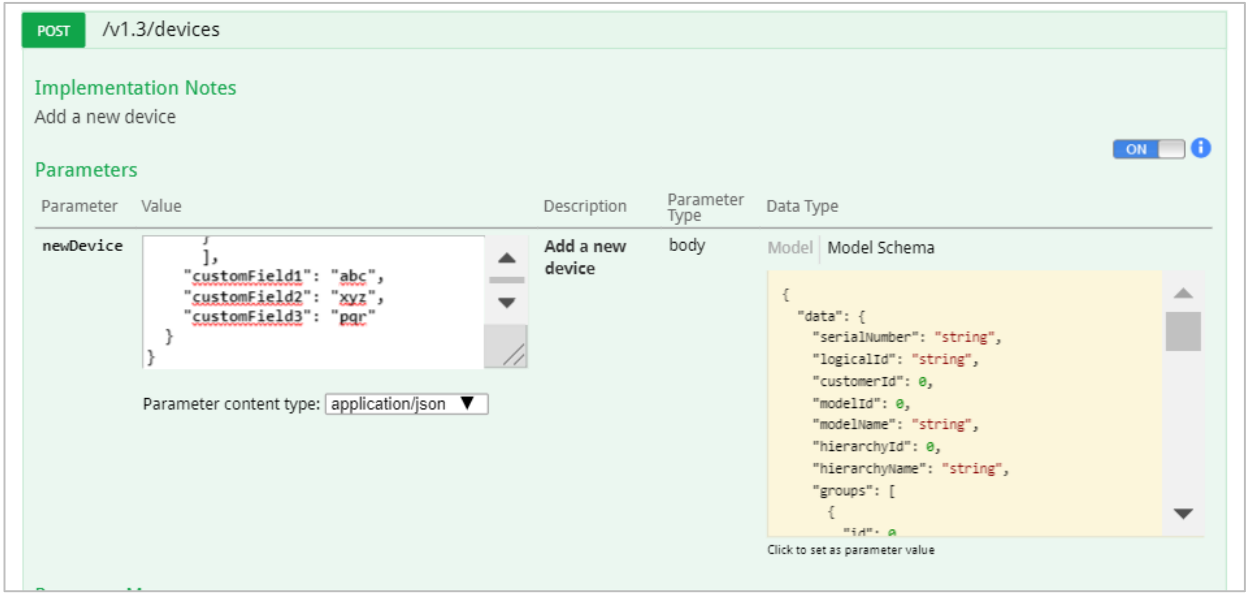
Create Device API - Sample Code

SAMPLE CODE SNIPPET

```
public class CreateDevice {

   private  static String url = "http://<servername:port>/apis/v1.3/";
   // This is the base URL, it will be appended as per the entity which is invoked
   // Request body
   private static String postparameter = "{\r\n" +
         "  \"data\": {\r\n" +
         "     \"serialNumber\": \"4-45-867\",\r\n" +
         "     \"modelName\": \"CM5\",\r\n" +
         "     \"hierarchyId\": \"6215\",\r\n" +
         "     \"customerId\": 11482,\r\n" +
         "     \"groups\": [{\r\n" +
         "           \"name\": \"01AutoGroup\"\r\n" +
         "        },\r\n" +
         "        {\r\n" +
         "           \"name\": \"01ShyCsv\"\r\n" +
         "        }\r\n" +
         "     ],\r\n" +
         "     \"softwareConfigurations\": {\r\n" +
         "        \"downloadOn\": \"NEXT_CONTACT\",\r\n" +
         "        \"softwareAssignmentType\": \"PACKAGE\",\r\n" +
         "        \"directAssignment\": true,\r\n" +
         "        \"software\": [{\r\n" +
         "              \"type\": \"PACKAGE\",\r\n" +
         "              \"id\": 100045\r\n" +
         "           },\r\n" +
         "           {\r\n" +
         "              \"type\": \"KEYHANDLE\",\r\n" +
         "              \"id\": 100549\r\n" +
         "           }\r\n" +
         "        ],\r\n" +
         "        \"parameterTemplates\": [{\r\n" +
         "              \"applicationId\": 114239,\r\n" +
         "              \"templateId\": 21461\r\n" +
         "           },\r\n" +
         "           {\r\n" +
         "              \"applicationName\": \"com.verifone.xyz \",\r\n" +
         "              \"templateId\": 21461\r\n" +
         "           }\r\n" +
         "        ]\r\n" +
         "     },\r\n" +
         "     \"parameterConfigurations\": [\r\n" +
         "\r\n" +
         "        {\r\n" +
         "           \"applicationName\": null,\r\n" +
         "           \"applicationVersion\": \"0.24.5\",\r\n" +
         "           \"parameters\": [{\r\n" +
         "              \"name\": \"Merchant.*/MerchantId\",\r\n" +
         "              \"value\": \"4\",\r\n" +
         "              \"key\": true\r\n" +
         "           }]\r\n" +
         "        }\r\n" +
         "\r\n" +
         "\r\n" +
         "     ],\r\n" +
         "     \"customField1\": \"abc\",\r\n" +
         "     \"customField2\": \"xyz\",\r\n" +
         "     \"customField3\": \"pqr\"\r\n" +
         "  }\r\n" +
         "}"; // Post Request body in JSON format

   /**
    * @param completeURL -> This is the URL which is appended as per the
   entity which is invoked.
    * @param method -> This refer to which method we are going to
   send e.g. get,post,put.
    * @return -> httpCon with Content Type and Authorization token.
    * @throws IOException
    */
   HttpURLConnection getConnection(URL completeURL, String method) throws IOException {
      HttpURLConnection httpCon = (HttpURLConnection)completeURL.openConnection();
      httpCon.setDoOutput(true);
      httpCon.setRequestProperty("Authorization", "Bearer e9bce2f8-d2bc-30c3-991b-eafe85ecadea");
      httpCon.setRequestMethod(method);
      httpCon.setRequestProperty("Content-Type", "application/json");
      return httpCon;
   }
   /**
    * This method creates a new entity.
    * @param body -> The request body which is required to be posted,
   These changes are as per the entity.
    * @param Api -> This refers to the endpoint of the URL for the
   particular entity e.g. for Device entity it is Devices
```

```
    */
   String post(String body, String apiEndPoint, String [] queryparameter, String queryParamValue) {
      url = url.concat(apiEndPoint);
      String id = "";
      if(null!=queryparameter){
         for(String fields: queryparameter){
            url = url +"?"+fields+"="+queryParamValue;
            System.out.println("URL :::: "+url);
         }
      }
      try {
         System.out.println("Url :::::::: " +url);
         URL completeURL = new URL(url);
         // Getting All Header Information

         HttpURLConnection httpCon = getConnection(completeURL,
               "POST");
         OutputStream os = httpCon.getOutputStream();
         os.write(postparameter.getBytes());
         os.flush();
         if (httpCon.getResponseCode() != HttpURLConnection.HTTP_CREATED) {
            if (httpCon.getResponseCode() == 409) {
               throw new RuntimeException(apiEndPoint + "Already Exist : Conflict " +
httpCon.getResponseCode());
            }
         }
         for(java.util.Map.Entry<String, List<String>> entry: httpCon.getHeaderFields().entrySet()){
            if(null !=entry.getKey()
                  &&entry.getKey().toLowerCase().contains("new")){
               id = entry.getValue().get(0);
               id = id.substring(id.lastIndexOf("/")+1);
            }
         }
         System.out.println(id);
         System.out.println(httpCon.getHeaderFields());
         System.out.println("Response Code for POST method->"+httpCon.getResponseCode());
         System.out.println(httpCon.getResponseMessage());
      } catch (Exception ex) {
         int statusCode ;
         ex.printStackTrace();
         if (ex.getMessage().equalsIgnoreCase("404 Not Found")) {
            statusCode = 404;
            throw new RuntimeException("Not Found " + statusCode);
         }
         else if (ex.getMessage().equalsIgnoreCase("400 Bad Request")) {
            statusCode = 400;
            throw new RuntimeException("Bad Request " + statusCode);
         }
         else if (ex.getMessage().equalsIgnoreCase("409 Conflict"))
         {
            statusCode = 409;
            throw new RuntimeException("Already Exist : Conflict " +
                  statusCode);
         } else {
            statusCode = 500;
            throw new RuntimeException("Already Exist : Conflict " +
                  statusCode);
         }
      }
      //System.out.println("Added Successfully");
      return id;
   }
   public static void main(String[] args) {
      //Post method will accept two parameter, Request Body and API
      System.out.println(":::::::: Create Device calling start ::::::::");
      String [] queryparameter= {"customerId"};
      String queryParamValue = "11482";
      CreateDevice = new CreateDevice();
      createDevice.post(postparameter, "devices",queryparameter,queryParamValue);
      System.out.println(":::::::: Create Device calling End ::::::::");
   }
}
```

Sample using Swagger - Request



POST /v1.3/devices

Implementation Notes

Add a new device

Parameters

Parameter	Value	Description	Parameter Type	Data Type
newDevice		Add a new device	body	Model Model Schema

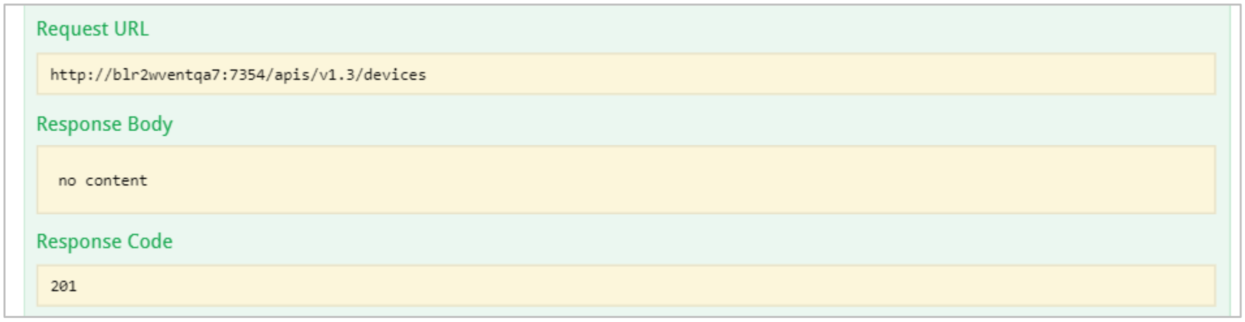
```
      ],
    "customField1": "abc",
    "customField2": "xyz",
    "customField3": "pqr"
  }
}
```

Parameter content type: application/json

```
{
  "data": {
    "serialNumber": "string",
    "logicalId": "string",
    "customerId": 0,
    "modelId": 0,
    "modelName": "string",
    "hierarchyId": 0,
    "hierarchyName": "string",
    "groups": [
      {
        "id": 0
```

Click to set as parameter value

Sample using Swagger – Response



Request URL

```
http://blr2wventqa7:7354/apis/v1.3/devices
```

Response Body

```
no content
```

Response Code

```
201
```

Error Codes

HTTP Status Code	Application Code	Name	Severity	Message
201	S201	S_CREATED	-	Success.
401	E101	E_TOKEN_INVALID_OR_EXPIRED	2	Token is expired or invalid.
401	E102	E_UNAUTHORIZED_ACCESS	2	The user does not have the right to perform the operation.
500	E103	E_GENERAL_ERROR	3	API failed.

409	E116	E_DEVICE_ALREADY_EXISTS	2	Device already exists.
500	E128	E_DEVICE_COUNT_EXCEED	3	Device count exceeded as per license.
500	E130	EX_HIERARCHY_DOES_NOT_EXISTS	3	Invalid hierarchy id.

UPDATE DEVICES-1.4

This API is used to update the device details. The location ID of the parent device is updated along with connected child devices. The location ID of the child device cannot be updated directly. When the location ID of the Parent device is updated, the location ID of the child device is updated automatically.

Pre-requisites

Following are the pre-requisites for calling Update Device API.
The user can also provide names instead of IDs as input to update the device.

- Software (Software applications)
- Applications

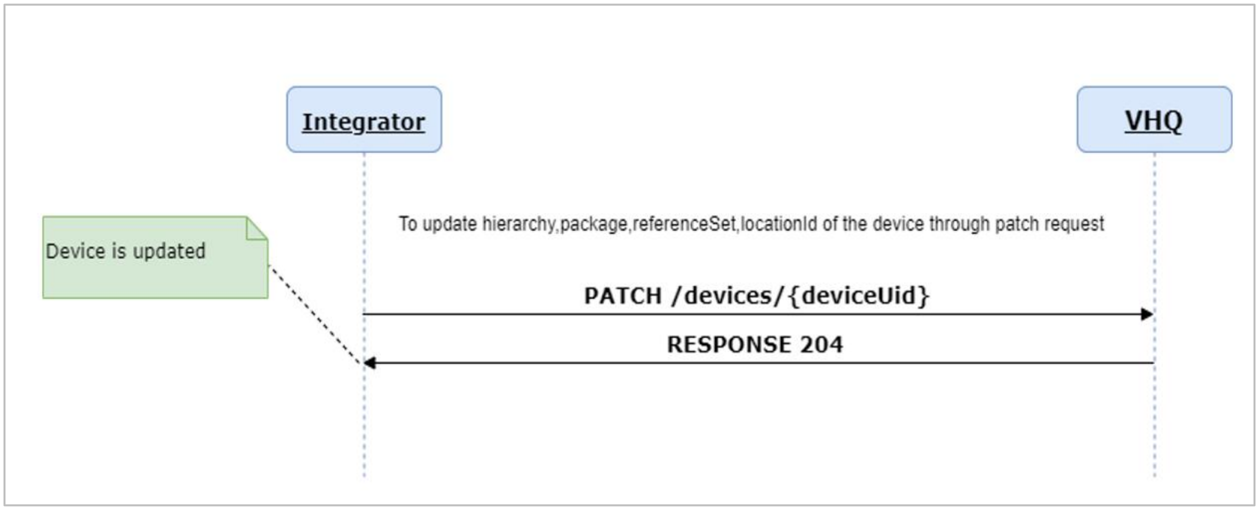
  Note: The applicationName and application version both can be used as criteria instead of applicationId to update the device details.

- KeyHandles

Go to the following path:

```
API - PATCH/devices/{deviceUid}
URL - https://<API Server>/apis/v1.4/devices/{deviceUid}
e.g. https://vhqtest.verifone.com/apis/v1.4/devices/{deviceUid}
```

Sequence Diagram


Integrator
VHQ
Device is updated
To update hierarchy,package,referenceSet,locationId of the device through patch request
PATCH /devices/{deviceUid}
RESPONSE 204

Update Device API - Sample Code

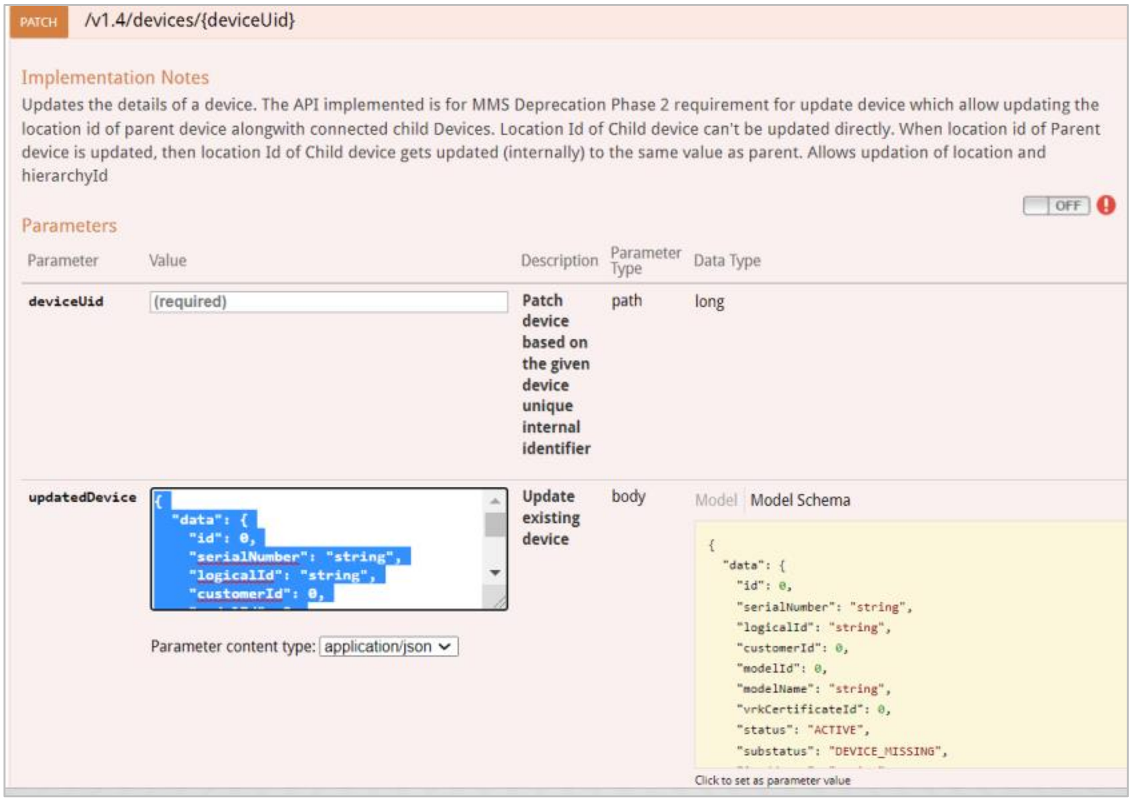
SAMPLE CODE SNIPPET

```
public class UpdateDeviceByPackage {

private static String url = "http://<servername:port>/apis/v1.4/";


//This is the base URL, it will be appended as per the entity which is invoked along with the
corresponding deviceUid

private static String requestBody = "{\r\n" +

"\"data\":\r\n" +

{\r\n" +
\r\n" +
\"customerId\" :11482,\r\n" +
\"softwareConfigurations\":{\r\n" +
\"softwareAssignmentType\":\"PACKAGE\",\r\n" +
\"software\": [\r\n" +

"                  {\r\n" +
"                  \"type\": \"Package\",\r\n" +
"                  \"id\": 100019\r\n" +
"                  },\r\n" +
"                  {\r\n" +

\"type\": \"KeyHandle\",\r\n" +
\"name\": \"A-KEYTST-NIKE-03_01\"\r\n" +
}\r\n" +

"\r\n" +

],\r\n" +
\"parameterTemplates\": [\r\n" +

"                  {\r\n" +
"                  \"applicationId\":114215,\r\n" +
"                  \"applicationName\":\"com.verifone.pos\",\r\n" +
"                  \"templateId\": 21463,\r\n" +
"                  \"templateName\":\"VposParameters\"\r\n" +
"                  \r\n" +
"                  }\r\n" +

\r\n" +
\r\n" +
]\r\n" +
}\r\n" +

"}\r\n" +

"}\r\n" +

""; // Patch Request body in JSON format

/**


@param Api -> This refers to the endpoint of the URL for the particular entity e.g. for Device entity it
is devices
@return

*/

HttpHeaders getHeader(String Api) {
```

```
HttpHeaders headers = new HttpHeaders();


headers.set("Authorization", "Bearer e22abce8-c938-3b47-87ac-8edafca2f12d");
headers.setContentType(MediaType.APPLICATION_JSON); return headers;

}

/**


@param Api -> This refers to the endpoint of the URL for the particular entity e.g. for Device entity it
is devices

@param body -> The request body which is required to be patched, This changes as per the entity


* @param id -> This refer to which specific Row need to change */

void patch(String api, String body, int id) {

int statusCode;


Getting All Header Information

like(contentType,CustomerName,CustomerId) HttpHeaders headers = getHeader(api);
try {
HttpEntity<String> entity = new HttpEntity<String>(body, headers);


RestTemplate restTemplate = new RestTemplate(); HttpComponentsClientHttpRequestFactory requestFactory =
new


HttpComponentsClientHttpRequestFactory(); requestFactory.setConnectTimeout(18000);
requestFactory.setReadTimeout(18000); restTemplate.setRequestFactory(requestFactory);


//HttpEntity<String> result = restTemplate.exchange(BASE_PATH, HttpMethod.PATCH, entity, String.class);


ResponseEntity<String> responseEntity = restTemplate.exchange(BASE_PATH, HttpMethod.PATCH, entity,
String.class);

statusCode =  responseEntity.getStatusCode().value();

System.out.println(statusCode);


} catch (Exception ex) { ex.printStackTrace();

if (ex.getMessage().equalsIgnoreCase("404 Not Found")) { statusCode = 404;

throw new RuntimeException("Not Found " + statusCode);


} else if (ex.getMessage().equalsIgnoreCase("400 Bad Request")) { statusCode = 400;
throw new RuntimeException("Bad Request " + statusCode);

} else if (ex.getMessage().equalsIgnoreCase("409 Conflict")) { statusCode = 409;

throw new RuntimeException("Already Exist : Conflict " + statusCode); } else {

statusCode = 500;

throw new RuntimeException("Already Exist : Conflict " + statusCode);
}

}

}
public static void main(String[] args) {
new UpdateDeviceByPackage().patch("devices", requestBody, 78500); System.out.println("Updated
Successfully");
}
}
```

Sample using Swagger – Request

The body tag has the following request as shown on the screen to update the parameters.



Implementation Notes

Updates the details of a device. The API implemented is for MMS Deprecation Phase 2 requirement for update device which allow updating the location id of parent device alongwith connected child Devices. Location Id of Child device can't be updated directly. When location id of Parent device is updated, then location Id of Child device gets updated (internally) to the same value as parent. Allows updation of location and hierarchyId

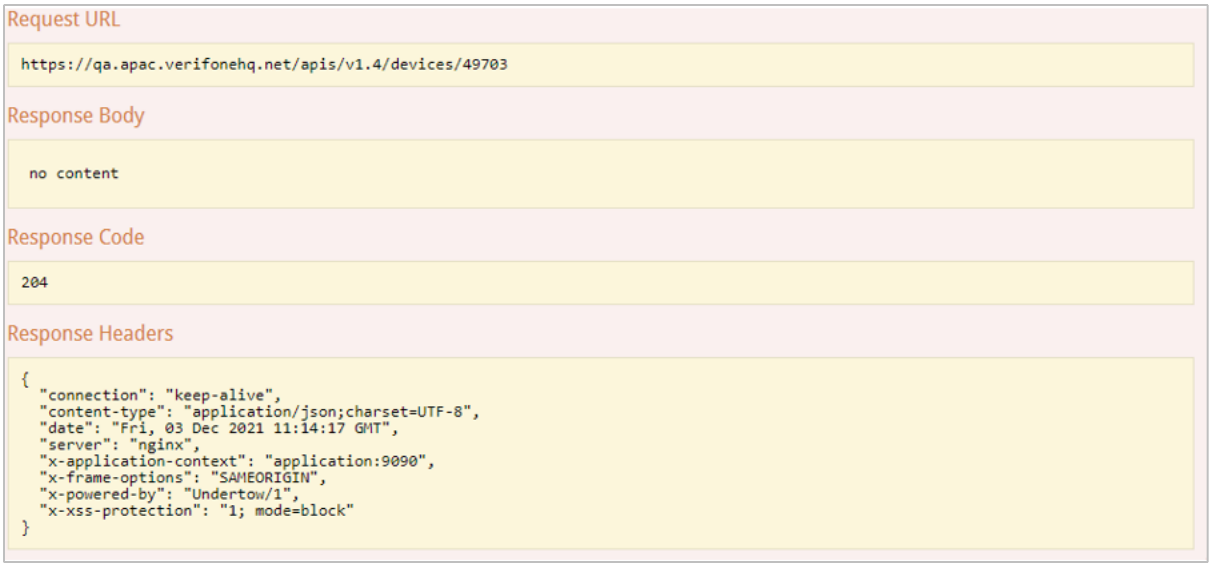
Parameters

Parameter	Value	Description	Parameter Type	Data Type
deviceUid	(required)	Patch device based on the given device unique internal identifier	path	long
updatedDevice	{ "data": { "id": 0, "serialNumber": "string", "logicalId": "string", "customerId": 0, Parameter content type: application/json	Update existing device	body	Model | Model Schema

```
{
  "data": {
    "id": 0,
    "serialNumber": "string",
    "logicalId": "string",
    "customerId": 0,
    "modelId": 0,
    "modelName": "string",
    "vrkCertificateId": 0,
    "status": "ACTIVE",
    "substatus": "DEVICE_MISSING",
```

Sample using Swagger – Response

The updated device details can be validated by fetching the device.



Request URL

```
https://qa.apac.verifonehq.net/apis/v1.4/devices/49703
```

Response Body

```
no content
```

Response Code

```
204
```

Response Headers

```
{
  "connection": "keep-alive",
  "content-type": "application/json;charset=UTF-8",
  "date": "Fri, 03 Dec 2021 11:14:17 GMT",
  "server": "nginx",
  "x-application-context": "application:9090",
  "x-frame-options": "SAMEORIGIN",
  "x-powered-by": "Undertow/1",
  "x-xss-protection": "1; mode=block"
}
```

UPDATE DEVICE

This API is used to update the device details of a specified device Uid.
E.g., a new Hierarchy, a new Group with a specific set of Software Applications and Parameter Templates, etc.

Request Body

```
{
  "data": {
    "customerId": 0,
    "serialNumber": "string",
    "logicalId": "string",
    "modelId": 0,
    "modelName": "string",
    "autoDownload": "string",
    "hierarchyId": 0,
    "hierarchyName": "string",
    "groups": [
      {
        "id": 0,
        "name": "string"
      }
    ],
    "softwareConfigurations": {
            "softwareAssignmentType": "REFERENCE_SET",
            "directAssignment": true,
            "downloadOn": "NEXT_CONTACT",
            "referenceSetName": "string",
            "referenceSetId": 0,
              "software": [
                {
                  "type": "Package",
                  "id": 0,
                  "name": "string"
                }
            ],
            "parameterTemplates": [
            {
              "applicationId": 0,
              "applicationName": "string",
              "applicationVersion": "string",
              "templateName": "string",
              "templateId": 0
            }
            ],
      "customField1": "string",
      "customField2": "string",
      "customField3": "string"
    }
}
```

Update Device API

Scenario 1: To update the device details for hierarchy, groups, or reference set.

Pre-requisites

Following are the pre-requisites for calling the update device:

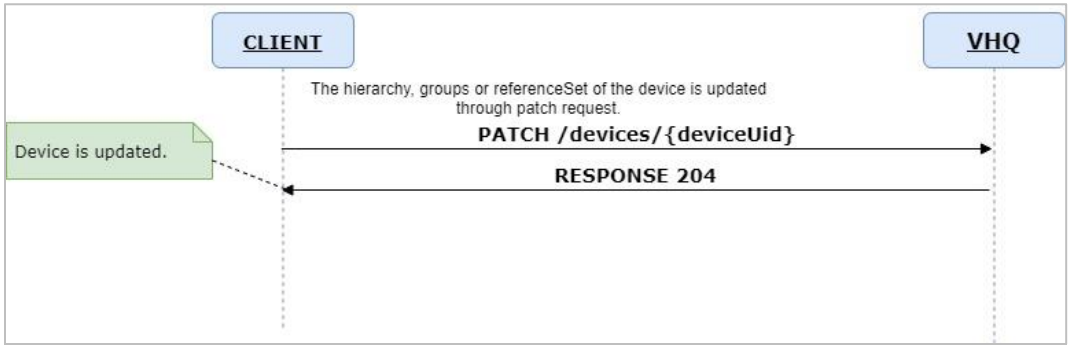
The user can use the names of the entities instead of IDs while updating the device.

- Hierarchy
  NOTE: hierarchy full path value should be given in the hierarchy name field of the request body.
- Groups
- Reference Set

Once you have the device details and the hierarchy details, go to the following path:

```
API - PATCH/devices/{deviceUid}
URL - https://<API Server>/apis/v1.3/devices/{deviceUid}
e.g. https://vhqtest.verifone.com/apis/v1.3/devices/{deviceUid}
```

Sequence Diagram


CLIENT
VHQ
The hierarchy, groups or referenceSet of the device is updated through patch request.
PATCH /devices/{deviceUid}
RESPONSE 204
Device is updated.

Input Parameters

Element	Optional/Mandatory	Data Type	Description
udi	Mandatory	Int	Unique Device identifier.
status	Optional	enum	Status of the device to be updated, enumStatus will contain the below values. • Inactive • Deleted • BlackListed • PendingRegistration
hierarchyId	Optional	Int	New Hierarchy of the device.
groupIds	Optional	String[]	Groups of the device. The device will be added to the specified list of Groups. All current assignments will be removed.

software	Optional	Array	Software packages. The device will be assigned new software.
parameterTemplates	Optional	Array	New Parameter Templates will be assigned to the device.
referenceSet	Optional	String	The device will be assigned a new reference set.

Output Parameters

None

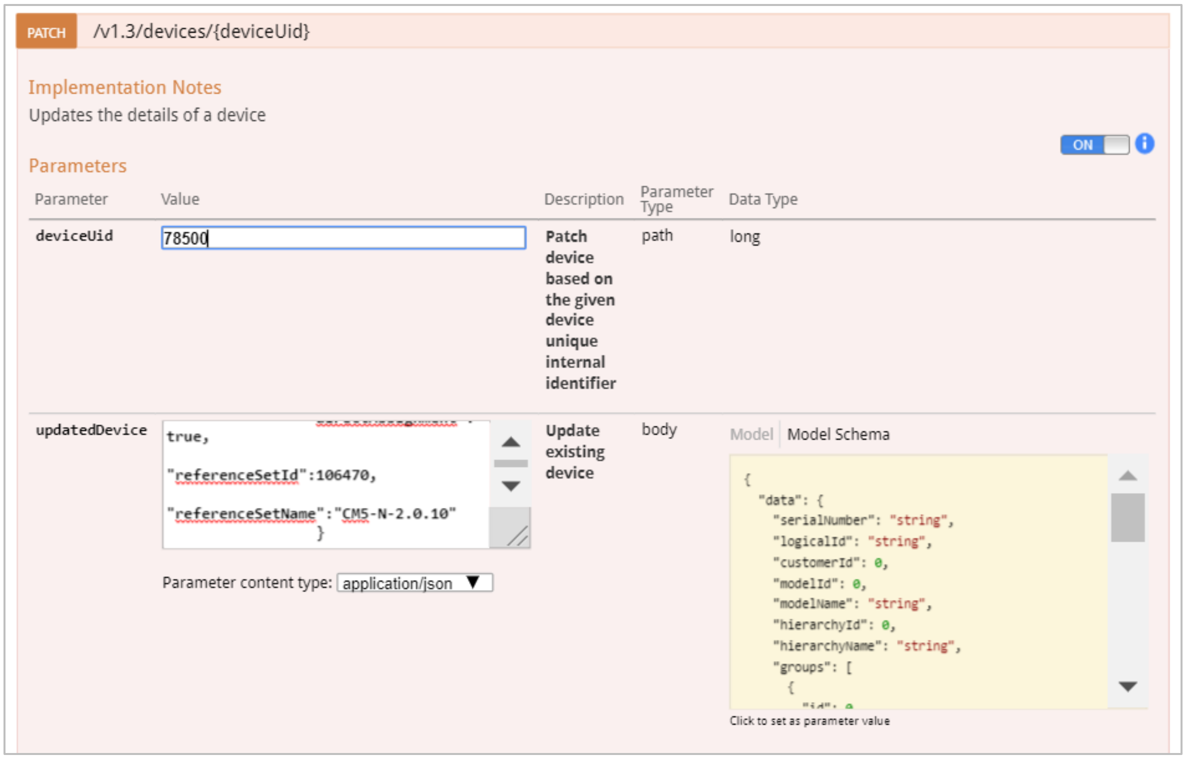
Update Device API - Sample Code

SAMPLE CODE SNIPPET

```
public class UpdateDeviceByHierarchy {
   private static String url = "http://<servername:port>/apis/v1.3/";
   //This is the base URL, it will be appended as per the entity which is invoked along with the
corresponding deviceUid
   private static String requestBody = "{\r\n" +
         "    \"data\": \r\n" +
         "        {\r\n" +
         "            \"customerId\": 11482,\r\n" +
         "            \"hierarchyName\":\"AutomationRootHierarchy\",\r\n" +
         "            \"groups\": [{\r\n" +
         "        \r\n" +
         "        \"name\": \"Test1Group\"\r\n" +
         "\r\n" +
         "     }, {\r\n" +
         "        \"id\": 20782\r\n" +
         "        \r\n" +
         "\r\n" +
         "     }],\r\n" +
         "            \"softwareConfigurations\": {\r\n" +
         "                \"softwareAssignmentType\": \"REFERENCE_SET\",\r\n" +
         "                \"directAssignment\": true,\r\n" +
         "                \"referenceSetId\":106470,\r\n" +
         "                 \"referenceSetName\":\"CM5-N-2.0.10\"\r\n" +
         "                 }\r\n" +
         "            \r\n" +
         "           \r\n" +
         "        }\r\n" +
         "        \r\n" +
         "    }\r\n" +
         "    "; // Patch Request body in JSON format
   /**
    * @param Api -> This refers to the endpoint of the URL for the particular entity e.g. for Device
entity it is devices
    * @return
    */
   HttpHeaders getHeader(String Api) {
      HttpHeaders headers = new HttpHeaders();
      headers.set("Authorization", "Bearer 7b00d958-6bab-33eb-8a96-1a367bb466a5");
      headers.setContentType(MediaType.APPLICATION_JSON);
      return headers;
   }
   /**
    * @param Api ->  This refers to the endpoint of the URL for the particular entity e.g. for Device
entity it is devices
    * @param body -> The request body which is required to be patched, This changes as per the entity
    * @param id   -> This refer to which specific Row need to change
    */
   void patch(String api, String body, int id) {
      int statusCode;
      String BASE_PATH = url.concat(api) + "/" + id;
      // Getting All Header Information
      // like(contentType,CustomerName,CustomerId)
      HttpHeaders headers = getHeader(api);
      try {
         HttpEntity<String> entity = new HttpEntity<String>(body, headers);
         RestTemplate restTemplate = new RestTemplate();
         HttpComponentsClientHttpRequestFactory requestFactory = new 
HttpComponentsClientHttpRequestFactory();
         requestFactory.setConnectTimeout(18000);
         requestFactory.setReadTimeout(18000);
         restTemplate.setRequestFactory(requestFactory);
         //HttpEntity<String> result = restTemplate.exchange(BASE_PATH, HttpMethod.PATCH, entity, 
String.class);
         ResponseEntity<String> responseEntity = restTemplate.exchange(BASE_PATH, HttpMethod.PATCH, 
entity, String.class);
         statusCode =  responseEntity.getStatusCode().value();
         System.out.println(statusCode);

      } catch (Exception ex) {
         ex.printStackTrace();

         if (ex.getMessage().equalsIgnoreCase("404 Not Found")) {
            statusCode = 404;
            throw new RuntimeException("Not Found " + statusCode);
         } else if (ex.getMessage().equalsIgnoreCase("400 Bad Request")) {
            statusCode = 400;
            throw new RuntimeException("Bad Request " + statusCode);
         } else if (ex.getMessage().equalsIgnoreCase("409 Conflict")) {
            statusCode = 409;
            throw new RuntimeException("Already Exist : Conflict " + statusCode);
         } else {
            statusCode = 500;
            throw new RuntimeException("Already Exist : Conflict " + statusCode);
         }
      }
   }
   public static void main(String[] args) {
      new UpdateDeviceByHierarchy().patch("devices", requestBody, 78500);
      System.out.println("Updated Successfully");
   }
}
```

Sample using Swagger - Request

The body tag has the following request as shown on the screen to update the hierarchy of the device.



/v1.3/devices/{deviceUid}

Implementation Notes

Updates the details of a device

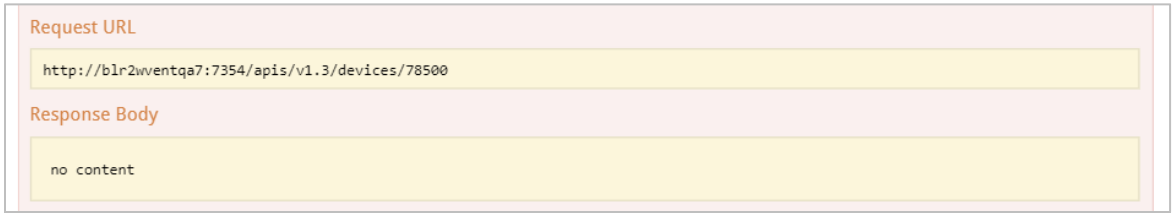
Parameters

Parameter	Value	Description	Parameter Type	Data Type
deviceUid	78500	Patch device based on the given device unique internal identifier	path	long
updatedDevice	true, "referenceSetId":106470, "referenceSetName":"CM5-N-2.0.10" } Parameter content type: application/json	Update existing device	body	Model | Model Schema

```
{
  "data": {
    "serialNumber": "string",
    "logicalId": "string",
    "customerId": 0,
    "modelId": 0,
    "modelName": "string",
    "hierarchyId": 0,
    "hierarchyName": "string",
    "groups": [
      {
        "id": 0
```

Click to set as parameter value

Sample using Swagger – Response



Request URL

```
http://blr2wventqa7:7354/apis/v1.3/devices/78500
```

Response Body

```
no content
```

The updated device details can be validated by fetching the device.

Error Codes

HTTP Status Code	Application Code	Name	Sev erity	Message
200	S200	S_SUCCESS	-	Success.
401	E101	E_TOKEN_INVALID_OR_EXPIRED	2	Token is expired or invalid.
401	E102	E_UNAUTHORIZED_ACCESS	2	The user does not have the right to operate.
500	E103	E_GENERAL_ERROR	3	API failed.

500	E114	E_UPDATEDEVICESTATUS_FAILED	3	Device Status Update failed.
409	E116	E_DEVICE_ALREADY_EXISTS	2	Device already exists.
500	E126	E_DEVICE_ALREADY_DELETED_OR_BLACKLISTED	3	The device is already in the Deleted/Blacklist status.

Update Device API

Scenario 2: To update the package, key handle, or parameter template of the device

Pre-requisites

Following are the pre-requisites for calling the update device:

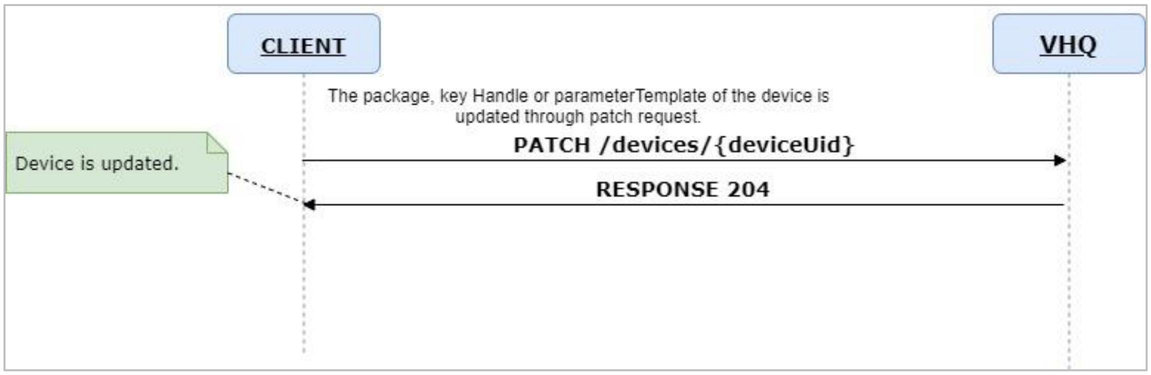
The user can also provide names instead of IDs as input to update the device.

- Software (Software applications)
- Applications
  Note: applicationName and application version can be used instead of applicationId.
- KeyHandles

Once you have the device and the package details, go to the following path:

```
API - PATCH/devices/{deviceUid}
URL - https://<API Server>/apis/v1.3/devices/{deviceUid}
e.g. https://vhqtest.verifone.com/apis/v1.3/devices/{deviceUid}
```

Sequence Diagram


CLIENT
VHQ
The package, key Handle or parameterTemplate of the device is updated through patch request.
PATCH /devices/{deviceUid}
RESPONSE 204
Device is updated.

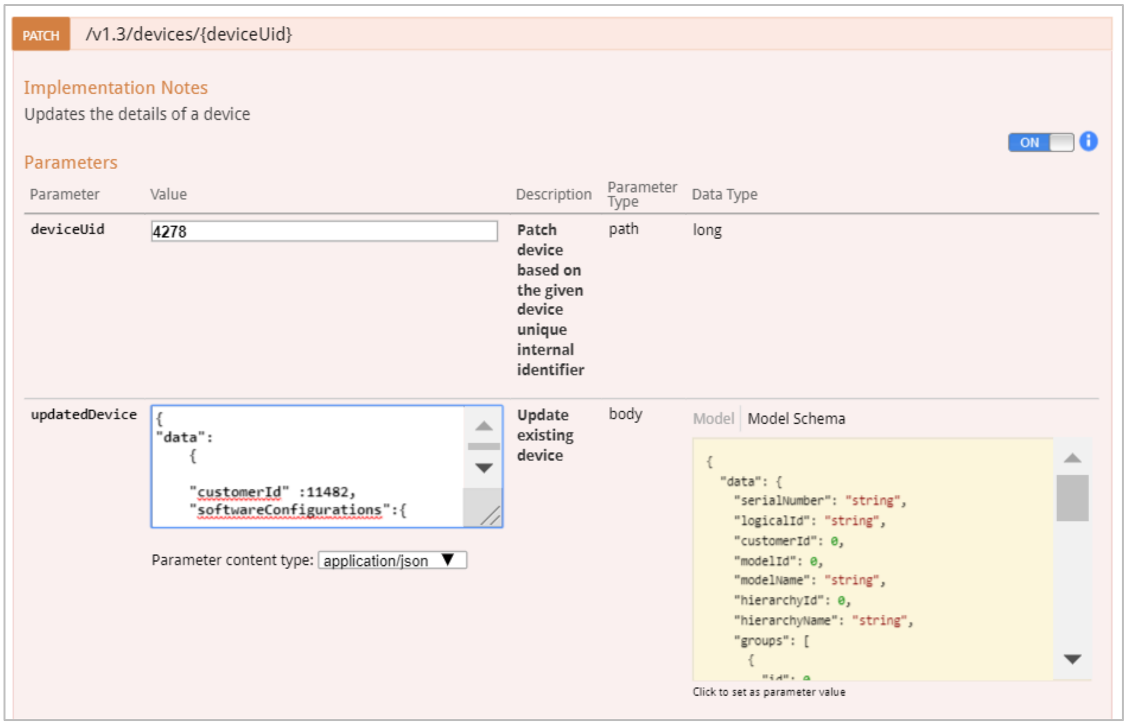
Update Device API - Sample Code

SAMPLE CODE SNIPPET

```
public class UpdateDeviceByPackage {
   private static String url = "http://<servername:port>/apis/v1.3/";
   //This is the base URL, it will be appended as per the entity which is invoked along with the
corresponding deviceUid
   private static String requestBody = "{\r\n" +
         "\"data\":\r\n" +
         "    {\r\n" +
         "   \r\n" +
         "    \"customerId\" :11482,\r\n" +
         "    \"softwareConfigurations\":{\r\n" +
         "        \"softwareAssignmentType\":\"PACKAGE\",\r\n" +
         "        \"software\": [\r\n" +
         "                {\r\n" +
         "                  \"type\": \"Package\",\r\n" +
         "                  \"id\": 100019\r\n" +
         "                },\r\n" +
         "                {\r\n" +
         "           \"type\": \"KeyHandle\",\r\n" +
         "           \"name\": \"A-KEYTST-NIKE-03_01\"\r\n" +
         "        }\r\n" +
         "\r\n" +
         "            ],\r\n" +
         "         \"parameterTemplates\": [\r\n" +
         "                {\r\n" +
         "                  \"applicationId\":114215,\r\n" +
         "                  \"applicationName\":\"com.verifone.pos\",\r\n" +
         "                  \"templateId\": 21463,\r\n" +
         "                  \"templateName\":\"VposParameters\"\r\n" +
         "                 \r\n" +
         "                }\r\n" +
         "               \r\n" +
         "               \r\n" +
         "      ]\r\n" +
         "    }\r\n" +
         "}\r\n" +
         "}\r\n" +
         ""; // Patch Request body in JSON format
   /**
    * @param Api -> This refers to the endpoint of the URL for the particular entity e.g. for Device
entity it is devices
    * @return
    */
   HttpHeaders getHeader(String Api) {
      HttpHeaders headers = new HttpHeaders();
      headers.set("Authorization", "Bearer e22abce8-c938-3b47-87ac-8edafca2f12d");
      headers.setContentType(MediaType.APPLICATION_JSON);
      return headers;
   }
   /**
    * @param Api ->  This refers to the endpoint of the URL for the particular entity e.g. for Device
entity it is devices
    * @param body -> The request body which is required to be patched, This changes as per the entity
    * @param id   -> This refer to which specific Row need to change
    */
   void patch(String api, String body, int id) {
      int statusCode;
      String BASE_PATH = url.concat(api) + "/" + id;
      // Getting All Header Information
      // like(contentType,CustomerName,CustomerId)
      HttpHeaders headers = getHeader(api);
      try {
         HttpEntity<String> entity = new HttpEntity<String>(body, headers);
         RestTemplate restTemplate = new RestTemplate();
         HttpComponentsClientHttpRequestFactory requestFactory = new
HttpComponentsClientHttpRequestFactory();
         requestFactory.setConnectTimeout(18000);
         requestFactory.setReadTimeout(18000);
         restTemplate.setRequestFactory(requestFactory);
         //HttpEntity<String> result = restTemplate.exchange(BASE_PATH, HttpMethod.PATCH, entity,
String.class);
         ResponseEntity<String> responseEntity = restTemplate.exchange(BASE_PATH, HttpMethod.PATCH,
entity, String.class);
         statusCode =  responseEntity.getStatusCode().value();
         System.out.println(statusCode);

      } catch (Exception ex) {
         ex.printStackTrace();

         if (ex.getMessage().equalsIgnoreCase("404 Not Found")) {
            statusCode = 404;
            throw new RuntimeException("Not Found " + statusCode);
         } else if (ex.getMessage().equalsIgnoreCase("400 Bad Request")) {
            statusCode = 400;
            throw new RuntimeException("Bad Request " + statusCode);
         } else if (ex.getMessage().equalsIgnoreCase("409 Conflict")) {
            statusCode = 409;
            throw new RuntimeException("Already Exist : Conflict " + statusCode);
         } else {
            statusCode = 500;
            throw new RuntimeException("Already Exist : Conflict " + statusCode);
         }
      }
   }
   public static void main(String[] args) {
      new UpdateDeviceByPackage().patch("devices", requestBody, 78500);
      System.out.println("Updated Successfully");
   }
}
```

Sample using Swagger – Request

The body tag has the following request as shown on the screen to update the parameters.



Implementation Notes

Updates the details of a device

Parameters

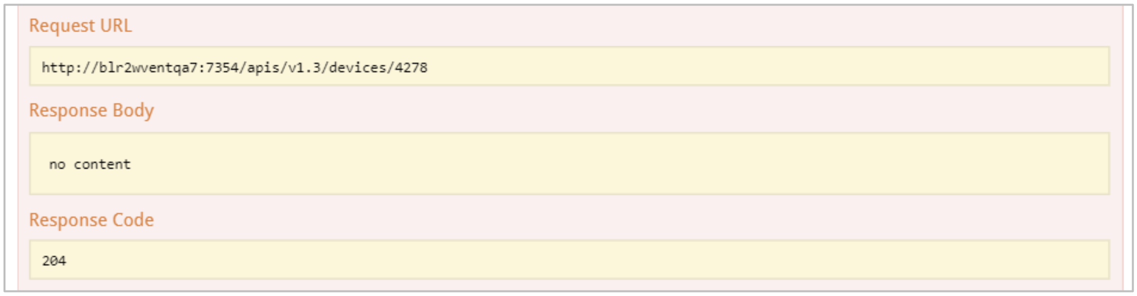
Parameter	Value	Description	Parameter Type	Data Type
deviceUid	4278	Patch device based on the given device unique internal identifier	path	long
updatedDevice	{ "data": { "customerId" :11482, "softwareConfigurations":{ Parameter content type: application/json	Update existing device	body	Model / Model Schema

```
{
"data":
    {

    "customerId" :11482,
    "softwareConfigurations":{
```

```
{
  "data": {
    "serialNumber": "string",
    "logicalId": "string",
    "customerId": 0,
    "modelId": 0,
    "modelName": "string",
    "hierarchyId": 0,
    "hierarchyName": "string",
    "groups": [
      {
        "id": 0
```

Click to set as parameter value

Sample using Swagger – Response



Request URL

```
http://blr2wventqa7:7354/apis/v1.3/devices/4278
```

Response Body

```
no content
```

Response Code

```
204
```

The updated device details can be validated by fetching the device.

GET DEVICES-1.4

This API is used to get the list of all devices except deleted and blacklisted devices under the root hierarchy that are accessible to the logged-in user. Pagination is supported for this API.

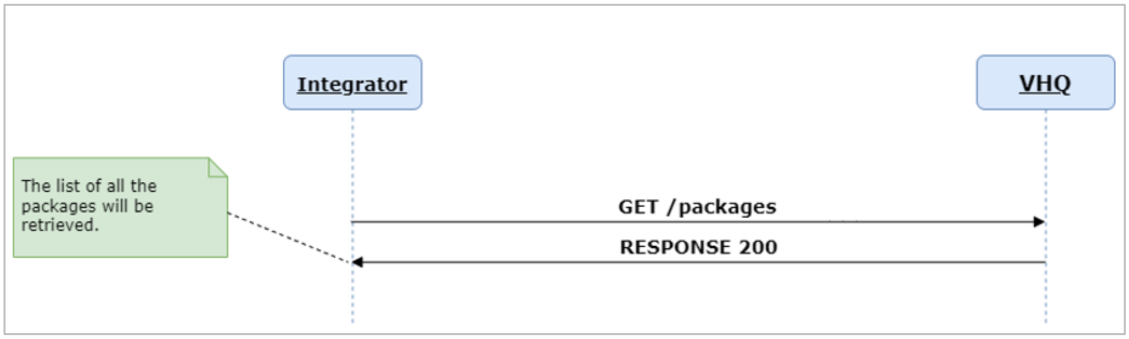
Get Devices API

To get the list of all device details such as: Device ID, Serial Number, Model ID, Model Name, Status, Sub-status, IP Address, MAC Address, and Location ID.

Go to the following path:

```
API - GET/devices

URL - https://<API Server>/apis/v1.4/devices

e.g.  https://vhqtest.verifone.com/apis/v1.4/devices
```

Sequence Diagram



Input Parameters

Input parameters must be sent in the query string.

Element	Optional/Mandatory	Data Type	Description
serialNumber	Optional	String	The serial number of the device.
modelId	Optional	Int	The model ID of the device.
deviceId	Optional	String	Device Identification.
hierarchyId/hierarchyname	Optional	Int	Hierarchy Id.
groupId/groupName	Optional	Int	Group Id.
applications	Optional	String	The array of applications. Value format with Application = {Application Name}: {Version}.
reportedApplications	Optional	String	The array of reported applications. Value format with Application = {Application Name}: {Version}.

packageName	Optional	String	The array of packages.
limit	Optional	Int	Rows per request.
offset	Optional	Int	Offset required to calculate the page number.

Output Parameters

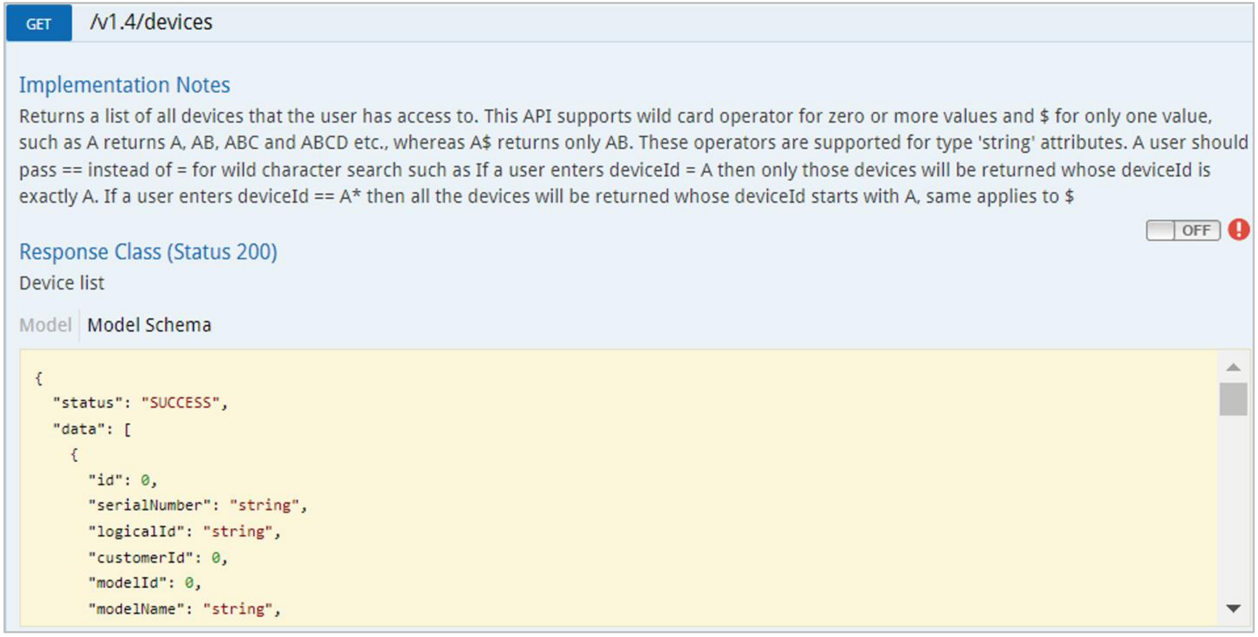
Element	Attribute	Data Type
devices (collection of devices)	uniqueDeviceId	Int
	serialNumber	String
	deviceId	String
	status	String
	model	String
	hierarchyFullPath	String
	locationId	String

Get Devices API - Sample Code

SAMPLE CODE SNIPPET

```
public class GetDevices {

   private static String url = "http://<servername:port>/apis/v1.4/";
   private int statusCode;
   // This is the base URL, it will be appended as per the entity which is invoked

   /**
    * @param completeURL -> This is the URL which appended as per the
   entity which is invoked
    * @param method -> This refer to which method we are going to
   send e.g. get,post,put
    * @return -> httpCon with Content Type and Authorization token
    * @throws IOException
    */
   HttpURLConnection getConnection(URL completeURL, String method)
         throws IOException {
      HttpURLConnection httpCon = (HttpURLConnection)
            completeURL.openConnection();
      httpCon.setDoOutput(true);
      httpCon.setRequestProperty("Authorization", "Bearer 29797eae-3879-3827-8ec3-
5295796c5af1");
      httpCon.setRequestMethod(method);
      httpCon.setRequestProperty("Content-Type", "application/json");
      return httpCon;
   }
   /**
    * This method get all data of a entity
    * @param API -> This refers to the endpoint of the URL for the
   particular entity e.g. for Device entity it is devices
    * @param id -> This refer to show specific data from Entity based
   on Id.
    * @param fields -> This refers to show specific fields, which we
   pass as a parameter.
    * @param queryparameter
    * @return response from entity
    */
   String get(String Api, int id,String fullPath,String[] queryparameter,String 
queryParamValue) {
      StringBuffer response=null;
      if(id!=0) {
         url = url.concat(Api) + "/" + id;
      }else {
         url = url.concat(Api);
      }
      
      if(fullPath!=null){
         url=url+"/".concat(fullPath);
      }
      if(null!=queryparameter){
         for(String fields: queryparameter){
            url = url +"?"+fields+"="+queryParamValue;
            
            
            System.out.println("URL :::: "+url);
         }
      }
      try {
         URL completeURL = new URL(url);
         // Getting All Header Information
         // like(contentType,CustomerName,CustomerId)
         HttpURLConnection httpCon = getConnection(completeURL, "GET");
         httpCon.getResponseCode();
         BufferedReader in = new BufferedReader(new 
InputStreamReader(httpCon.getInputStream()));
         String inputLine;
         response =new StringBuffer();
         while ((inputLine = in.readLine()) != null) {
            response.append(inputLine);
         }
         System.out.println(response);
         System.out.println(httpCon.getResponseCode());
         System.out.println(httpCon.getResponseMessage());
      } catch (Exception ex) {
         ex.printStackTrace();
         if (ex.getMessage().equalsIgnoreCase("404 Not Found")) {
            statusCode = 404;
            throw new RuntimeException("Not Found " + statusCode);
            
         } else if (ex.getMessage().equalsIgnoreCase("400 Bad Request")) {
            statusCode = 400;
            throw new RuntimeException("Bad Request " + statusCode);
         } else if (ex.getMessage().equalsIgnoreCase("409 Conflict"))
         {
            statusCode = 409;
            throw new RuntimeException("Already Exist : Conflict " +
                  statusCode);
         } else {
            statusCode = 500;
            throw new RuntimeException("Already Exist : Conflict " +
                  statusCode);
         }
      }
      return response.toString();
   }
   public static void main(String[] args) {
      String [] queryparameter= {"customerId"};
      String queryParamValue = "11482";
      new GetDevices().get("devices",0,null,queryparameter,queryParamValue);
   }
}
```

Sample using Swagger - Request



/v1.4/devices

Implementation Notes

Returns a list of all devices that the user has access to. This API supports wild card operator for zero or more values and $ for only one value, such as A returns A, AB, ABC and ABCD etc., whereas A$ returns only AB. These operators are supported for type 'string' attributes. A user should pass == instead of = for wild character search such as If a user enters deviceId = A then only those devices will be returned whose deviceId is exactly A. If a user enters deviceId == A* then all the devices will be returned whose deviceId starts with A, same applies to $

Response Class (Status 200)

Device list

```
{
  "status": "SUCCESS",
  "data": [
    {
      "id": 0,
      "serialNumber": "string",
      "logicalId": "string",
      "customerId": 0,
      "modelId": 0,
      "modelName": "string",
```

Click *Try it out* button to get the device details.

Error Codes

HTTP Status Code	Application Code	Name	Severity	Message
200	S100	S_OK	-	Success.
401	E101	E_TOKEN_INVALID_OR_EXPIRED	2	Token is expired or invalid.
401	E102	E_UNAUTHORIZED_ACCESS	2	The user does not have the right to operate.
500	E103	E_GENERAL_ERROR	3	API failed.

GET DEVICES

This API is used to get the list of all devices except deleted and blacklisted devices under the root hierarchy that are accessible to the logged-in user. Pagination is supported for this API.

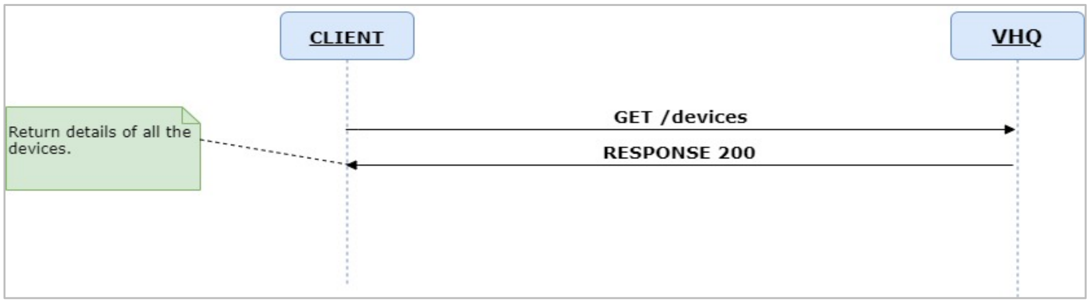
Get Devices API

To get the list of all device details such as: Device ID, Serial Number, Model ID, Model Name, Status, Sub-status, IP Address, MAC Address, and Location ID.

Go to the following path:

```
API - GET/devices

URL - https://<API Server>/apis/v1.3/devices
e.g.  https://vhqtest.verifone.com/apis/v1.3/devices
```

Sequence Diagram



Input Parameters

Input parameters must be sent in the query string.

Element	Optional/Mandatory	Data Type	Description
serialNumber	Optional	String	The serial number of the device.
modelId	Optional	int	The model ID of the device.
deviceId	Optional	string	Device Identification.
hierarchyId/hierarchyname	Optional	int	Hierarchy Id.
groupId/groupName	optional	int	Group Id.
applications	Optional	String	The array of applications. Value format with Application = {Application Name}: {Version}
reportedApplications	Optional	String	The array of reported applications. Value format with Application = {Application Name}: {Version}.
packageName	Optional	String	The array of packages.
limit	Optional	Int	Rows per request.
offset	Optional	Int	Offset required to calculate the page number.

Output Parameters

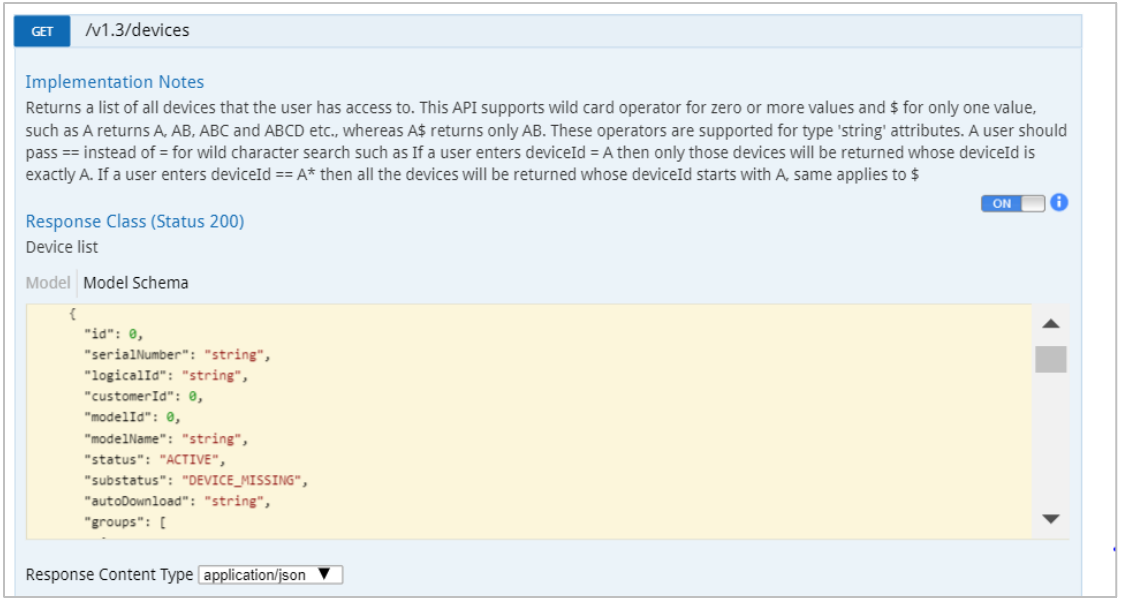
Element	Attribute	Data Type
devices (collection of devices)	uniqueDeviceId	int
	serialNumber	String
	deviceId	String
	status	String
	model	String
	hierarchyFullPath	String

Get Devices API - Sample Code

SAMPLE CODE SNIPPET

```
public class GetDevices {

   private static String url = "http://<servername:port>/apis/v1.3/";
   private int statusCode;
   // This is the base URL, it will be appended as per the entity which is invoked

   /**
    * @param completeURL -> This is the URL which appended as per the
   entity which is invoked
    * @param method -> This refer to which method we are going to
   send e.g. get,post,put
    * @return -> httpCon with Content Type and Authorization token
    * @throws IOException
    */
   HttpURLConnection getConnection(URL completeURL, String method)
         throws IOException {
      HttpURLConnection httpCon = (HttpURLConnection)
            completeURL.openConnection();
      httpCon.setDoOutput(true);
      httpCon.setRequestProperty("Authorization", "Bearer 29797eae-3879-3827-8ec3-
5295796c5af1");
      httpCon.setRequestMethod(method);
      httpCon.setRequestProperty("Content-Type", "application/json");
      return httpCon;
   }
   /**
    * This method get all data of a entity
    * @param API -> This refers to the endpoint of the URL for the
   particular entity e.g. for Device entity it is devices
    * @param id -> This refer to show specific data from Entity based
   on Id.
    * @param fields -> This refers to show specific fields, which we
   pass as a parameter.
    * @param queryparameter
    * @return response from entity
    */
   String get(String Api, int id,String fullPath,String[] queryparameter,String
queryParamValue) {
      StringBuffer response=null;
      if(id!=0) {
         url = url.concat(Api) + "/" + id;
      }else {
         url = url.concat(Api);
      }

      if(fullPath!=null){
         url=url+"/".concat(fullPath);
      }
      if(null!=queryparameter){
         for(String fields: queryparameter){
            url = url +"?"+fields+"="+queryParamValue;


            System.out.println("URL :::: "+url);
         }
      }
      try {
         URL completeURL = new URL(url);
         // Getting All Header Information
         // like(contentType,CustomerName,CustomerId)
         HttpURLConnection httpCon = getConnection(completeURL, "GET");
         httpCon.getResponseCode();
         BufferedReader in = new BufferedReader(new
InputStreamReader(httpCon.getInputStream()));
         String inputLine;
         response =new StringBuffer();
         while ((inputLine = in.readLine()) != null) {
            response.append(inputLine);
         }
         System.out.println(response);
         System.out.println(httpCon.getResponseCode());
         System.out.println(httpCon.getResponseMessage());
      } catch (Exception ex) {
         ex.printStackTrace();
         if (ex.getMessage().equalsIgnoreCase("404 Not Found")) {
            statusCode = 404;
            throw new RuntimeException("Not Found " + statusCode);

         } else if (ex.getMessage().equalsIgnoreCase("400 Bad Request")) {
            statusCode = 400;
            throw new RuntimeException("Bad Request " + statusCode);
         } else if (ex.getMessage().equalsIgnoreCase("409 Conflict"))
         {
            statusCode = 409;
            throw new RuntimeException("Already Exist : Conflict " +
                  statusCode);
         } else {
            statusCode = 500;
            throw new RuntimeException("Already Exist : Conflict " +
                  statusCode);
         }
      }
      return response.toString();
   }
   public static void main(String[] args) {
      String [] queryparameter= {"customerId"};
      String queryParamValue = "11482";
      new GetDevices().get("devices",0,null,queryparameter,queryParamValue);
   }
}
```

Sample using Swagger - Request



/v1.3/devices

Implementation Notes

Returns a list of all devices that the user has access to. This API supports wild card operator for zero or more values and $ for only one value, such as A returns A, AB, ABC and ABCD etc., whereas A$ returns only AB. These operators are supported for type 'string' attributes. A user should pass == instead of = for wild character search such as If a user enters deviceId = A then only those devices will be returned whose deviceId is exactly A. If a user enters deviceId == A* then all the devices will be returned whose deviceId starts with A, same applies to $

Response Class (Status 200)

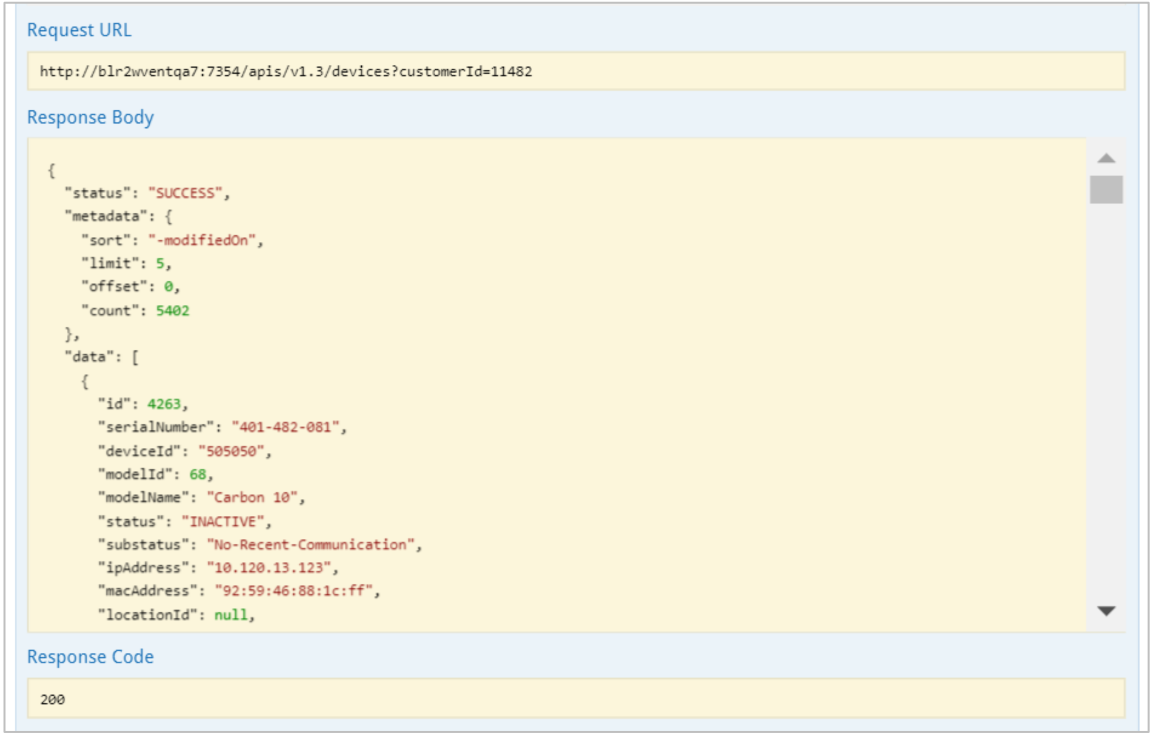
Device list

```
{
  "id": 0,
  "serialNumber": "string",
  "logicalId": "string",
  "customerId": 0,
  "modelId": 0,
  "modelName": "string",
  "status": "ACTIVE",
  "substatus": "DEVICE_MISSING",
  "autoDownload": "string",
  "groups": [
```

Response Content Type application/json ▼

Click *Try it out* button to get the device details.

Sample using Swagger - Response



Request URL

```
http://blr2wventqa7:7354/apis/v1.3/devices?customerId=11482
```

Response Body

```
{
  "status": "SUCCESS",
  "metadata": {
    "sort": "-modifiedOn",
    "limit": 5,
    "offset": 0,
    "count": 5402
  },
  "data": [
    {
      "id": 4263,
      "serialNumber": "401-482-081",
      "deviceId": "505050",
      "modelId": 68,
      "modelName": "Carbon 10",
      "status": "INACTIVE",
      "substatus": "No-Recent-Communication",
      "ipAddress": "10.120.13.123",
      "macAddress": "92:59:46:88:1c:ff",
      "locationId": null,
```

Response Code

```
200
```

Error Codes

HTTP Status Code	Application Code	Name	Severity	Message
200	S100	S_OK	-	Success.
401	E101	E_TOKEN_INVALID_OR_EXPIRED	2	Token is expired or invalid.
401	E102	E_UNAUTHORIZED_ACCESS	2	The user does not have the right to operate.
500	E103	E_GENERAL_ERROR	3	API failed.

Get Devices by DeviceUid API

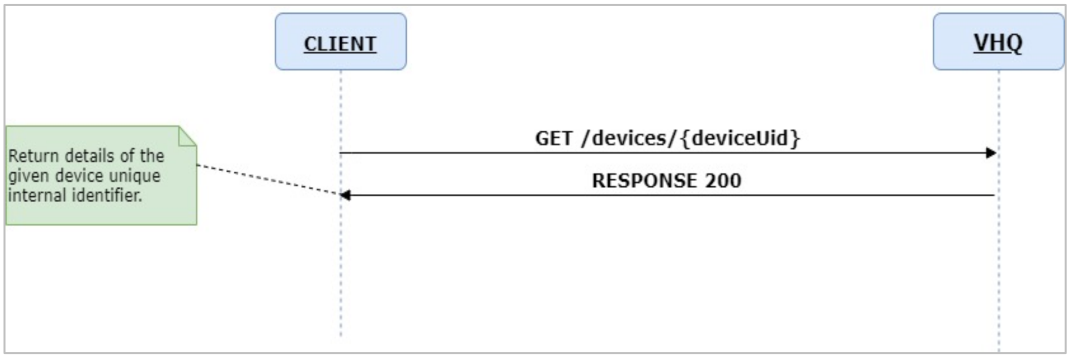
This API is used to get the device details of a single device. The following device details such as Device ID, Serial Number, Model ID, Model Name, Status, Sub-status, IP Address, MAC Address, Location ID, etc., are retrieved.

Go to the following path:

```
API - GET/devices/{deviceUid}
```

```
URL - https://<API Server>/apis/v1.3/devices/{deviceUid}
e.g.  https://vhqtest.verifone.com/apis/v1.3/devices/{deviceUid}
```

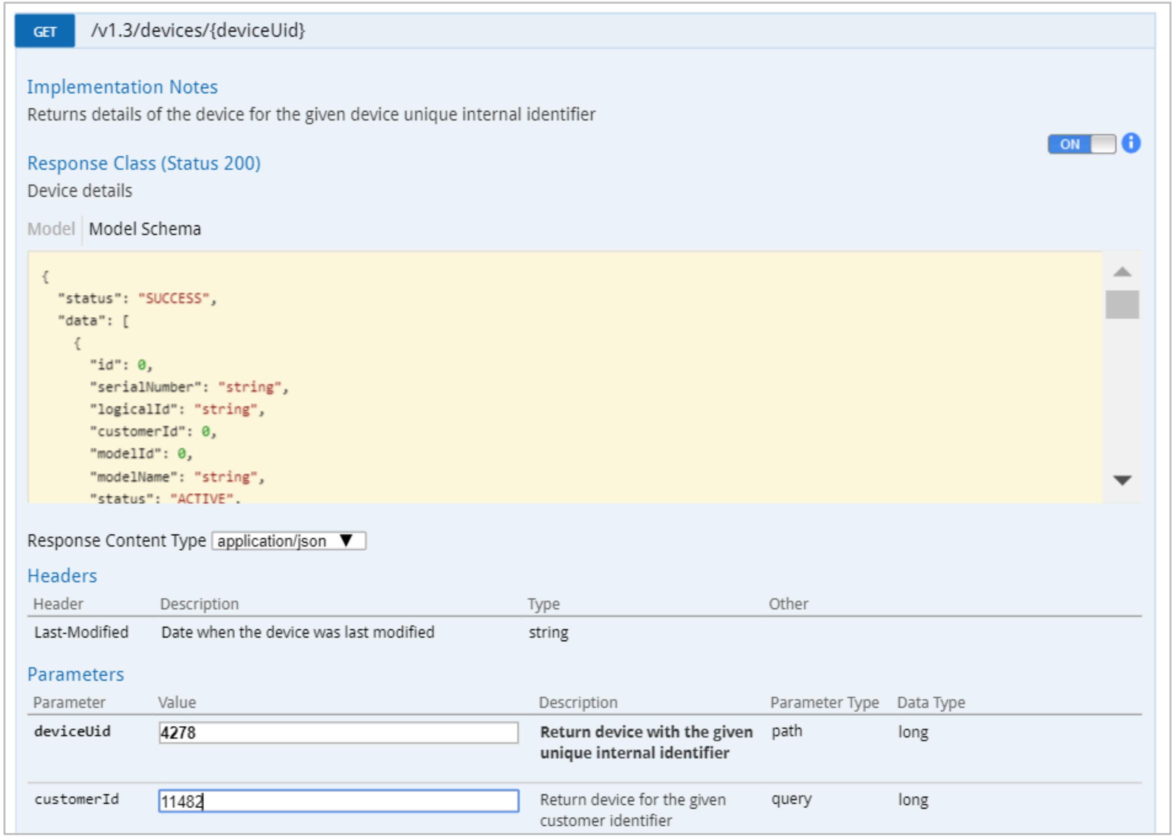
Sequence Diagram


CLIENT
VHQ
GET /devices/{deviceUid}
RESPONSE 200
Return details of the given device unique internal identifier.

Get Device by DeviceUid API - Sample Code

NOTE	The sample code for both getDeviceById() and getAllDevices() are same. But the deviceUid parameter is passed to retrieve the details of a particular device for getDeviceById().

Sample using Swagger – Request



/v1.3/devices/{deviceUid}

Implementation Notes

Returns details of the device for the given device unique internal identifier

Response Class (Status 200)

Device details

Model | Model Schema

```
{
  "status": "SUCCESS",
  "data": [
    {
      "id": 0,
      "serialNumber": "string",
      "logicalId": "string",
      "customerId": 0,
      "modelId": 0,
      "modelName": "string",
      "status": "ACTIVE",
```

Response Content Type: application/json

Headers

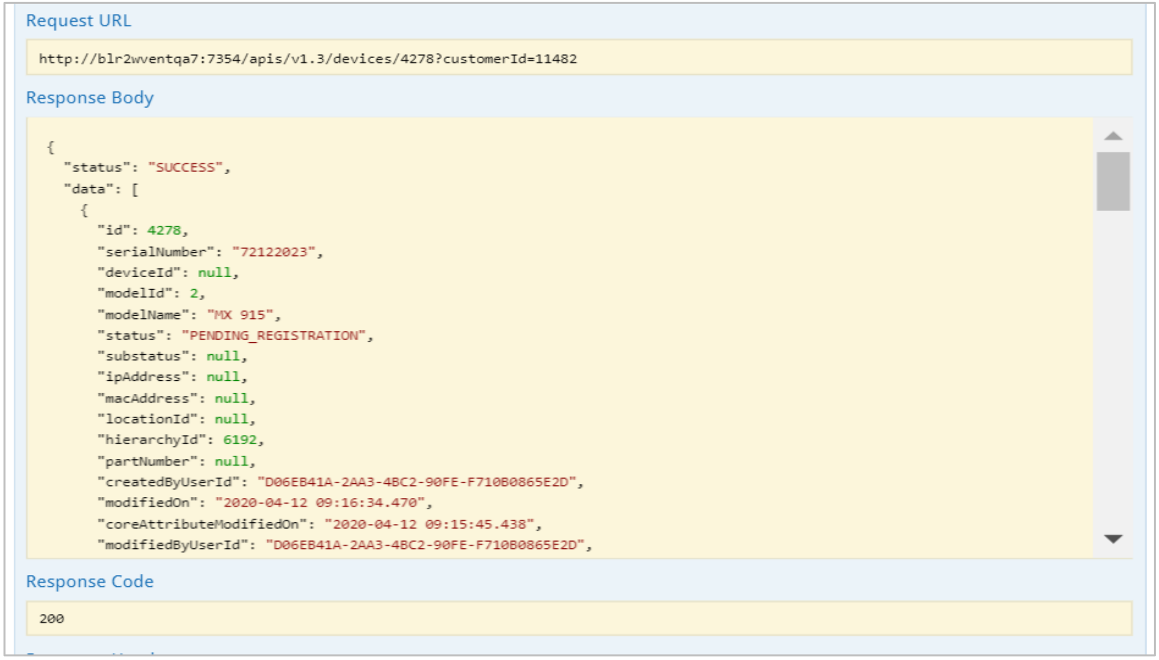
Header	Description	Type	Other
Last-Modified	Date when the device was last modified	string	

Parameters

Parameter	Value	Description	Parameter Type	Data Type
deviceUid	4278	**Return device with the given unique internal identifier**	path	long
customerId	11482	Return device for the given customer identifier	query	long

Click *Try it out* button to get the device details.

Sample using Swagger – Response



Request URL

```
http://blr2wventqa7:7354/apis/v1.3/devices/4278?customerId=11482
```

Response Body

```
{
  "status": "SUCCESS",
  "data": [
    {
      "id": 4278,
      "serialNumber": "72122023",
      "deviceId": null,
      "modelId": 2,
      "modelName": "MX 915",
      "status": "PENDING_REGISTRATION",
      "substatus": null,
      "ipAddress": null,
      "macAddress": null,
      "locationId": null,
      "hierarchyId": 6192,
      "partNumber": null,
      "createdByUserId": "D06EB41A-2AA3-4BC2-90FE-F710B0865E2D",
      "modifiedOn": "2020-04-12 09:16:34.470",
      "coreAttributeModifiedOn": "2020-04-12 09:15:45.438",
      "modifiedByUserId": "D06EB41A-2AA3-4BC2-90FE-F710B0865E2D",
```

Response Code

```
200
```

DELETE DEVICE

This API is used to delete a single device based on the unique device ID.

Pre-requisites
Get the device ID to be deleted.

Go to the following path:

```
API - DELETE/devices/{deviceUid}
URL - https://<API Server>/apis/v1.3/devices/{deviceUid}
e.g. https://vhqtest.verifone.com/apis/v1.3/devices/{deviceUid}
```

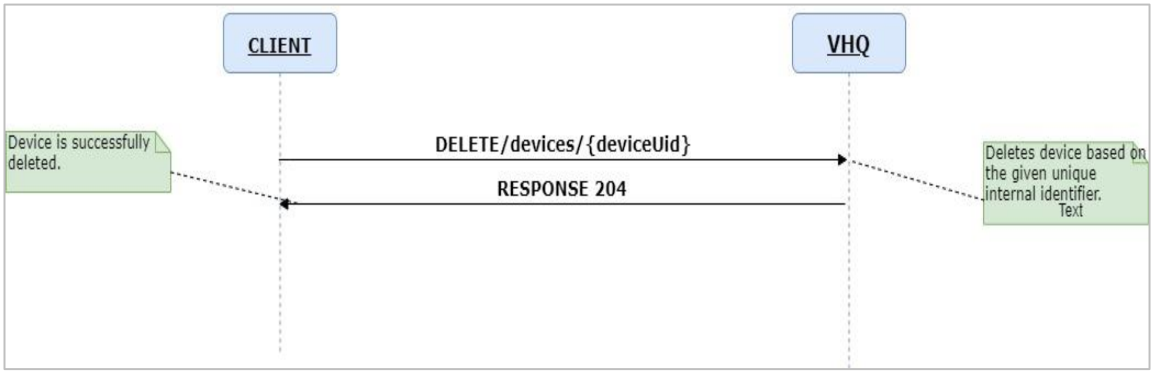
Soft Delete

Soft Delete - does not permanently delete the device from the database. The device can be deleted using **DeviceUid**. The status of the device is seen as 'DELETED' when you try to retrieve the device.

Hard Delete

Hard Delete - to permanently delete a device, the value is given as “true” for the key parameter “purge”.

Sequence Diagram


CLIENT
VHQ
Device is successfully deleted.
DELETE/devices/{deviceUid}
RESPONSE 204
Deletes device based on the given unique internal identifier.
Text

Input Parameters

Input parameters must be part of the query string.

Element	Optional/Mandatory	Data Type	Description
Udi	Mandatory	Int	Unique device identifier to be set as the path of the URI.

Output Parameters

None

Delete Device API - Sample Code for Soft Delete

SAMPLE CODE SNIPPET

```
public class DeleteDeviceSoftSample {
   private static String url = "http://<servername:port>/apis/v1.3/";
   // This is the base URL, it will be appended as per the entity which is invoked
   /**
    * @param Api -> This refers to the endpoint of the URL for the particular entity e.g. for device
entity it is devices
    * @return
    */
```

```
HttpURLConnection getConnection(URL completeURL, String method) throws IOException {
   HttpURLConnection httpCon = (HttpURLConnection)completeURL.openConnection();
   httpCon.setDoOutput(true);
   httpCon.setRequestProperty("Authorization", "Bearer dffc9915-cb1f-31d4-9562-bcb00862adfa");
   httpCon.setRequestMethod(method);
   httpCon.setRequestProperty("Content-Type", "application/json");
   return httpCon;
}
/**
 * @param Api -> This refers to the endpoint of the URL for the
particular entity e.g. for Device entity it is devices
 * @param body -> The request body which is required to be patched,
This changes as per the entity
 * @param id -> This refer to which specific Row need to change
 */
void delete(String api, String body, int id,String[] queryparameter,String queryParamValue) {
   StringBuffer response = null;
   //String BASE_PATH = url.concat(api) + "/" + id;
   if(id!=0) {
      url = url.concat(api) + "/" + id;
   }else {
      url = url.concat(api);
   }

   if(body!=null){
      url=url+"/".concat(body);
   }
   if(null!=queryparameter){
      for(String fields: queryparameter){
         url = url +"?"+fields+"="+queryParamValue;
         System.out.println("URL :::: "+url);
      }
   }
   //Getting All Header Information
   //like(contentType,CustomerName,CustomerId)

   try {

      System.out.println("url ::: "+url);
      URL completeURL = new URL(url);
      // Getting All Header Information
      // like(contentType,CustomerName,CustomerId)
      HttpURLConnection httpCon = getConnection(completeURL,"DELETE");
      httpCon.getResponseCode();
      BufferedReader in = new BufferedReader(new InputStreamReader(httpCon.getInputStream()));
      String inputLine;
      response = new StringBuffer();
      while ((inputLine = in.readLine()) != null) {
         response.append(inputLine);
      }
      System.out.println("Response ::: "+response);
      System.out.println(httpCon.getResponseCode());
      System.out.println(httpCon.getResponseMessage());
   } catch (Exception ex) {
      int statusCode;
      ex.printStackTrace();
      if (ex.getMessage().equalsIgnoreCase("404 Not Found")) {
         statusCode = 404;
         throw new RuntimeException("Not Found " + statusCode);
      } else if (ex.getMessage().equalsIgnoreCase("400 Bad Request")) {
         statusCode = 400;
         throw new RuntimeException("Bad Request " + statusCode);
      } else if (ex.getMessage().equalsIgnoreCase("409 Conflict"))
      {
         statusCode = 409;
         throw new RuntimeException("Already Exist : Conflict " +
               statusCode);
      } else {
         statusCode = 500;
```

```
            throw new RuntimeException("Already Exist : Conflict " +
                  statusCode);
         }
      }
      System.out.println("Device remove success !!");
   }
   public static void main(String[] args) {
      String [] queryparameter= {"customerId"};
      String queryParamValue = "11482";
      new DeleteDeviceSoftSample().delete("devices", null, 78507, queryparameter, queryParamValue);
   }
}
```

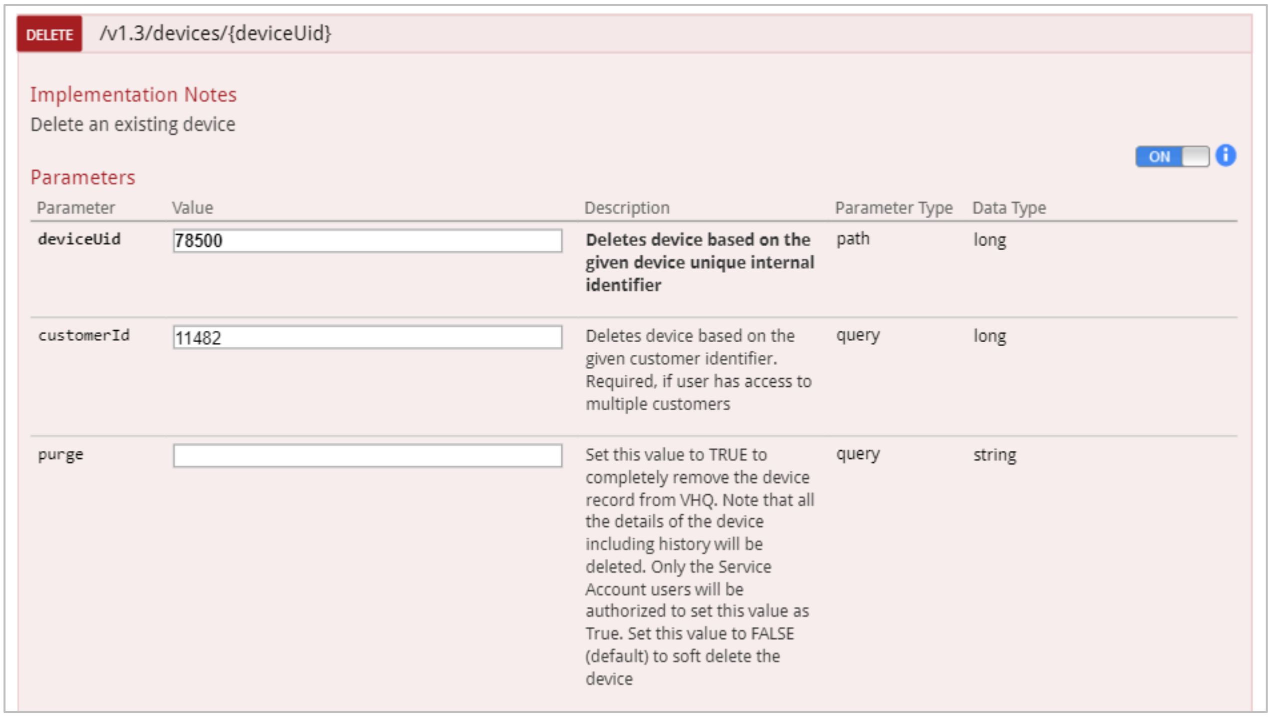
Delete Device API - Sample Code for Hard Delete

SAMPLE CODE SNIPPET

```
public class DeleteDeviceHardSample {
   private static String url = "http://<servername:port>/apis/v1.3/";
   // This is the base URL, it will be appended as per the entity which is invoked
   /**
    * @param Api -> This refers to the endpoint of the URL for the particular entity e.g. for device
entity it is devices
    * @return
    */
   HttpURLConnection getConnection(URL completeURL, String method) throws IOException {
      HttpURLConnection httpCon = (HttpURLConnection)completeURL.openConnection();
      httpCon.setDoOutput(true);
      httpCon.setRequestProperty("Authorization", "Bearer dffc9915-cb1f-31d4-9562-bcb00862adfa");
      httpCon.setRequestMethod(method);
      httpCon.setRequestProperty("Content-Type", "application/json");
      return httpCon;
   }
   /**
    * @param Api -> This refers to the endpoint of the URL for the
   particular entity e.g. for Device entity it is devices
    * @param body -> The request body which is required to be patched,
   This changes as per the entity
    * @param id -> This refer to which specific Row need to change
    */
   void delete(String api, String body, int id,String[] queryparameter,String[] queryParamValue) {
      StringBuffer response = null;
      //String BASE_PATH = url.concat(api) + "/" + id;
      if(id!=0) {
         url = url.concat(api) + "/" + id;
      }else {
         url = url.concat(api);
      }
      if(body!=null){
         url=url+"/".concat(body);
      }
      url=url+ "?";
      if(null!=queryparameter && null!=queryParamValue){
         int len = queryparameter.length;
         int value=0;
         while(value!=len){
            url = url  + queryparameter[value];
            url = url + "=" + queryParamValue[value] ;
            value++;
            if(value!=len)
            {
               url=url+"&";
            }
         }
      }
      try {
         System.out.println("url ::: "+url);
         URL completeURL = new URL(url);
         // Getting All Header Information
         // like(contentType,CustomerName,CustomerId)
         HttpURLConnection httpCon = getConnection(completeURL,"DELETE");
         httpCon.getResponseCode();
         BufferedReader in = new BufferedReader(new InputStreamReader(httpCon.getInputStream()));
         String inputLine;
         response = new StringBuffer();
         while ((inputLine = in.readLine()) != null) {
            response.append(inputLine);
         }
         System.out.println("Response ::: "+response);
         System.out.println(httpCon.getResponseCode());
         System.out.println(httpCon.getResponseMessage());
      } catch (Exception ex) {
         int statusCode;
         ex.printStackTrace();
         if (ex.getMessage().equalsIgnoreCase("404 Not Found")) {
            statusCode = 404;
            throw new RuntimeException("Not Found " + statusCode);
         } else if (ex.getMessage().equalsIgnoreCase("400 Bad Request")) {
            statusCode = 400;
            throw new RuntimeException("Bad Request " + statusCode);
         } else if (ex.getMessage().equalsIgnoreCase("409 Conflict"))
         {
            statusCode = 409;
            throw new RuntimeException("Already Exist : Conflict " +
                  statusCode);
         } else {
            statusCode = 500;
            throw new RuntimeException("Already Exist : Conflict " +
                  statusCode);
         }
      }
      System.out.println("Device remove success !!");
   }
   public static void main(String[] args) {
      String [] queryparameter= {"customerId","purge"};
      String [] queryParamValue = {"11482","true"};
      new DeleteDeviceHardSample().delete("devices", null, 78508, queryparameter, queryParamValue);
   }
}
```

Sample using Swagger – Soft Delete Request

The device is deleted for the specified UID.



DELETE /v1.3/devices/{deviceUid}

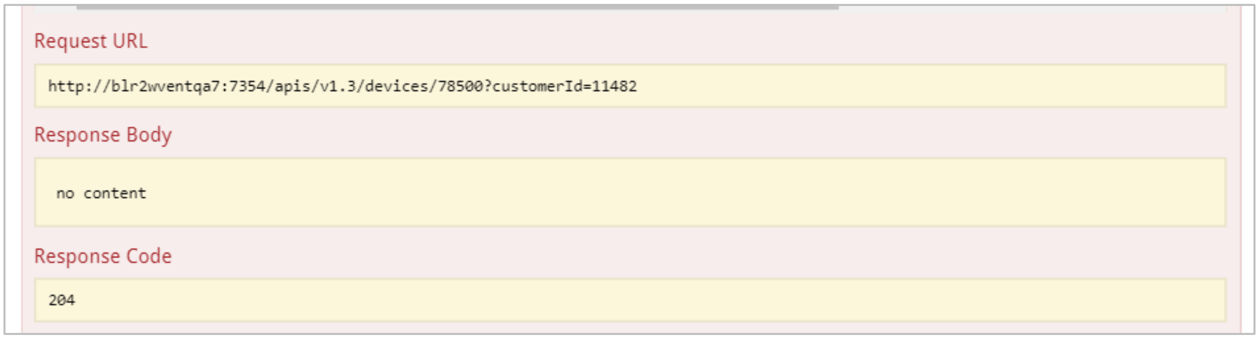
Implementation Notes

Delete an existing device

Parameters

Parameter	Value	Description	Parameter Type	Data Type
deviceUid	78500	**Deletes device based on the given device unique internal identifier**	path	long
customerId	11482	Deletes device based on the given customer identifier. Required, if user has access to multiple customers	query	long
purge		Set this value to TRUE to completely remove the device record from VHQ. Note that all the details of the device including history will be deleted. Only the Service Account users will be authorized to set this value as True. Set this value to FALSE (default) to soft delete the device	query	string

Sample using Swagger – Soft Delete Response



Request URL

```
http://blr2wventqa7:7354/apis/v1.3/devices/78500?customerId=11482
```

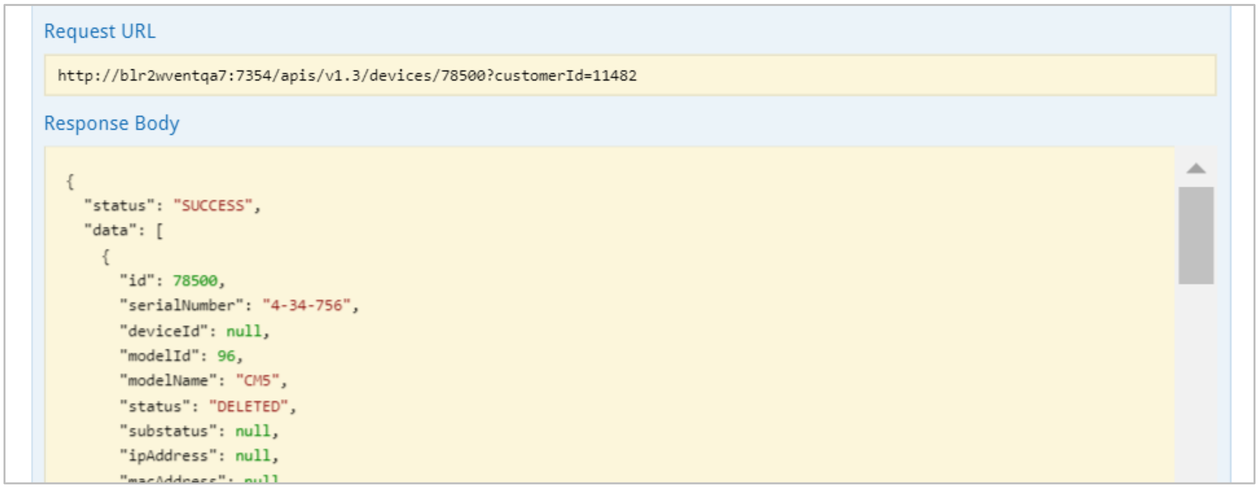
Response Body

```
no content
```

Response Code

```
204
```

The status is seen as DELETED while trying to retrieve the device.



Request URL

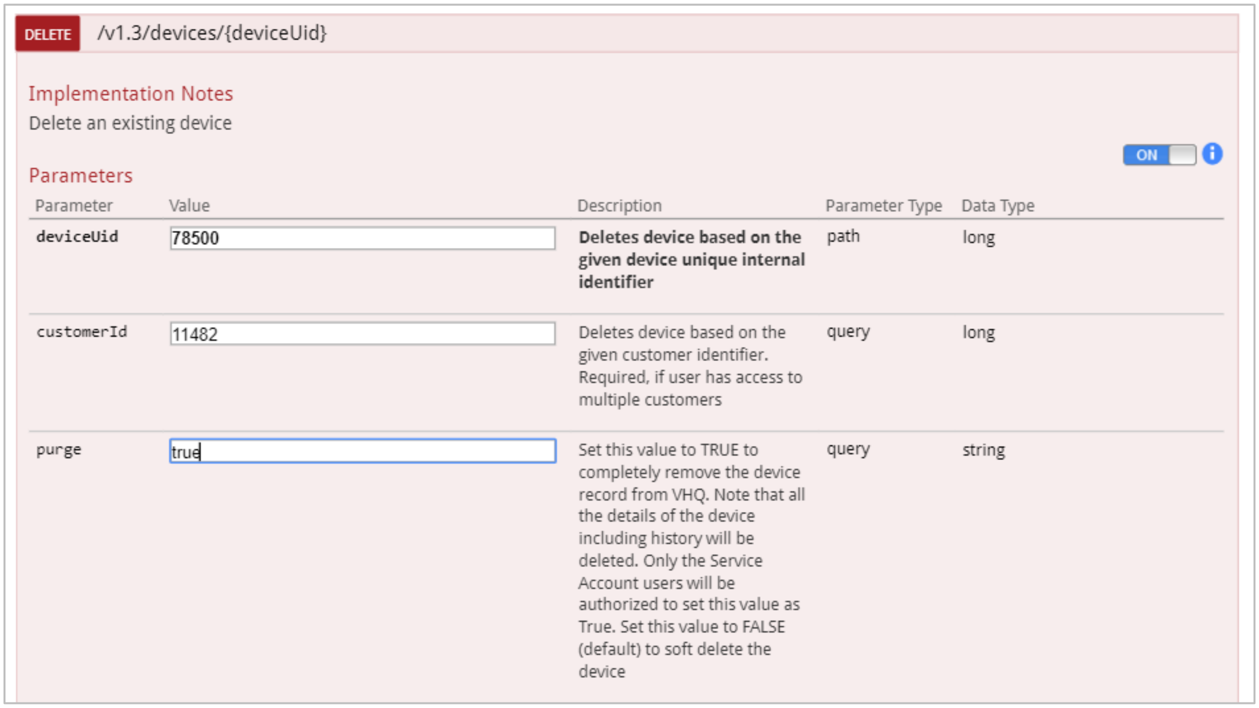
```
http://blr2wventqa7:7354/apis/v1.3/devices/78500?customerId=11482
```

Response Body

```
{
  "status": "SUCCESS",
  "data": [
    {
      "id": 78500,
      "serialNumber": "4-34-756",
      "deviceId": null,
      "modelId": 96,
      "modelName": "CM5",
      "status": "DELETED",
      "substatus": null,
      "ipAddress": null,
      "macAddress": null
```

Sample using Swagger – Hard Delete Request

To delete the device, the value is given as "true" for the key parameter "purge.”



/v1.3/devices/{deviceUid}

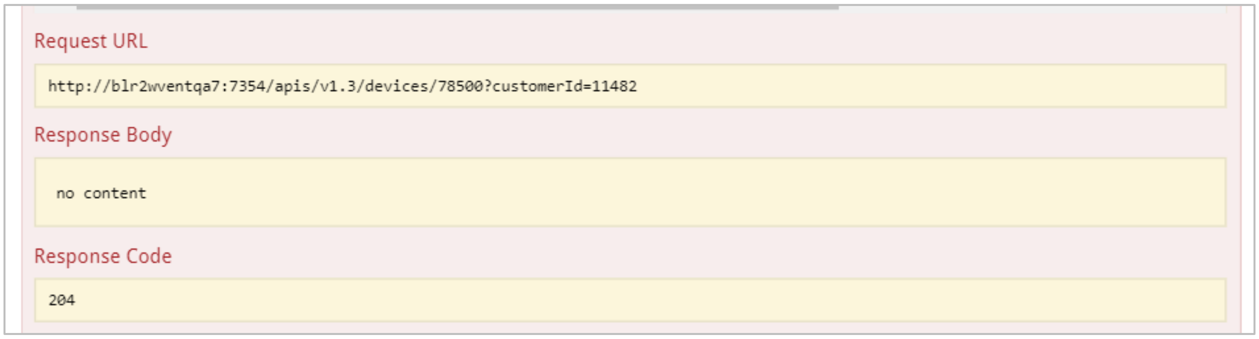
Implementation Notes

Delete an existing device

Parameters

Parameter	Value	Description	Parameter Type	Data Type
deviceUid	78500	**Deletes device based on the given device unique internal identifier**	path	long
customerId	11482	Deletes device based on the given customer identifier. Required, if user has access to multiple customers	query	long
purge	true	Set this value to TRUE to completely remove the device record from VHQ. Note that all the details of the device including history will be deleted. Only the Service Account users will be authorized to set this value as True. Set this value to FALSE (default) to soft delete the device	query	string

Sample using Swagger – Hard Delete Response



Request URL

```
http://blr2wventqa7:7354/apis/v1.3/devices/78500?customerId=11482
```

Response Body

```
no content
```

Response Code

```
204
```

Error Codes

HTTP Status Code	Application Code	Name	Severity	Message
204	S204	S_DELETED	-	Success.
401	E101	E_TOKEN_INVALID_OR_EXPIRED	2	Token expired or invalid.
401	E102	E_UNAUTHORIZED_ACCESS	2	The user does not have the right to perform the operation.
500	E103	E_GENERAL_ERROR	3	API failed.
500	E114	EX_UPDATEDEVICESTATUS_FAILED	3	Device Status failed.
500	E126	E_DEVICE_ALREADY_DELETED_OR_BLACKLISTED	3	The device is already in the Deleted/Blacklist status.

PARAMETER MANAGEMENT APIs

Applications on the device comprise a set of executables and configuration files. Configuration files store the application parameters.

The Parameter Management feature makes it easier for the user to configure, edit, download, and manage the values of these application parameters. The user defines the configuration values and downloads the application (including OS) parameters onto the device.

CREATE PARAMETER

This API is used to create application parameters for a device.

Request Body

```
{
  "data": [
    {
      "deviceUid": 0,
      "customerId": 0,
      "applicationId": 0,

      "applicationName": string",

      "applicationVersion": string",

      "deviceParameters": [

        {

          "name": "string",

          "value": "string",

        }

      ]

    }

  ]
}
```

Pre-requisites

Following are the pre-requisites for creating a new parameter:

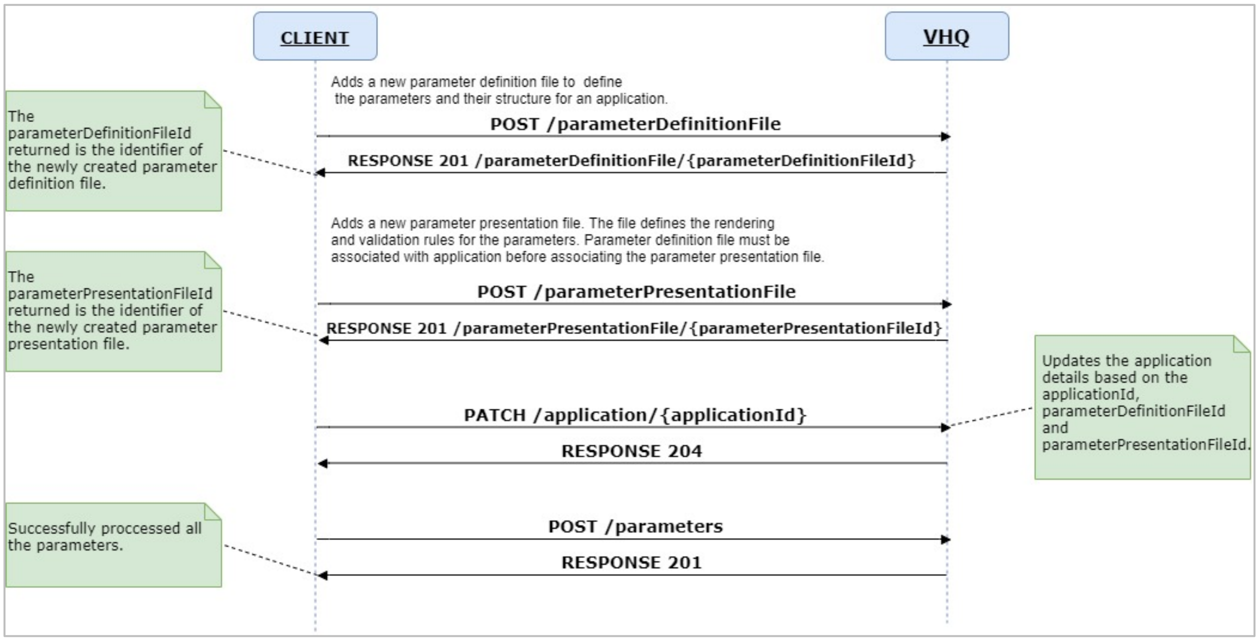
The user can use either name of the entities or IDs while updating the device.

- Hierarchy
  Note: hierarchyFullPath value should be given in hierarchyName field of the request body.
- Groups
- Reference Set

Once you have the parent hierarchy details, groups, and reference set, go to the following path:

```
API - POST/parameters
URL - https://<API Server>/apis/v1.3/parameters
e.g. https://vhqtest.verifone.com/apis/v1.3/parameters
```

Sequence Diagram


CLIENT
VHQ
POST /parameterDefinitionFile
RESPONSE 201 /parameterDefinitionFile/{parameterDefinitionFileId}
POST /parameterPresentationFile
RESPONSE 201 /parameterPresentationFile/{parameterPresentationFileId}
PATCH /application/{applicationId}
RESPONSE 204
POST /parameters
RESPONSE 201

Adds a new parameter definition file to define the parameters and their structure for an application.

The parameterDefinitionFileId returned is the identifier of the newly created parameter definition file.

Adds a new parameter presentation file. The file defines the rendering and validation rules for the parameters. Parameter definition file must be associated with application before associating the parameter presentation file.

The parameterPresentationFileId returned is the identifier of the newly created parameter presentation file.

Updates the application details based on the applicationId, parameterDefinitionFileId and parameterPresentationFileId.

Successfully proccessed all the parameters.

Input Parameters

Element	attribute	Optional/ Mandatory	Data Type	Description
deviceUid		Mandatory	Int	Unique device identifier to be set as the path of the URI.
applicationId / applicationName / applicationName and applicationVersion		Mandatory	Int	Application Identifier.
parameters (collection of parameters)	id	Mandatory	int	Parameter identifier.
	value	Mandatory	string	Parameter value.

Output Parameters

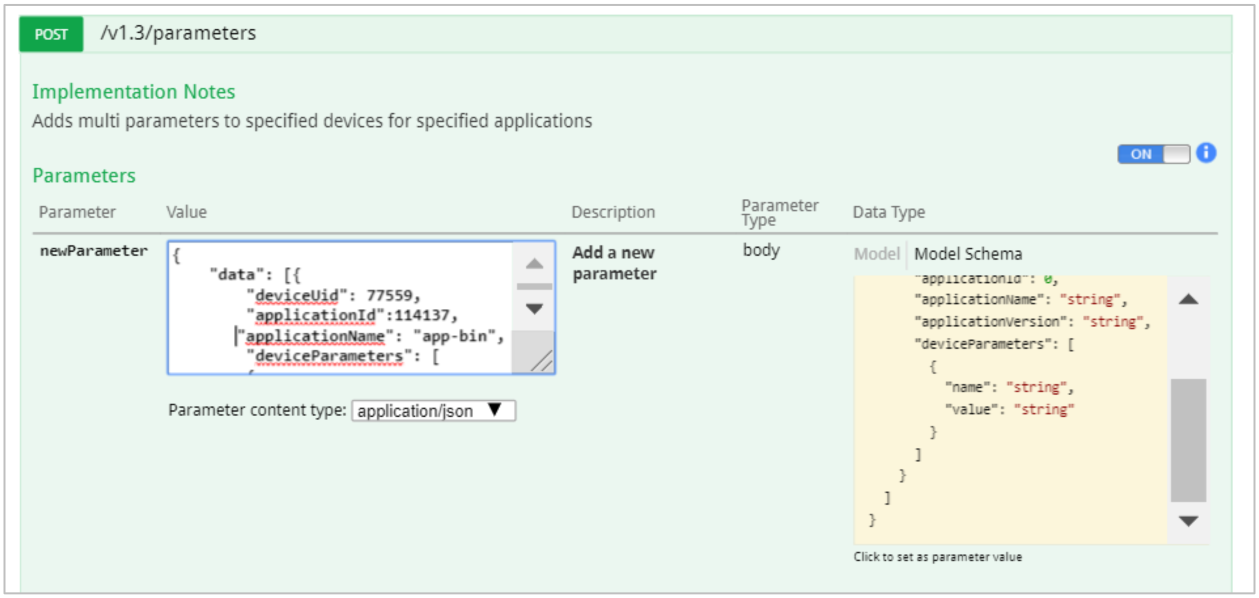
None

Create Parameter API - Sample Code

SAMPLE CODE SNIPPET

```
public class CreateParameters {
   //Common API URL
   private  static String url = "http://<servername:port>/apis/v1.3/";
   // This is the base URL, it will be appended as per the entity which is invoked
   // Request body
   private static String postparameter = "{\r\n" +
         "  \"data\": [{\r\n" +
         "     \"deviceUid\": 77559,\r\n" +
         "     \"applicationId\":114137,\r\n" +
         "        \"deviceParameters\": [\r\n" +
         "        {\r\n" +
         "          \"name\": \"Merchant.*/id\",\r\n" +
         "          \"value\": 3,\r\n" +
         "          \"key\": true\r\n" +
         "        },\r\n" +
         "        {\r\n" +
         "          \"name\": \"Merchant.*/merchant_id\",\r\n" +
         "          \"value\": 4,\r\n" +
         "          \"key\": true\r\n" +
         "        },\r\n" +
         "        {\r\n" +
         "          \"name\": \"Merchant.*/name\",\r\n" +
         "          \"value\": 5\r\n" +
         "          ,\r\n" +
         "          \"key\": true\r\n" +
         "        }\r\n" +
         "      ],\r\n" +
         "     \"customerId\": 11482\r\n" +
         "  }]\r\n" +
         "}"; // Post Request body in JSON format

   /**
    * @param completeURL -> This is the URL which appended as per the
   entity which is invoked
    * @param method -> This refer to which method we are going to
   send e.g. get,post,put
    * @return -> httpCon with Content Type and Authorization token
    * @throws IOException
    */
   HttpURLConnection getConnection(URL completeURL, String method) throws IOException {
      HttpURLConnection httpCon = (HttpURLConnection)completeURL.openConnection();
      httpCon.setDoOutput(true);
      httpCon.setRequestProperty("Authorization", "Bearer 68ed3f3b-7f64-33c7-9d18-40076046a47d");
      httpCon.setRequestMethod(method);
      httpCon.setRequestProperty("Content-Type", "application/json");
      return httpCon;
   }
   /**
    * This method creates a new entity
    * @param body -> The request body which is required to be posted,
   This changes as per the entity
    * @param Api -> This refers to the endpoint of the URL for the
   particular entity e.g. for Parameter entity it is parameter
    */
   String post(String body, String apiEndPoint, String [] queryparameter, String queryParamValue) {
      url = url.concat(apiEndPoint);
      String id = "";
      if(null!=queryparameter){
         for(String fields: queryparameter){
            url = url +"?"+fields+"="+queryParamValue;
            System.out.println("URL :::: "+url);
         }
      }
      try {
         System.out.println("Url :::::::: " +url);
         URL completeURL = new URL(url);
         // Getting All Header Information
         // like(contentType,CustomerName,CustomerId)
         HttpURLConnection httpCon = getConnection(completeURL,
               "POST");
         OutputStream os = httpCon.getOutputStream();
         os.write(postparameter.getBytes());
         os.flush();
         if (httpCon.getResponseCode() != HttpURLConnection.HTTP_CREATED) {
            if (httpCon.getResponseCode() == 409) {
               throw new RuntimeException(apiEndPoint + "Already Exist : Conflict " + httpCon.getResponseCode());
            }
         }
         for(java.util.Map.Entry<String, List<String>> entry: httpCon.getHeaderFields().entrySet()){
            if(null !=entry.getKey()
                  &&entry.getKey().toLowerCase().contains("new")){
               id = entry.getValue().get(0);
               id = id.substring(id.lastIndexOf("/")+1);
            }
         }
         System.out.println(id);
         System.out.println(httpCon.getHeaderFields());
         System.out.println("Response Code for POST method->"+httpCon.getResponseCode());
         System.out.println(httpCon.getResponseMessage());
      } catch (Exception ex) {
         int statusCode ;
         ex.printStackTrace();
         if (ex.getMessage().equalsIgnoreCase("404 Not Found")) {
            statusCode = 404;
            throw new RuntimeException("Not Found " + statusCode);
         }
         else if (ex.getMessage().equalsIgnoreCase("400 Bad Request")) {
            statusCode = 400;
            throw new RuntimeException("Bad Request " + statusCode);
         }
         else if (ex.getMessage().equalsIgnoreCase("409 Conflict"))
         {
            statusCode = 409;
            throw new RuntimeException("Already Exist : Conflict " +
                  statusCode);
         } else {
            statusCode = 500;
            throw new RuntimeException("Already Exist : Conflict " +
                  statusCode);
         }
      }
      //System.out.println("Added Successfully");
      return id;
   }
   public static void main(String[] args) {
      //Post method will accept two parameter, Request Body and API
      System.out.println(":::::::: Create Parameters calling start ::::::::");
      String [] queryparameter= {"customerId"};
      String queryParamValue = "11482";
      CreateParameters = new CreateParameters();
      createParameters.post(postparameter, "parameters",queryparameter,queryParamValue);
      System.out.println(":::::::: Create Parameters calling End ::::::::");
   }
}
```

Sample using Swagger – Request



/v1.3/parameters

Implementation Notes

Adds multi parameters to specified devices for specified applications

Parameters

Parameter	Value	Description	Parameter Type	Data Type
newParameter		**Add a new parameter**	body	Model Model Schema

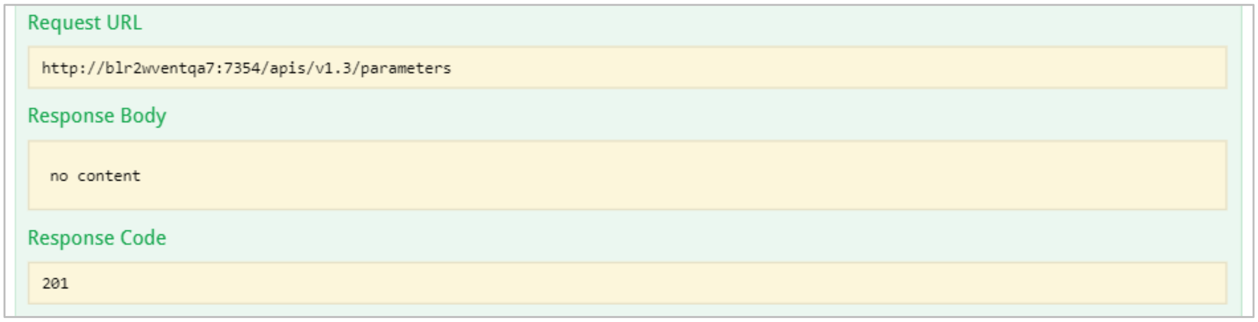
```
{
    "data": [{
        "deviceUid": 77559,
        "applicationId":114137,
       "applicationName": "app-bin",
        "deviceParameters": [
```

Parameter content type: application/json

```
      "applicationId": 0,
      "applicationName": "string",
      "applicationVersion": "string",
      "deviceParameters": [
        {
          "name": "string",
          "value": "string"
        }
      ]
    }
  ]
}
```

Click to set as parameter value

Sample using Swagger – Response



Request URL

```
http://blr2wventqa7:7354/apis/v1.3/parameters
```

Response Body

```
no content
```

Response Code

```
201
```

Error Codes

HTTP Status Code	Application Code	Name	Severity	Message
200	S200	S_SUCCESS	-	Success.
401	E101	E_TOKEN_INVALID_OR_EXPIRED	2	Token expired or invalid.
401	E102	E_UNAUTHORIZED_ACCESS	2	The user does not have the right to perform the operation.
500	E112	E_INVALID_APPLICATION	3	Form file does not exist for the given application.

500	E123	E_PARAMS_NOT_EXISTS	3	Parameters does not exist for the given ids.
500	E121	E_INVALID_PARAM_VALUE	3	Invalid parameters values given.
500	E103	E_GENERAL_ERROR	3	API failed.

UPDATE PARAMETER

This API is used to update the application parameters of a device.

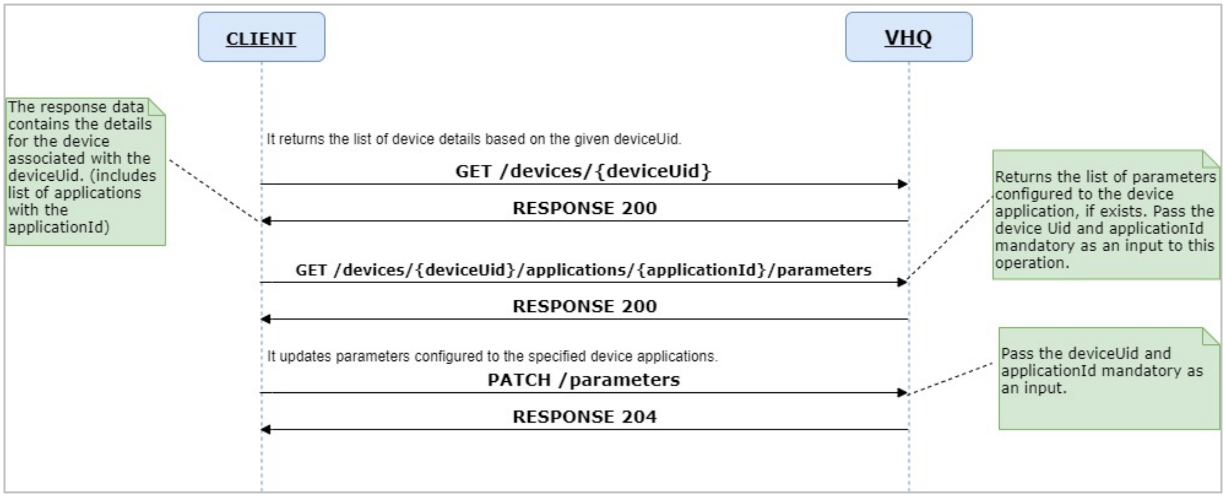
Request Body

```
{
  "data": [
    {
      "deviceUid": 0,
      "customerId": 0,
      "applicationId": 0,
      "applicationName": string",
      "applicationVersion": string",
      "deviceParameters": [
        {
          "name": "string",
          "value": "string",
        }
      ]
    }
  ]
}
```

Go to the following path:

```
API - PATCH/parameters
URL - https://<API Server>/apis/v1.3/parameters
e.g. https://vhqtest.verifone.com/apis/v1.3/parameters
```

Sequence Diagram


CLIENT
VHQ
The response data contains the details for the device associated with the deviceUid. (includes list of applications with the applicationId)
It returns the list of device details based on the given deviceUid.
GET /devices/{deviceUid}
RESPONSE 200
Returns the list of parameters configured to the device application, if exists. Pass the device Uid and applicationId mandatory as an input to this operation.
GET /devices/{deviceUid}/applications/{applicationId}/parameters
RESPONSE 200
It updates parameters configured to the specified device applications.
PATCH /parameters
Pass the deviceUid and applicationId mandatory as an input.
RESPONSE 204

Input Parameters

Element	attribute	Optional/ Mandatory	Data Type	Description
deviceUid		Mandatory	Int	Unique device identifier to be set as the path of the URI.
applicationId / applicationName / applicationName and applicationVersion		Mandatory	Int	Application Identifier.
parameters (collection of parameters)	id	Mandatory	Int	Parameter identifier.
	value	Mandatory	String	Parameter value.

Output Parameters

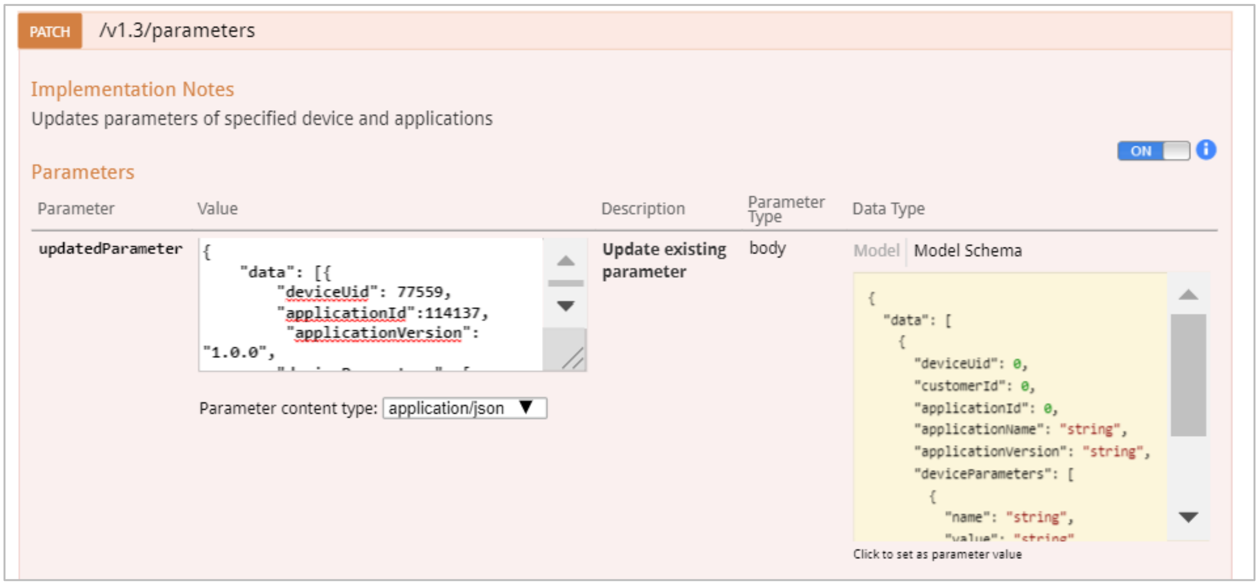
None

Update Parameter - Sample Code

SAMPLE CODE SNIPPET

```
public class UpdateParameter {
    // NOTE: Replace data with appropriate values
   private static String requestBody = "{\r\n" +
         "    \"data\": [\r\n" +
         "        {\r\n" +
         "            \"customerId\": 1473,\r\n" +
         "            \"applicationId\": 112343,\r\n" +
         "            \"deviceParameters\": [\r\n" +
         "                {\r\n" +
         "                    \"key\": false,\r\n" +
         "                    \"name\": \"Merchant.1/address_2\",\r\n" +
         "                    \"value\": \"6\"\r\n" +
         "                }\r\n" +
         "            ],\r\n" +
         "            \"deviceUid\": 2591\r\n" +
         "        }\r\n" +
         "    ]\r\n" +
         "}"; // Patch Request body in JSON format
   private static String url = "http://<servername:port>/apis/v1.3/";
   private int statusCode;
// NOTE: Replace the request body values with appropriate data.
   /**
    * @param Api -> This refers to the endpoint of the URL for the particular entity e.g. for Hierarchy entity it is
hierarchies
    * @return
    */
    HttpHeaders getHeader(String Api) {
         HttpHeaders headers = new HttpHeaders();
         headers.set("Authorization", "Bearer 1c58de5b-7033-3cec-a2c4-99d2668f95c9");
         headers.setContentType(MediaType.APPLICATION_JSON);
         return headers;
         }
   /**
    * @param Api ->  This refers to the endpoint of the URL for the particular entity e.g. for Parameter entity it is
parameters
    * @param body -> The request body which is required to be patched, This changes as per the entity
    * @param id   -> This refer to which specific Row need to change
    */
   void patch(String api, String body,String[] queryparameter,String [] queryparamvalue) {
      String BASE_PATH = url.concat(api) + "/" ;
      url =url.concat(api) +"?";
      if(null!=queryparameter && null!=queryparamvalue){
         int value=0;
         url = url  + queryparameter[value]+ "=" + queryparamvalue[value] ;

      }
      System.out.println(url);

      // Getting All Header Information
      // like(contentType,CustomerName,CustomerId)
      HttpHeaders headers = getHeader(api);
      try {
         HttpEntity<String> entity = new HttpEntity<String>(body, headers);
         RestTemplate = new RestTemplate();
         HttpComponentsClientHttpRequestFactory requestFactory = new HttpComponentsClientHttpRequestFactory();
         requestFactory.setConnectTimeout(18000);
         requestFactory.setReadTimeout(18000);
         restTemplate.setRequestFactory(requestFactory);
         System.out.println("BASE_PATH ::"+BASE_PATH);
         ResponseEntity<String> responseEntity  = restTemplate.exchange(url, HttpMethod.PATCH, entity, String.class);
         statusCode = responseEntity.getStatusCode().value();
         System.out.println(statusCode);
         System.out.println(responseEntity.getBody());
      } catch (Exception ex) {
         ex.printStackTrace();
         if (ex.getMessage().equalsIgnoreCase("404 Not Found")) {
            statusCode = 404;
            throw new RuntimeException("Not Found " + statusCode);
         } else if (ex.getMessage().equalsIgnoreCase("400 Bad Request")) {
            statusCode = 400;
            throw new RuntimeException("Bad Request " + statusCode);
         } else if (ex.getMessage().equalsIgnoreCase("409 Conflict")) {
            statusCode = 409;
            throw new RuntimeException("Already Exist : Conflict " + statusCode);
         } else {
            statusCode = 500;
            throw new RuntimeException("Already Exist : Conflict " + statusCode);
         }
      }
      System.out.println("Updated Successfully");
   }
   public static void main(String[] args) {
      String []queryparameter= {"customerId"};
      String []queryparamvalue= {"1473"};
      new UpdateParameter().patch("parameters", requestBody,queryparameter,queryparamvalue);
   }
}
```

Sample using Swagger – Request



/v1.3/parameters

Implementation Notes

Updates parameters of specified device and applications

Parameters

Parameter	Value	Description	Parameter Type	Data Type
updatedParameter		**Update existing parameter**	body	Model | Model Schema

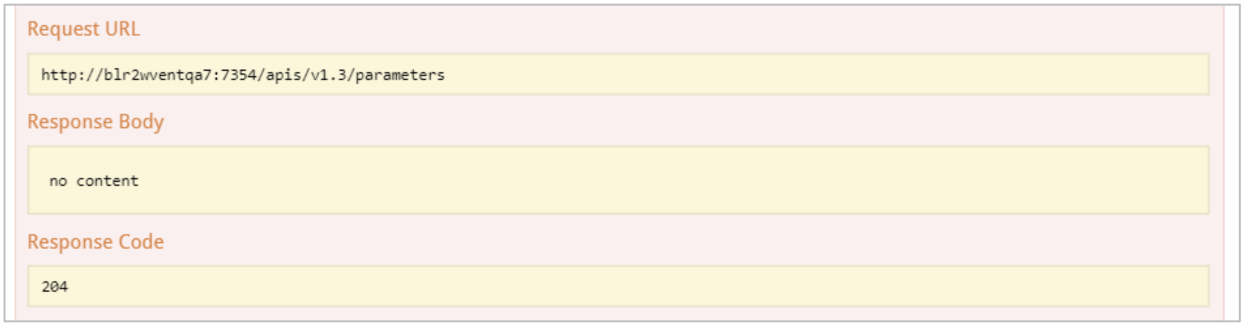
```
{
    "data": [{
        "deviceUid": 77559,
        "applicationId":114137,
         "applicationVersion":
"1.0.0",
```

Parameter content type: application/json

```
{
  "data": [
    {
      "deviceUid": 0,
      "customerId": 0,
      "applicationId": 0,
      "applicationName": "string",
      "applicationVersion": "string",
      "deviceParameters": [
        {
          "name": "string",
          "value": "string"
```

Click to set as parameter value

Sample using Swagger – Response



Request URL

```
http://blr2wventqa7:7354/apis/v1.3/parameters
```

Response Body

```
no content
```

Response Code

```
204
```

Error Codes

HTTP Status Code	Application Code	Name	Severity	Message
200	S200	S_SUCCESS	-	Success.
401	E101	E_TOKEN_INVALID_OR_EXPIRED	2	Token is expired or invalid.
401	E102	E_UNAUTHORIZED_ACCESS	2	The user does not have the right to perform the operation.
500	E112	E_INVALID_APPLICATION	3	Form file does not exist for the given application.
500	E123	E_PARAMS_NOT_EXISTS	3	Parameters does not exist with given ids.

500	E121	E_INVALID_PARAM_VALUE	3	Invalid parameter values given.
500	E103	E_GENERAL_ERROR	3	API failed.

DELETE PARAMETER

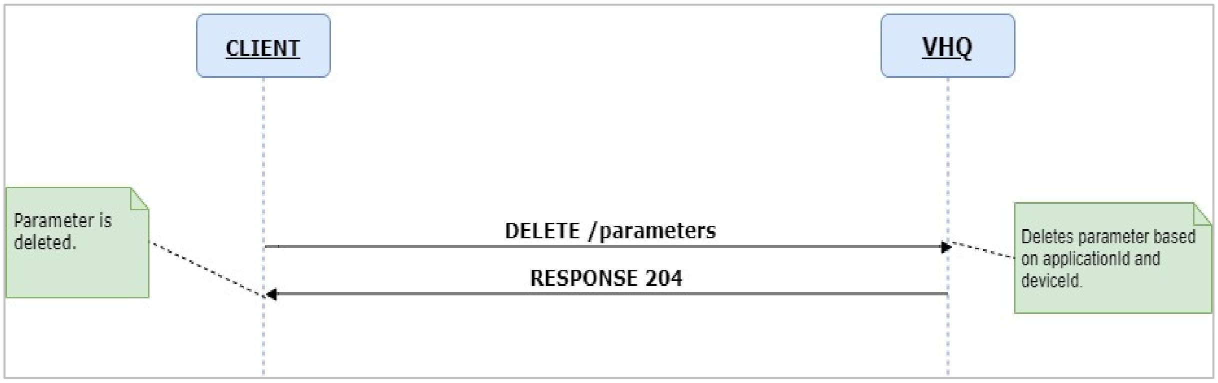
This API is used to delete a device parameter of a given deviceUid.

```
{
  "data": [
    {
      "deviceUid": 0,
      "customerId": 0,
      "applicationId": 0,
      "applicationName": string",
      "applicationVersion": string",
      "deviceParameters": [
        {
          "name": "string",
          "value": "string",
        }
      ]
    }
  ]
}
```

Go to the following path:

```
API - Delete/Parameters
URL - https://<API Server>/apis/v1.3/parameters
e.g. https://vhqtest.verifone.com/apis/v1.3/parameters
```

Sequence Diagram



Input Parameters

Element	attribute	Optional/ Mandatory	Data Type	Description
deviceUid		Mandatory	Int	Unique device identifier to be set as the path of the URI.
applicationId / applicationName / applicationName and applicationVersion		Mandatory	Int	Application Identifier.
parameters (collection of parameters)	id	Mandatory	Int	Parameter identifier.
	value	Mandatory	String	Parameter value.

Output Parameters

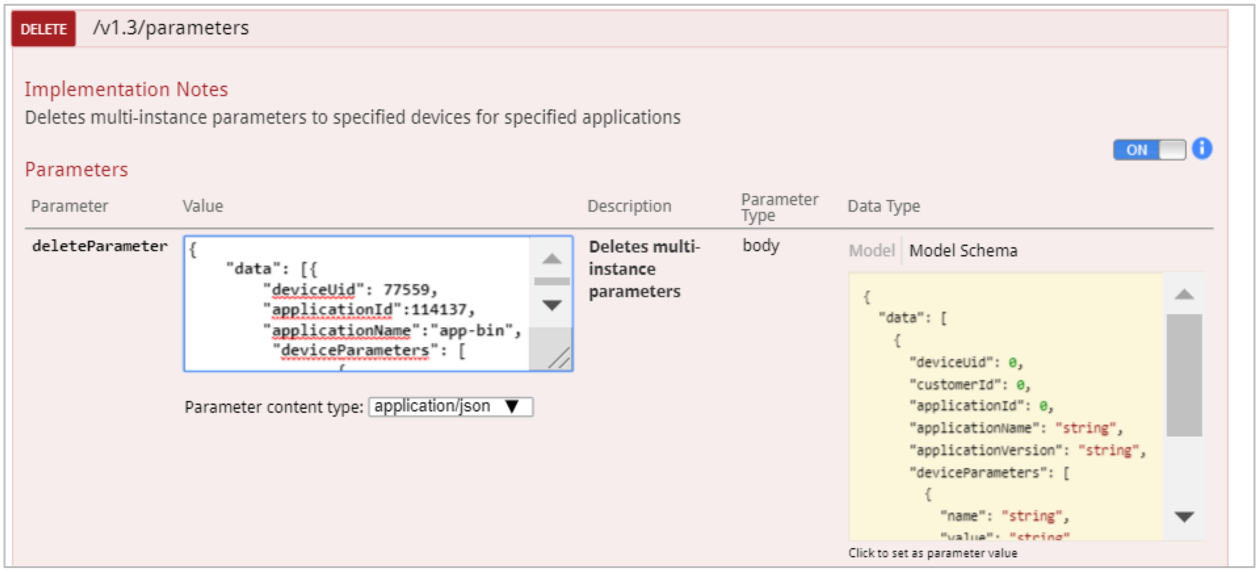
None

Delete Parameter API - Sample Code

SAMPLE CODE SNIPPET

```
public class DeleteParameters {
   private static String requestBody = "{\r\n" +
         "    \"data\": [{\r\n" +
         "        \"deviceUid\": 77559,\r\n" +
         "        \"applicationId\":114137,\r\n" +
         "        \"applicationName\":\"app-bin\",\r\n" +
         "         \"deviceParameters\": [\r\n" +
         "                {\r\n" +
         "                    \"key\": false,\r\n" +
         "                    \"name\": \"Merchant.1/address_1\"\r\n" +
         "                    \r\n" +
         "                 \r\n" +
         "                }\r\n" +
         "              \r\n" +
         "               ],\r\n" +
         "        \"customerId\": 11482\r\n" +
         "    }]\r\n" +
         "}"; // Patch Request body in JSON format
   private static String url = "http://<servername:port>/apis/v1.3/";
   private int statusCode;

   /**
    * @param Api -> This refers to the endpoint of the URL for the particular entity e.g. for Parameter entity it is
parameters
    * @return
    */
   HttpHeaders getHeader(String Api) {
      HttpHeaders headers = new HttpHeaders();
      headers.set("Authorization", "Bearer ffa840ff-489d-3d9c-94dc-9024e8bbcba9");
      headers.setContentType(MediaType.APPLICATION_JSON);
      return headers;
   }
   /**
    * @param Api ->  This refers to the endpoint of the URL for the particular entity e.g. for Parameter entity it is
parameters
    * @param body -> The request body which is required to be patched, This changes as per the entity
    * @param id   -> This refer to which specific Row need to cheange
    */
   void delete(String api, String body,String[] queryparameter,String [] queryparamvalue) {
      //String BASE_PATH = url.concat(api) + "/" ;
      url =url.concat(api) +"?";
      if(null!=queryparameter && null!=queryparamvalue){
         int value=0;
         url = url  + queryparameter[value]+ "=" + queryparamvalue[value] ;

      }
      System.out.println(url);

      // Getting All Header Information
      // like(contentType,CustomerName,CustomerId)
      HttpHeaders headers = getHeader(api);
      try {
         HttpEntity<String> entity = new HttpEntity<String>(body, headers);
         RestTemplate = new RestTemplate();
         HttpComponentsClientHttpRequestFactory requestFactory = new HttpComponentsClientHttpRequestFactory();
         requestFactory.setConnectTimeout(18000);
         requestFactory.setReadTimeout(18000);
         restTemplate.setRequestFactory(requestFactory);
         //System.out.println("BASE_PATH ::"+BASE_PATH);
         ResponseEntity<String> responseEntity  = restTemplate.exchange(url, HttpMethod.DELETE, entity, String.class);
         statusCode = responseEntity.getStatusCode().value();
         System.out.println(statusCode);
      } catch (Exception ex) {
         ex.printStackTrace();
         if (ex.getMessage().equalsIgnoreCase("404 Not Found")) {
            statusCode = 404;
            throw new RuntimeException("Not Found " + statusCode);
         } else if (ex.getMessage().equalsIgnoreCase("400 Bad Request")) {
            statusCode = 400;
            throw new RuntimeException("Bad Request " + statusCode);
         } else if (ex.getMessage().equalsIgnoreCase("409 Conflict")) {
            statusCode = 409;
            throw new RuntimeException("Already Exist : Conflict " + statusCode);
         } else {
            statusCode = 500;
            throw new RuntimeException("Already Exist : Conflict " + statusCode);
         }
      }
      System.out.println("Updated Successfully");
   }
   public static void main(String[] args) {
      String []queryparameter= {"customerId"};
      String []queryparamvalue= {"11482"};
      new DeleteParameters().delete("parameters", requestBody,queryparameter,queryparamvalue);
   }
}
```

Sample using Swagger – Request



Implementation Notes

Deletes multi-instance parameters to specified devices for specified applications

Parameters

Parameter	Value	Description	Parameter Type	Data Type
deleteParameter	(see below)	Deletes multi-instance parameters	body	Model | Model Schema

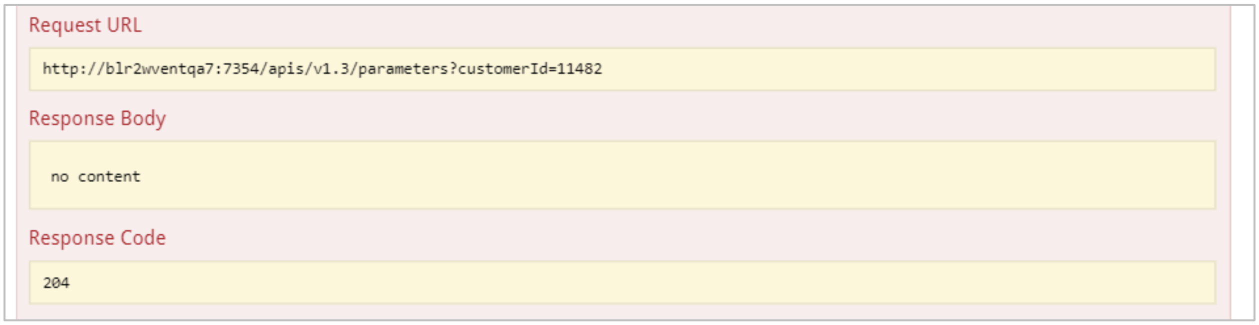
```
{
     "data": [{
          "deviceUid": 77559,
          "applicationId":114137,
          "applicationName":"app-bin",
           "deviceParameters": [
```

Parameter content type: application/json

```
{
  "data": [
    {
      "deviceUid": 0,
      "customerId": 0,
      "applicationId": 0,
      "applicationName": "string",
      "applicationVersion": "string",
      "deviceParameters": [
        {
          "name": "string",
          "value": "string"
```

Click to set as parameter value

Sample using Swagger - Response


Request URL

```
http://blr2wventqa7:7354/apis/v1.3/parameters?customerId=11482
```

Response Body

```
no content
```

Response Code

```
204
```

Error Codes

HTTP Status Code	Application Code	Name	Severity	Message
200	S200	S_SUCCESS	-	Success.
401	E101	E_TOKEN_INVALID_OR_EXPIRED	2	Token is expired, or invalid.
401	E102	E_UNAUTHORIZED_ACCESS	2	The user does not have the right to perform the operation.
500	E112	E_INVALID_APPLICATION	3	Form file does not exist for the given application.
500	E123	E_PARAMS_NOT_EXISTS	3	Parameters does not exist with given ids.
500	E121	E_INVALID_PARAM_VALUE	3	Invalid parameters values given.
500	E103	E_GENERAL_ERROR	3	API failed.

GET PARAMETERS

This API is used to get the parameters of a specific device Uid and application ID.

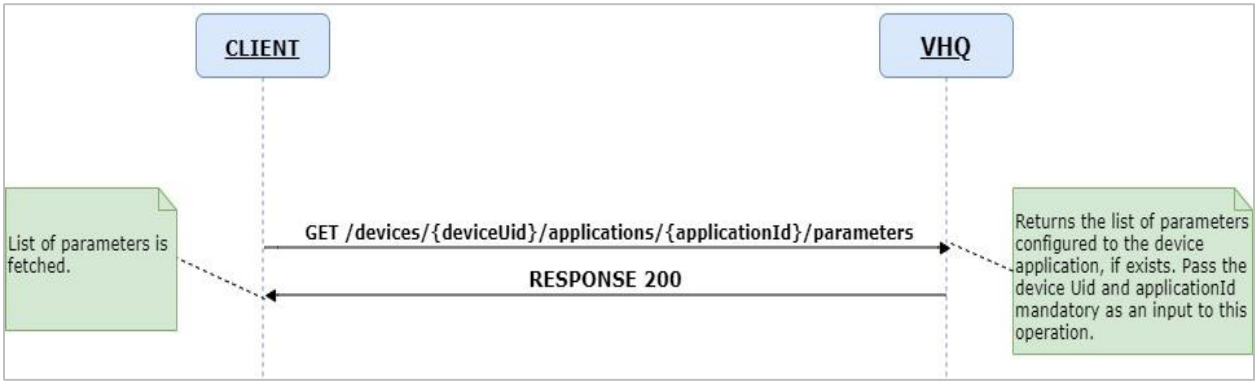
Get Parameters by DeviceUid

- This API gets the following fields in a fixed order: deviceUid, applicationId, and parameter name (by complete path including levels.)
- This API returns all three fields, deviceUid, applicationId, and parameter name irrespective of the fields mentioned as part of the **field(s)** clause in the request URL.
- Pagination parameters apply to the device parameters and not the devices.

Once you have the device Uid and application ID, go to the following path:

```
API - GET/devices/{deviceUid}/applications/{applicationId}/parameters
URL-
https://<APIServer>/apis/v1.3/devices/{deviceUid}/applications/{applicationId}paramete
rs
e.g.
https://vhqtest.verifone.com/apis/v1.3/devices/{deviceUid}/applications/{applicationId
}parameters
```

Sequence Diagram


CLIENT
VHQ
List of parameters is fetched.
GET /devices/{deviceUid}/applications/{applicationId}/parameters
RESPONSE 200
Returns the list of parameters configured to the device application, if exists. Pass the device Uid and applicationId mandatory as an input to this operation.

Input Parameters

Element	Optional/ Mandatory	Data Type	Description
udi	Mandatory	Int	Unique device identifier to be sent in the path of the URI.
appId	Mandatory	Int	Application Identifier.

Output Parameters

Element	Attribute	Data Type	Description
parameters (collection of device parameters)	id	int	Device parameter Identifier.
	name	String	Name of the parameter.
	value	String	Value of the parameter.

Get Parameters API - Sample Code

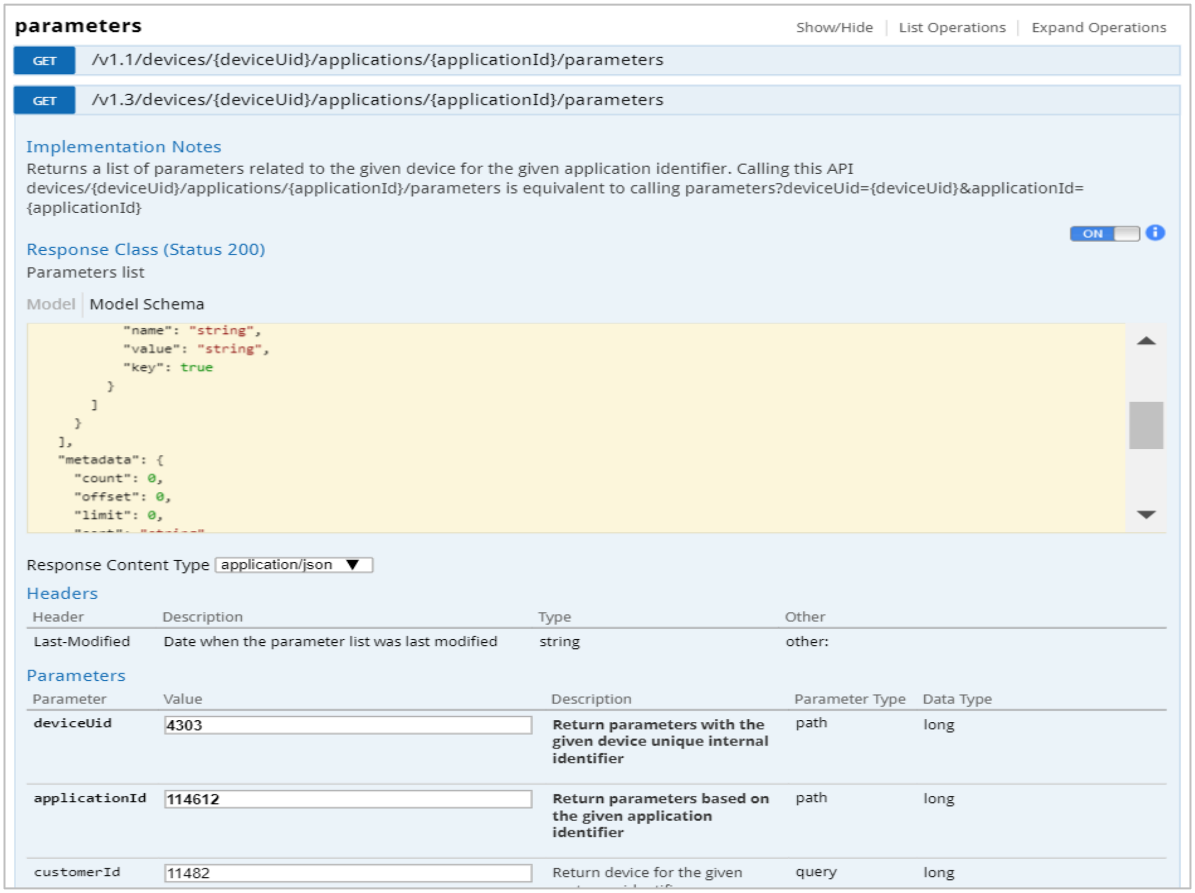
SAMPLE CODE SNIPPET

```
public class GetDeviceAppParameters {
   private static String url = "http://<servername:port>/apis/v1.3/";
   private int statusCode;
   private static String fullPath="applications/114137/parameters";
   // This is the base URL, it will be appended as per the entity which is invoked

   /**
    * @param completeURL -> This is the URL which appended as per the
   entity which is invoked
    * @param method -> This refer to which method we are going to
   send e.g. get,post,put
    * @return -> httpCon with Content Type and Authorization token
```

```
 * @throws IOException
 */
HttpURLConnection getConnection(URL completeURL, String method)
      throws IOException {
   HttpURLConnection httpCon = (HttpURLConnection)
         completeURL.openConnection();
   httpCon.setDoOutput(true);
   httpCon.setRequestProperty("Authorization", "Bearer b6223da0-c941-3342-83b8-e3965ae20484");
   httpCon.setRequestMethod(method);
   httpCon.setRequestProperty("Content-Type", "application/json");
   return httpCon;
}
/**
 * This method get all data of a entity
 * @param Api -> This refers to the endpoint of the URL for the
particular entity e.g. for Parameter entity it is parameters
 * @param id -> This refer to show specific data from Entity based
on Id.
 * @param fields -> This refers to show specific fields, which we
pass as a parameter.
 * @param queryparameter
 * @return response from entity
 */
String getById(String Api, int id,String fullPath,String[] queryparameter,String queryParamValue) {
   StringBuffer response=null;
   if(id!=0) {
      url = url.concat(Api) + "/" + id;
   }else {
      url = url.concat(Api);
   }

   if(fullPath!=null){
      url=url+"/".concat(fullPath);
   }
   if(null!=queryparameter){
      for(String fields: queryparameter){
         url = url +"?"+fields+"="+queryParamValue;
         System.out.println("URL :::: "+url);
      }
   }
   try {
      URL completeURL = new URL(url);
      // Getting All Header Information
      // like(contentType,CustomerName,CustomerId)
      HttpURLConnection httpCon = getConnection(completeURL, "GET");
      httpCon.getResponseCode();
      BufferedReader in = new BufferedReader(new InputStreamReader(httpCon.getInputStream()));
      String inputLine;
      response =new StringBuffer();
      while ((inputLine = in.readLine()) != null) {
         response.append(inputLine);
      }
      System.out.println(response);
      System.out.println(httpCon.getResponseCode());
      System.out.println(httpCon.getResponseMessage());
   } catch (Exception ex) {
      ex.printStackTrace();
      if (ex.getMessage().equalsIgnoreCase("404 Not Found")) {
         statusCode = 404;
         throw new RuntimeException("Not Found " + statusCode);

      } else if (ex.getMessage().equalsIgnoreCase("400 Bad Request")) {
         statusCode = 400;
         throw new RuntimeException("Bad Request " + statusCode);
      } else if (ex.getMessage().equalsIgnoreCase("409 Conflict"))
      {
         statusCode = 409;
         throw new RuntimeException("Already Exist : Conflict " +
               statusCode);
      } else {
         statusCode = 500;
         throw new RuntimeException("Already Exist : Conflict " +
               statusCode);
      }
   }
   return response.toString();
}
public static void main(String[] args) {
   String [] queryparameter= {"customerId"};
```

```
      String queryParamValue = "11482";
      new GetDeviceAppParameters().getById("devices",77559,fullPath,queryparameter,queryParamValue);
   }
}
```

Sample using Swagger – Request



/v1.1/devices/{deviceUid}/applications/{applicationId}/parameters

/v1.3/devices/{deviceUid}/applications/{applicationId}/parameters

Implementation Notes

Returns a list of parameters related to the given device for the given application identifier. Calling this API devices/{deviceUid}/applications/{applicationId}/parameters is equivalent to calling parameters?deviceUid={deviceUid}&applicationId={applicationId}

Response Class (Status 200)

Parameters list

Model | Model Schema

```
            "name": "string",
            "value": "string",
            "key": true
          }
        ]
      }
    ],
    "metadata": {
      "count": 0,
      "offset": 0,
      "limit": 0,
```

Response Content Type application/json ▼

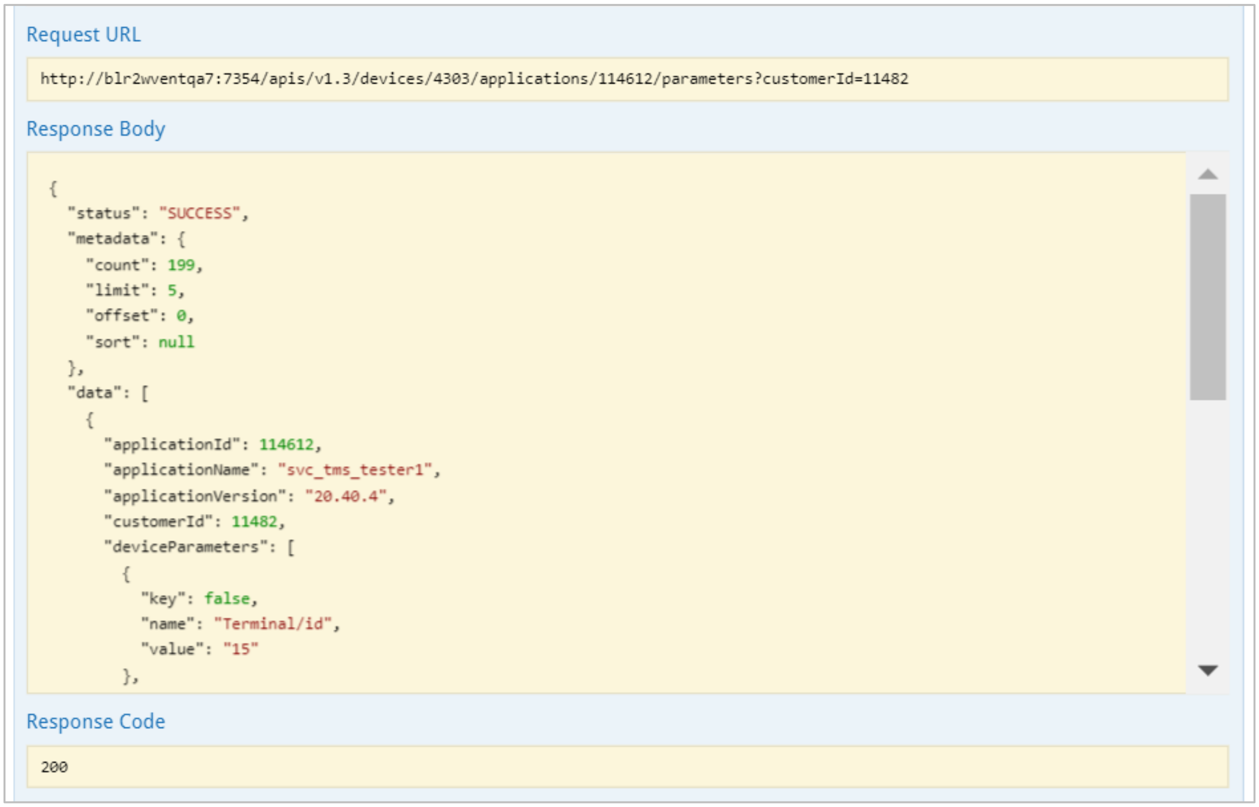
Headers

Header	Description	Type	Other
Last-Modified	Date when the parameter list was last modified	string	other:

Parameters

Parameter	Value	Description	Parameter Type	Data Type
deviceUid	4303	**Return parameters with the given device unique internal identifier**	path	long
applicationId	114612	**Return parameters based on the given application identifier**	path	long
customerId	11482	Return device for the given	query	long

Sample using Swagger – Response



Request URL

```
http://blr2wventqa7:7354/apis/v1.3/devices/4303/applications/114612/parameters?customerId=11482
```

Response Body

```
{
  "status": "SUCCESS",
  "metadata": {
    "count": 199,
    "limit": 5,
    "offset": 0,
    "sort": null
  },
  "data": [
    {
      "applicationId": 114612,
      "applicationName": "svc_tms_tester1",
      "applicationVersion": "20.40.4",
      "customerId": 11482,
      "deviceParameters": [
        {
          "key": false,
          "name": "Terminal/id",
          "value": "15"
        },
```

Response Code

```
200
```

Get Parameters API

This API will get all the parameters of a device.

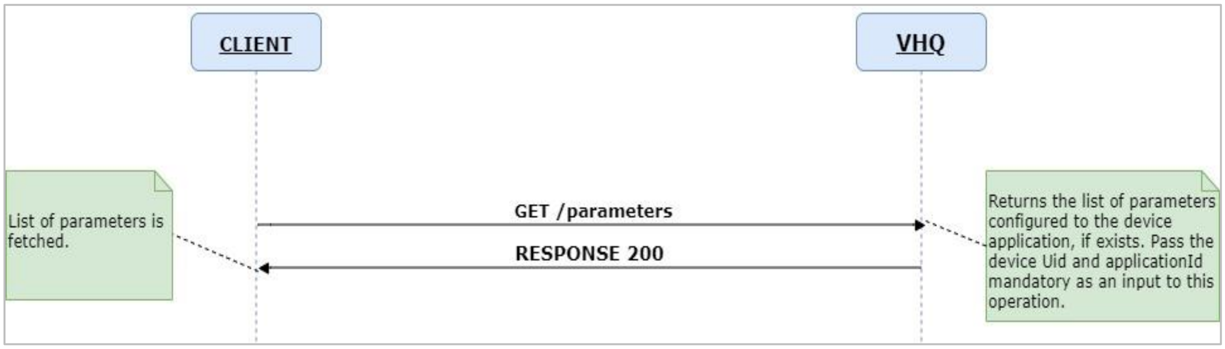
Go to the following path:

API - GET/parameters

URL- https://<APIServer>/apis/v1.3/devices/parameters

e.g. https://vhqtest.verifone.com/apis/v1.3/parameters

Sequence Diagram

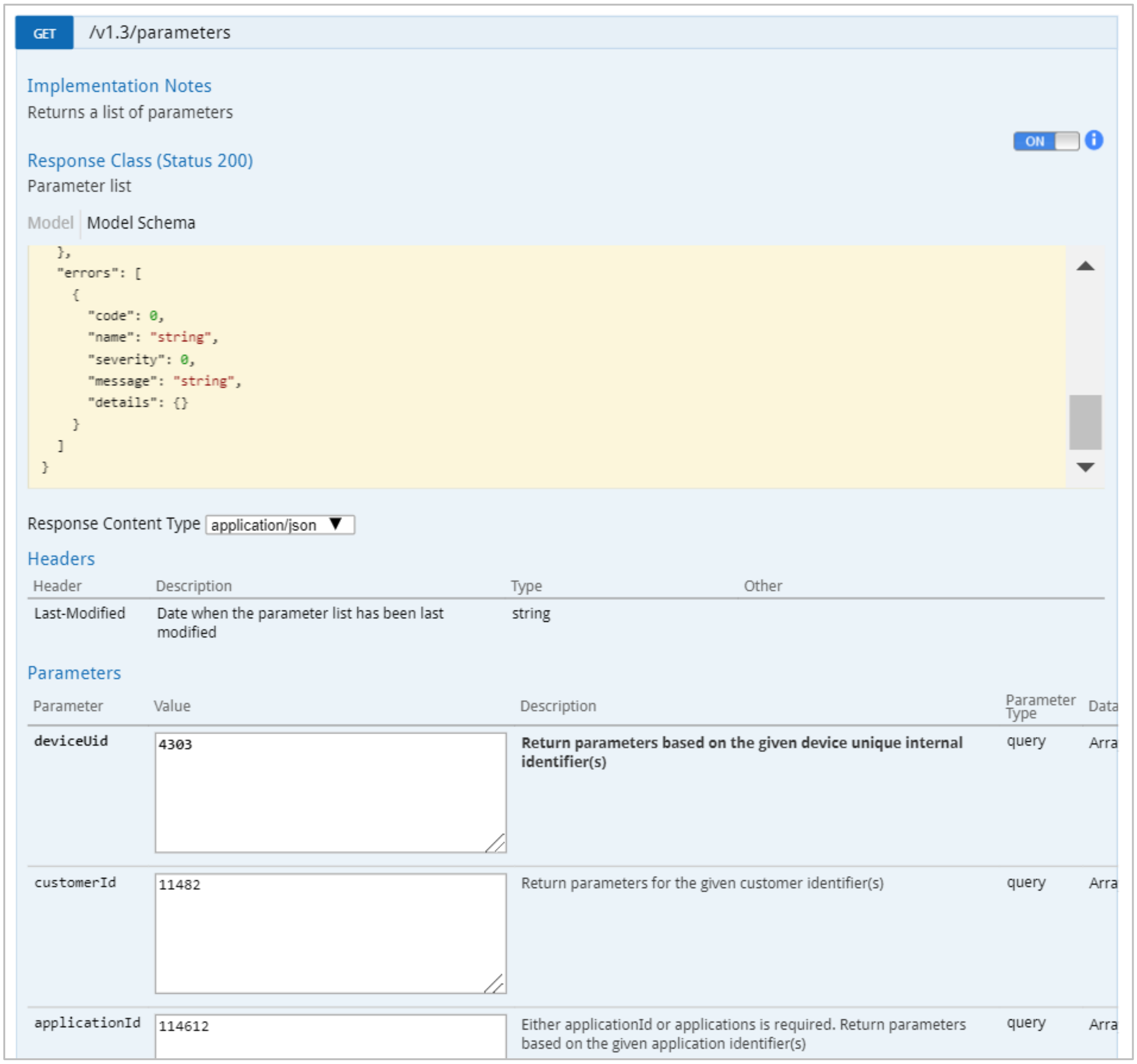

CLIENT
VHQ
List of parameters is fetched.
GET /parameters
RESPONSE 200
Returns the list of parameters configured to the device application, if exists. Pass the device Uid and applicationId mandatory as an input to this operation.

Get Parameters API - Sample Code

SAMPLE CODE SNIPPET

```
public class GetParameters {
   //Common API URL
   private static String url = "http://<servername:port>/apis/v1.3/";
   private int statusCode;
   // This is the base URL, it will be appended as per the entity which is invoked
      /**
    * @param completeURL -> This is the URL which appended as per the
   entity which is invoked
    * @param method -> This refer to which method we are going to
   send e.g. get,post,put
    * @return -> httpCon with Content Type and Authorization token
    * @throws IOException
    */
   HttpURLConnection getConnection(URL completeURL, String method)
         throws IOException {
      HttpURLConnection httpCon = (HttpURLConnection)
            completeURL.openConnection();
      httpCon.setDoOutput(true);
      httpCon.setRequestProperty("Authorization", "Bearer d6627009-a687-3960-9597-dcf8ba6ab764");
      httpCon.setRequestMethod(method);
      httpCon.setRequestProperty("Content-Type", "application/json");
      return httpCon;
   }
   /**
    * This method get all data of a entity
    * @param Api -> This refers to the endpoint of the URL for the
   particular entity e.g. for Parameter entity it is parameters
    * @param id -> This refer to show specific data from Entiy based
   on Id.
    * @param fields -> This refers to show specific fields, which we
   have paas as a parameter.
    * @param queryparameter
    * @param queryparamvalue
    * @return response from entity
    */
   String getById(String Api, String fullPath,String[] queryparameter,String[] queryparamvalue) {
      StringBuffer response = null;
      System.out.println("fullPath  :: "+fullPath);
      url = url.concat(Api) + "?" ;
      if(fullPath!=null){
         url=url+"/".concat(fullPath);
      }
      if(null!=queryparameter && null!=queryparamvalue){
         int len = queryparameter.length;
         int value=0;
         while(value!=len){
            url = url  + queryparameter[value];
            url = url + "=" + queryparamvalue[value] ;
            value++;
            if(value!=len)
            {
               url=url+"&";
            }
         }
      }
      try {
         URL completeURL = new URL(url);
// Getting All Header Information
// like(contentType,CustomerName,CustomerId)
         HttpURLConnection httpCon = getConnection(completeURL, "GET");
         httpCon.getResponseCode();
         BufferedReader in = new BufferedReader(new InputStreamReader(httpCon.getInputStream()));
         String inputLine;
         response = new StringBuffer();
         while ((inputLine = in.readLine()) != null) {
            response.append(inputLine);
         }
         System.out.println(response);
         System.out.println(httpCon.getResponseCode());
         System.out.println(httpCon.getResponseMessage());
      } catch (Exception ex) {
         ex.printStackTrace();
         if (ex.getMessage().equalsIgnoreCase("404 Not Found")) {
            statusCode = 404;
            throw new RuntimeException("Not Found " + statusCode);
         } else if (ex.getMessage().equalsIgnoreCase("400 Bad Request")) {
            statusCode = 400;
            throw new RuntimeException("Bad Request " + statusCode);
         } else if (ex.getMessage().equalsIgnoreCase("409 Conflict"))
         {
            statusCode = 409;
            throw new RuntimeException("Already Exist : Conflict " +
                  statusCode);
         } else {
            statusCode = 500;
            throw new RuntimeException("Already Exist : Conflict " +
                  statusCode);
         }
      }
      return response.toString();
   }
   public static void main(String[] args) {
      //Post method will accept two parameter, Request Body and API
      String []queryparameter= {"deviceUid","applicationId","customerId"};
      String []queryparamvalue= {"77559","114137","11482"};
      new GetParameters().getById("parameters",null,queryparameter,queryparamvalue);
   }
}
```

Sample using Swagger – Request



Implementation Notes

Returns a list of parameters

Response Class (Status 200)

Parameter list

Model | Model Schema

```
  },
  "errors": [
    {
      "code": 0,
      "name": "string",
      "severity": 0,
      "message": "string",
      "details": {}
    }
  ]
}
```

Response Content Type application/json ▼

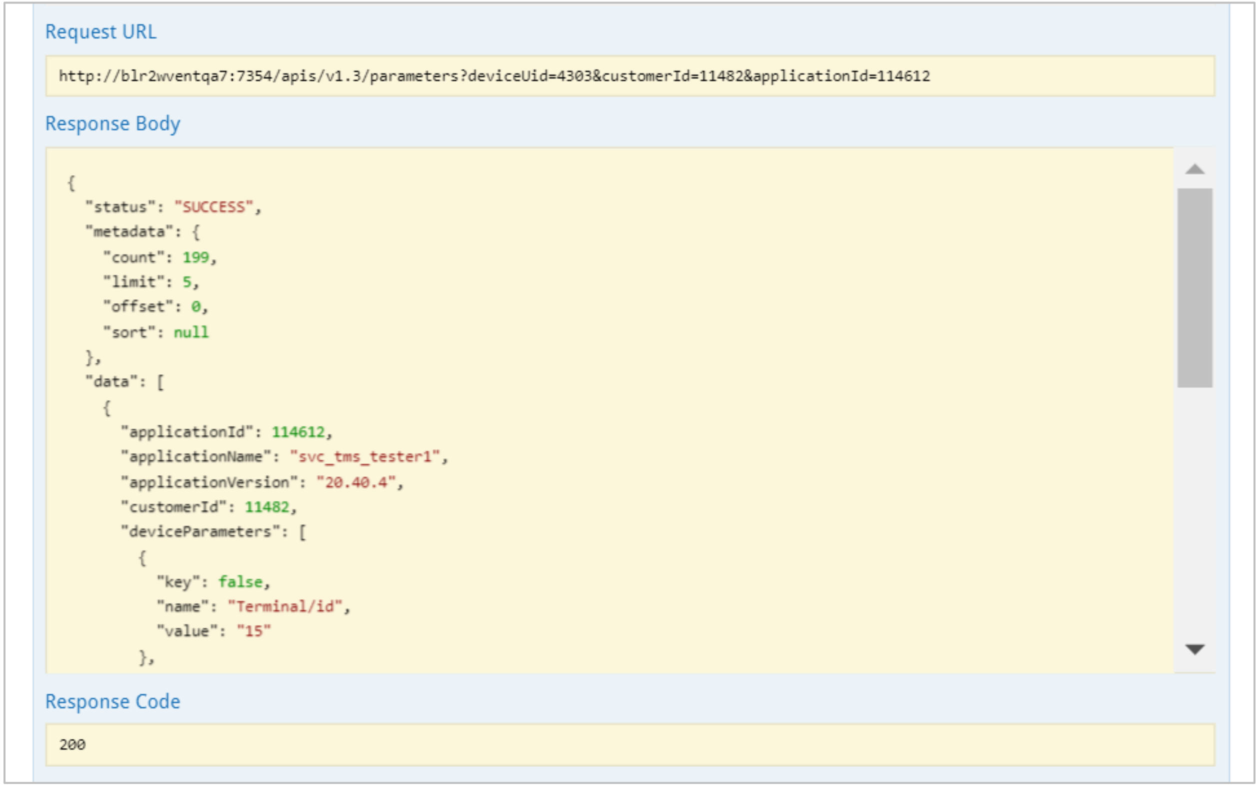
Headers

Header	Description	Type	Other
Last-Modified	Date when the parameter list has been last modified	string	

Parameters

Parameter	Value	Description	Parameter Type	Data
deviceUid	4303	**Return parameters based on the given device unique internal identifier(s)**	query	Arra
customerId	11482	Return parameters for the given customer identifier(s)	query	Arra
applicationId	114612	Either applicationId or applications is required. Return parameters based on the given application identifier(s)	query	Arra

Sample using Swagger - Response



Request URL

```
http://blr2wventqa7:7354/apis/v1.3/parameters?deviceUid=4303&customerId=11482&applicationId=114612
```

Response Body

```
{
  "status": "SUCCESS",
  "metadata": {
    "count": 199,
    "limit": 5,
    "offset": 0,
    "sort": null
  },
  "data": [
    {
      "applicationId": 114612,
      "applicationName": "svc_tms_tester1",
      "applicationVersion": "20.40.4",
      "customerId": 11482,
      "deviceParameters": [
        {
          "key": false,
          "name": "Terminal/id",
          "value": "15"
        },
```

Response Code

```
200
```

Error Codes

HTTP Status Code	Application Code	Name	Severity	Message
200	S100	S_OK	-	Success.
401	E101	E_TOKEN_INVALID_OR_EXPIRED	2	Token expired or invalid.
401	E102	E_UNAUTHORIZED_ACCESS	2	The user does not have the right to perform the operation.
500	E103	E_GENERAL_ERROR	3	API failed.

GET PARAMETER TEMPLATES

This API is used to get the list of parameter templates for the application ID.

Go to the following path:

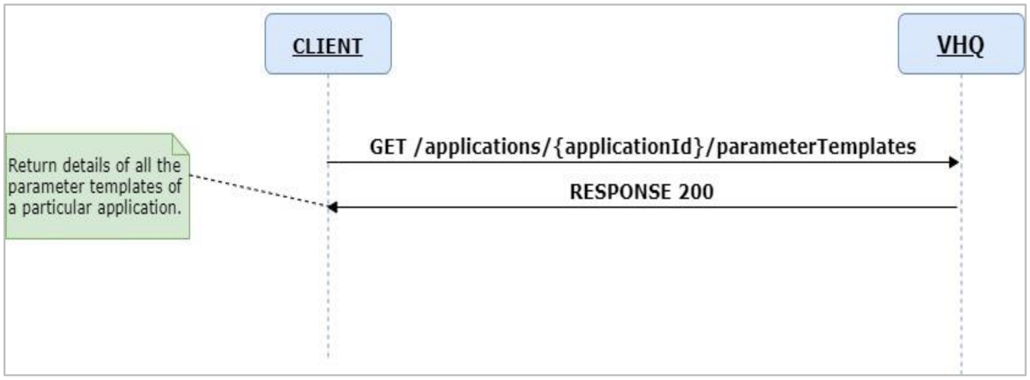
API - GET/applications/{applicationId}/parameterTemplates

URL - https://<APIError! Hyperlink reference not valid.Server>/apis/v1.1/applications/{applicationId}/parameterTemplates

e.g.

https://vhqtest.verifone.com/apis/v1.1/applications/{applicationId}/parameterTemplates

Sequence Diagram


CLIENT
VHQ
GET /applications/{applicationId}/parameterTemplates
RESPONSE 200
Return details of all the parameter templates of a particular application.

Input Parameters

Element	Optional/Mandatory	Data Type	Description
applicationId	Mandatory	Int	Identifier of the application to be set as the path of the URI.

Output Parameters

Element	Attribute	Data Type	Description
templateParameters (collection of downloads)	Parameter Name	String	Parameter name with a fully qualified path including the parent parameter and instance.
	Parameter Value	String	Parameter value. The value should conform to the validation rules defined by the presentation file.

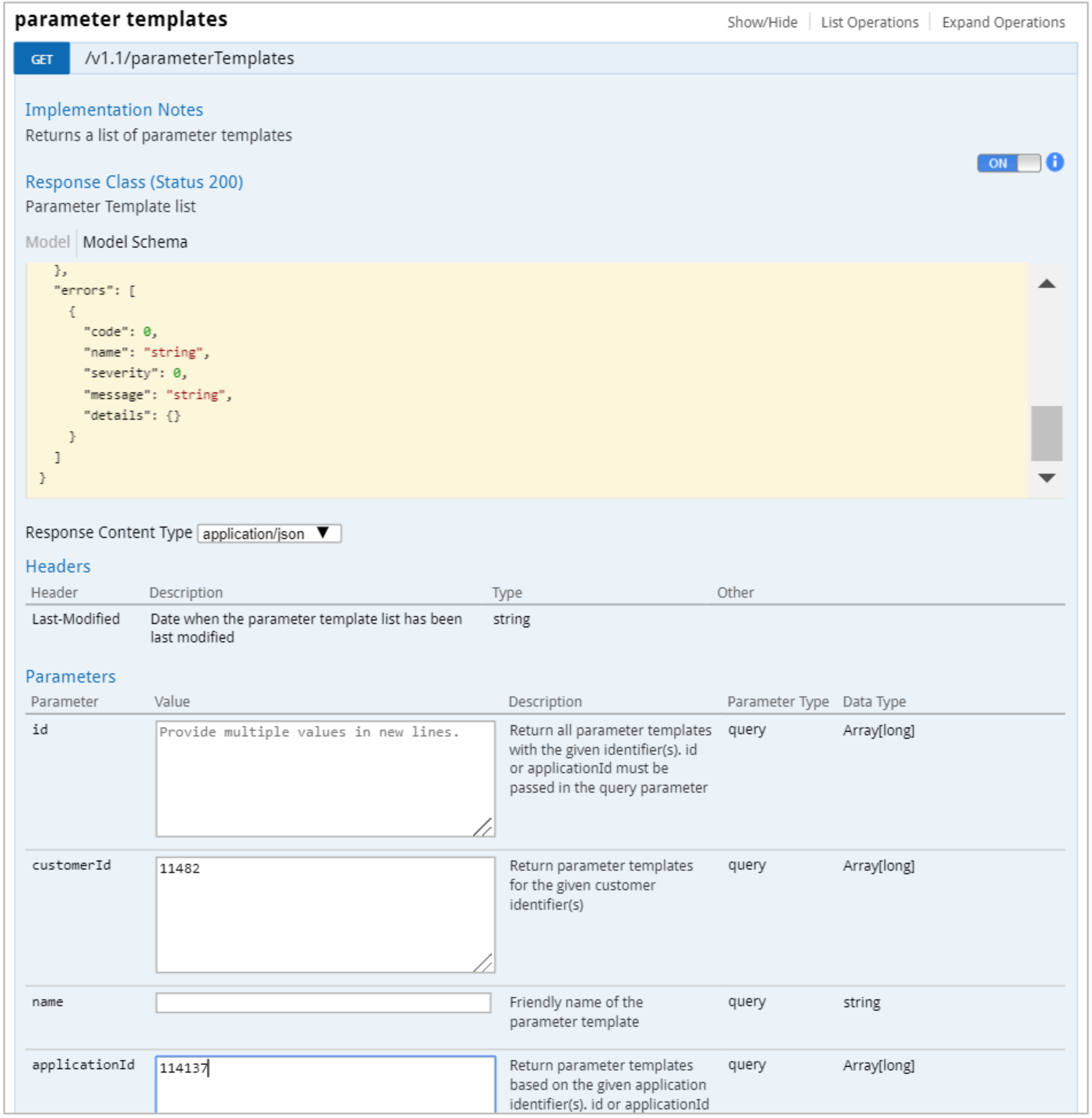
Get Parameter Templates API - Sample Code

SAMPLE CODE SNIPPET

```
public class GetParameterTemplate{
   private static String url = "http://<servername:port>/apis/v1.1/";
   private int statusCode;
   private static String fullPath="parameterTemplates";
   // This is the base URL, it will be appended as per the entity which is invoked

   /**
    * @param completeURL -> This is the URL which appended as per the
   entity which is invoked
    * @param method -> This refer to which method we are going to
   send e.g. get,post,put
    * @return -> httpCon with Content Type and Authorization token
    * @throws IOException
    */
   HttpURLConnection getConnection(URL completeURL, String method)
         throws IOException {
      HttpURLConnection httpCon = (HttpURLConnection)
            completeURL.openConnection();
      httpCon.setDoOutput(true);
      httpCon.setRequestProperty("Authorization", "Bearer 1102f4ff-dc68-30c8-aaa9-c40374ecde02");
      httpCon.setRequestMethod(method);
      httpCon.setRequestProperty("Content-Type", "application/json");
      return httpCon;
   }
   /**
```

```
    * This method get all data of a entity
    * @param Api -> This refers to the endpoint of the URL for the
   particular entity e.g. for Parameter entity it is parameters
    * @param id -> This refer to show specific data from Entity based
   on Id.
    * @param fields -> This refers to show specific fields, which we
   pass as a parameter.
    * @param queryparameter
    * @return response from entity
    */
   String getById(String Api, int id,String fullPath,String[] queryparameter,String queryParamValue) {
      StringBuffer response=null;
      if(id!=0) {
         url = url.concat(Api) + "/" + id;
      }else {
         url = url.concat(Api);
      }

      if(fullPath!=null){
         url=url+"/".concat(fullPath);
      }
      if(null!=queryparameter){
         for(String fields: queryparameter){
            url = url +"?"+fields+"="+queryParamValue;
            System.out.println("URL :::: "+url);
         }
      }
      try {
         URL completeURL = new URL(url);
         // Getting All Header Information
         // like(contentType,CustomerName,CustomerId)
         HttpURLConnection httpCon = getConnection(completeURL, "GET");
         httpCon.getResponseCode();
         BufferedReader in = new BufferedReader(new InputStreamReader(httpCon.getInputStream()));
         String inputLine;
         response =new StringBuffer();
         while ((inputLine = in.readLine()) != null) {
            response.append(inputLine);
         }
         System.out.println(response);
         System.out.println(httpCon.getResponseCode());
         System.out.println(httpCon.getResponseMessage());
      } catch (Exception ex) {
         ex.printStackTrace();
         if (ex.getMessage().equalsIgnoreCase("404 Not Found")) {
            statusCode = 404;
            throw new RuntimeException("Not Found " + statusCode);

         } else if (ex.getMessage().equalsIgnoreCase("400 Bad Request")) {
            statusCode = 400;
            throw new RuntimeException("Bad Request " + statusCode);
         } else if (ex.getMessage().equalsIgnoreCase("409 Conflict"))
         {
            statusCode = 409;
            throw new RuntimeException("Already Exist : Conflict " +
                  statusCode);
         } else {
            statusCode = 500;
            throw new RuntimeException("Already Exist : Conflict " +
                  statusCode);
         }
      }
      return response.toString();
   }
   public static void main(String[] args) {
      String [] queryparameter= {"customerId"};
      String queryParamValue = "11482";
      new GetParameterTemplate().getById("applications",114137,fullPath,queryparameter,queryParamValue);
   }
}
```

Sample using Swagger - Request



/v1.1/parameterTemplates

Implementation Notes

Returns a list of parameter templates

Response Class (Status 200)

Parameter Template list

Model | Model Schema

```
  },
  "errors": [
    {
      "code": 0,
      "name": "string",
      "severity": 0,
      "message": "string",
      "details": {}
    }
  ]
}
```

Response Content Type application/json

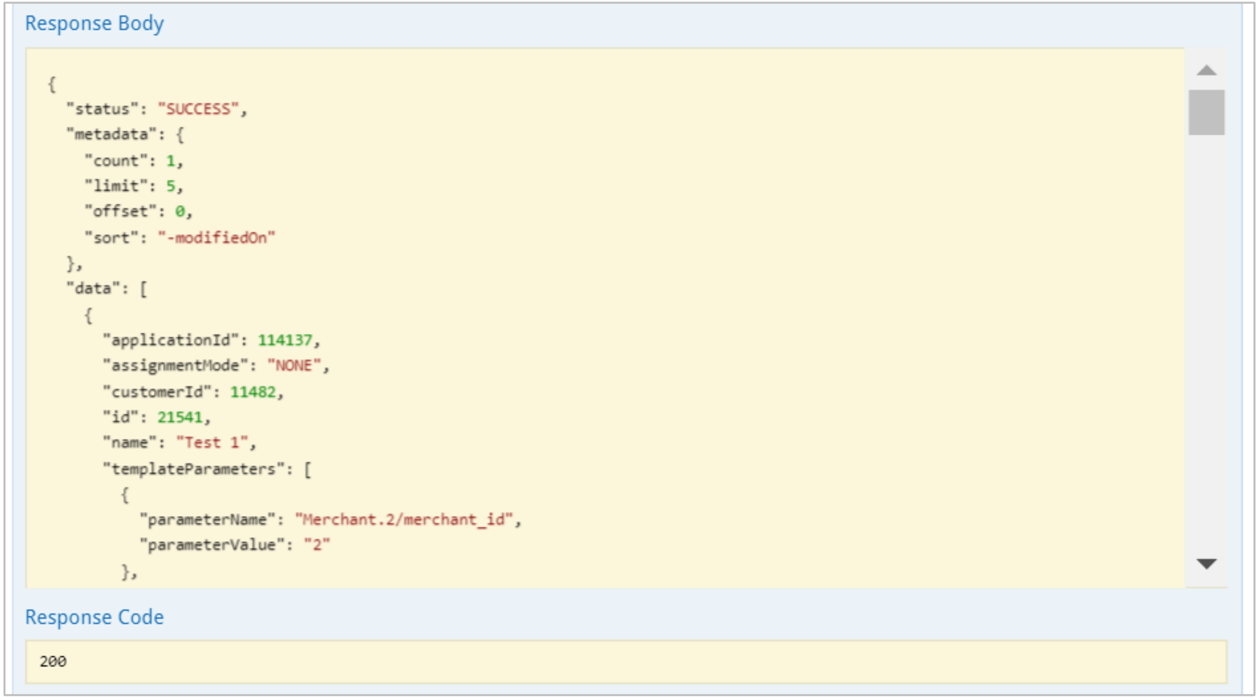
Headers

Header	Description	Type	Other
Last-Modified	Date when the parameter template list has been last modified	string	

Parameters

Parameter	Value	Description	Parameter Type	Data Type
id	Provide multiple values in new lines.	Return all parameter templates with the given identifier(s). id or applicationId must be passed in the query parameter	query	Array[long]
customerId	11482	Return parameter templates for the given customer identifier(s)	query	Array[long]
name		Friendly name of the parameter template	query	string
applicationId	114137	Return parameter templates based on the given application identifier(s). id or applicationId	query	Array[long]

Sample using Swagger – Response



Response Body

```
{
  "status": "SUCCESS",
  "metadata": {
    "count": 1,
    "limit": 5,
    "offset": 0,
    "sort": "-modifiedOn"
  },
  "data": [
    {
      "applicationId": 114137,
      "assignmentMode": "NONE",
      "customerId": 11482,
      "id": 21541,
      "name": "Test 1",
      "templateParameters": [
        {
          "parameterName": "Merchant.2/merchant_id",
          "parameterValue": "2"
        },
```

Response Code

```
200
```

Error Codes

HTTP Status Code	Application Code	Name	Severity	Message
200	S100	S_OK	-	Success.
401	E101	E_TOKEN_INVALID_OR_EXPIRED	2	Token expired or invalid.
401	E102	E_UNAUTHORIZED_ACCESS	2	The user does not have the right to perform the operation.
500	E103	E_GENERAL_ERROR	3	API failed.

HIERARCHY MANAGEMENT APIs

Hierarchies are the basis of device organization. The hierarchies in VHQ allow a high degree of flexibility and granularity in organizing the devices with up to fifteen levels of specificity. For example, hierarchies can be established based on Company, Region, Store, Lane, etc. Hierarchies help in creating structures aimed at organizing the managed devices into segments that are reflective of the topology into which these devices are deployed. The Hierarchies can be further enhanced to create groups that are indicative of how the devices are managed across organizational departments.

Create Hierarchies API is used to create a new hierarchy with a specific set of ReferenceSets and other non-mandatory fields as mentioned below according to the JSON request body.

```
{
  "data": {
    "id": 0,
    "customerId": 0,
    "hierarchyFullPath": "string",
    "parentHierarchyId": 0,
    "parentHierarchyName": 0,
    "name": "string",
    "description": "string",
    "ipStartingAddress": "string",
    "ipEndingAddress": "string",
    "locationIdentifier": "string",
    "timezoneId": 0,
    "entityId": "string",
    "childHierarchies": [
        {
         "id": 0,
         "name": "string"
        }
    ],

    "downloadAutomationEnabled": true,
    "inheritReferenceSet": true,
    "downloadOn": "NEXT_CONTACT",
    "referenceSets": [
      {
         "id": 0,
         "name": "string"
      }
    ]
  }
}
```

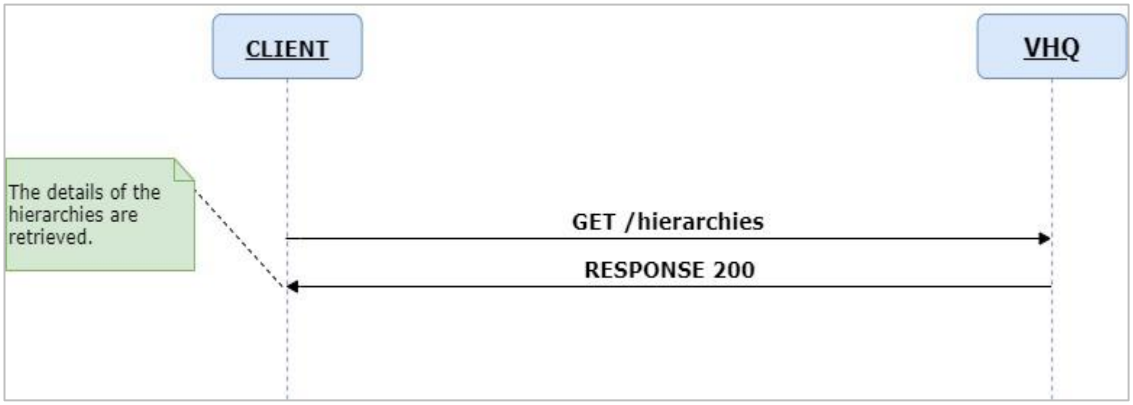
GET HIERARCHIES

The API gets the immediate child hierarchies of the given parent hierarchy.

Go to the following path:

```
API - GET/hierarchies
URL - https://<API Server>/apis/v1.3/hierarchies
e.g. https://vhqtest.verifone.com/apis/v1.3/hierarchies
```

Sequence Diagram



Input Parameters

Element	Optional/Mandatory	Data Type	Description
parentId	Mandatory	Int	Hierarchy Parent Id. If ‘0’ is passed as an input, then the hierarchy assigned to the user (User details present in token) will be considered and returns the child hierarchies of the user hierarchy.

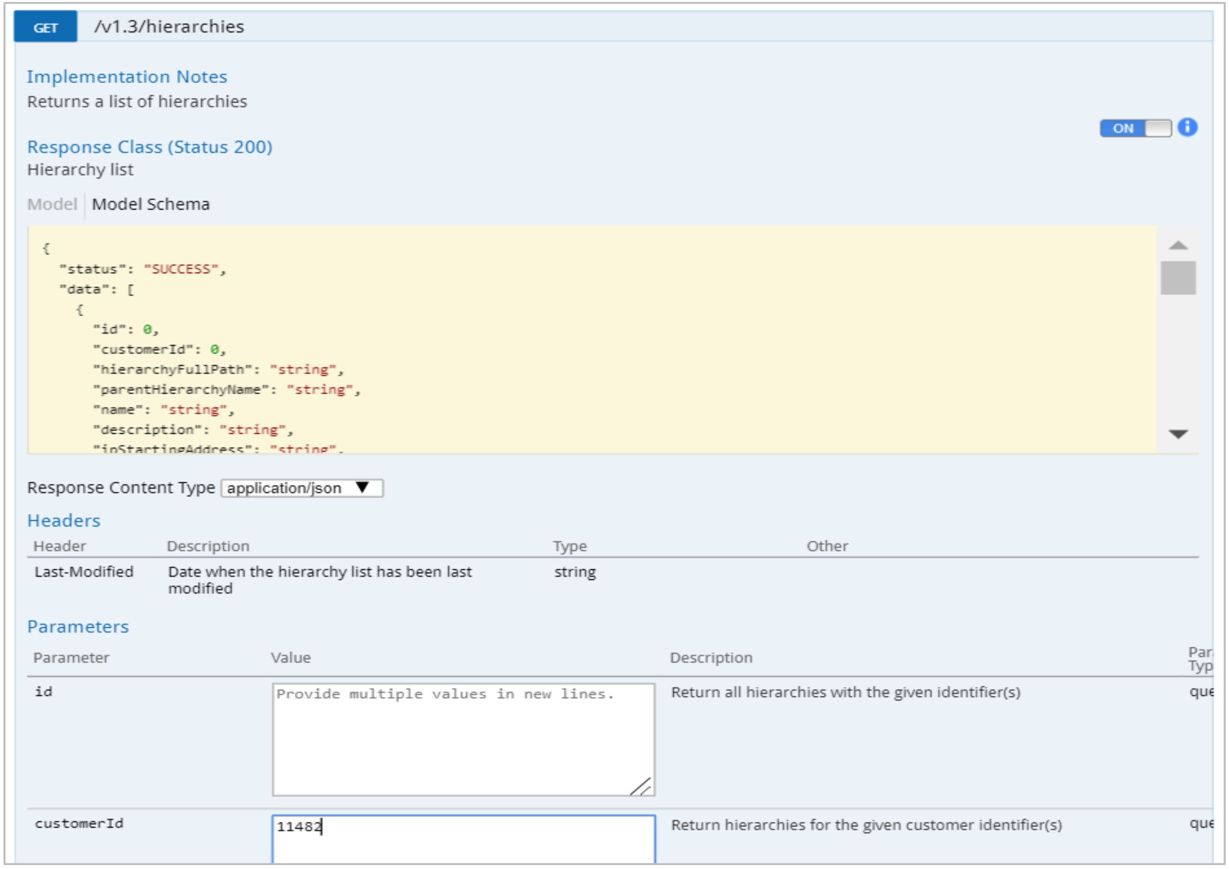
Output Parameters

Element	Attribute	Data Type	Description
hierarchies (collection of Group)	id	Int	Id of the hierarchy.
	name	String	Name of the hierarchy.

Get Hierarchies API - Sample Code

```
SAMPLE CODE SNIPPET
public class GetHierarchies {

       private static String url = "http://<servername:port>/apis/v1.3/";
       private int statusCode;
   // This is the base URL, it will be appended as per the entity which is invoked

       /**
        * @param completeURL -> This is the URL which appended as per the
       entity which is invoked
        * @param method -> This refer to which method we are going to
       send e.g. get,post,put
        * @return -> httpCon with Content Type and Authorization token
        * @throws IOException
        */
       HttpURLConnection getConnection(URL completeURL, String method)
               throws IOException {
           HttpURLConnection httpCon = (HttpURLConnection)
                   completeURL.openConnection();
           httpCon.setDoOutput(true);
        httpCon.setRequestProperty("Authorization", "Bearer 38f3f1ac-41ce-387d-8407-048b2b3c48b3");
                httpCon.setRequestMethod(method);
        httpCon.setRequestProperty("Content-Type", "application/json");
        return httpCon;
    }
    /**
     * This method get all data of a entity
     * @param Api -> This refers to the endpoint of the URL for the
    particular entity e.g. for Hierarchy entity it is hierarchies
     * @param id -> This refer to show specific data from Entity based
    on Id.
     * @param fields -> This refers to show specific fields, which we
     pass as a parameter.
     * @param queryparameter
     * @return response from entity
     */
    String getHierarchies(String Api, int id,String fullPath,String[] queryparameter,String queryParamValue) {
       StringBuffer response=null;
       if(id!=0) {
           url = url.concat(Api) + "/" + id;
       }else {
          url = url.concat(Api);
       }

        if(fullPath!=null){
            url=url+"/".concat(fullPath);
        }
        if(null!=queryparameter){
            for(String fields: queryparameter){
                url = url +"?"+fields+"="+queryParamValue;
                System.out.println("URL :::: "+url);
            }
        }
        try {
            URL completeURL = new URL(url);
// Getting All Header Information
// like(contentType,CustomerName,CustomerId)
            HttpURLConnection httpCon = getConnection(completeURL, "GET");
            httpCon.getResponseCode();
            BufferedReader in = new BufferedReader(new InputStreamReader(httpCon.getInputStream()));
            String inputLine;
            response =new StringBuffer();
            while ((inputLine = in.readLine()) != null) {
                response.append(inputLine);
            }
            System.out.println(response);
            System.out.println(httpCon.getResponseCode());
            System.out.println(httpCon.getResponseMessage());
        } catch (Exception ex) {
            ex.printStackTrace();
            if (ex.getMessage().equalsIgnoreCase("404 Not Found")) {
                statusCode = 404;
                throw new RuntimeException("Not Found " + statusCode);

            } else if (ex.getMessage().equalsIgnoreCase("400 Bad Request")) {
                    statusCode = 400;
            throw new RuntimeException("Bad Request " + statusCode);
        } else if (ex.getMessage().equalsIgnoreCase("409 Conflict"))
        {
            statusCode = 409;
            throw new RuntimeException("Already Exist : Conflict " +
                    statusCode);
        } else {
            statusCode = 500;
            throw new RuntimeException("Already Exist : Conflict " +
                    statusCode);
        }
    }
return response.toString();
}
    public static void main(String[] args) {
        String [] queryparameter= {"customerId"};
        String queryParamValue = "11482";
        new GetHierarchies().getHierarchies("hierarchies",0,null,queryparameter,queryParamValue);
    }
}
```

Sample using Swagger - Request



/v1.3/hierarchies

Implementation Notes

Returns a list of hierarchies

Response Class (Status 200)

Hierarchy list

Model | Model Schema

```
{
  "status": "SUCCESS",
  "data": [
    {
      "id": 0,
      "customerId": 0,
      "hierarchyFullPath": "string",
      "parentHierarchyName": "string",
      "name": "string",
      "description": "string",
      "ipStartingAddress": "string",
```

Response Content Type application/json ▼

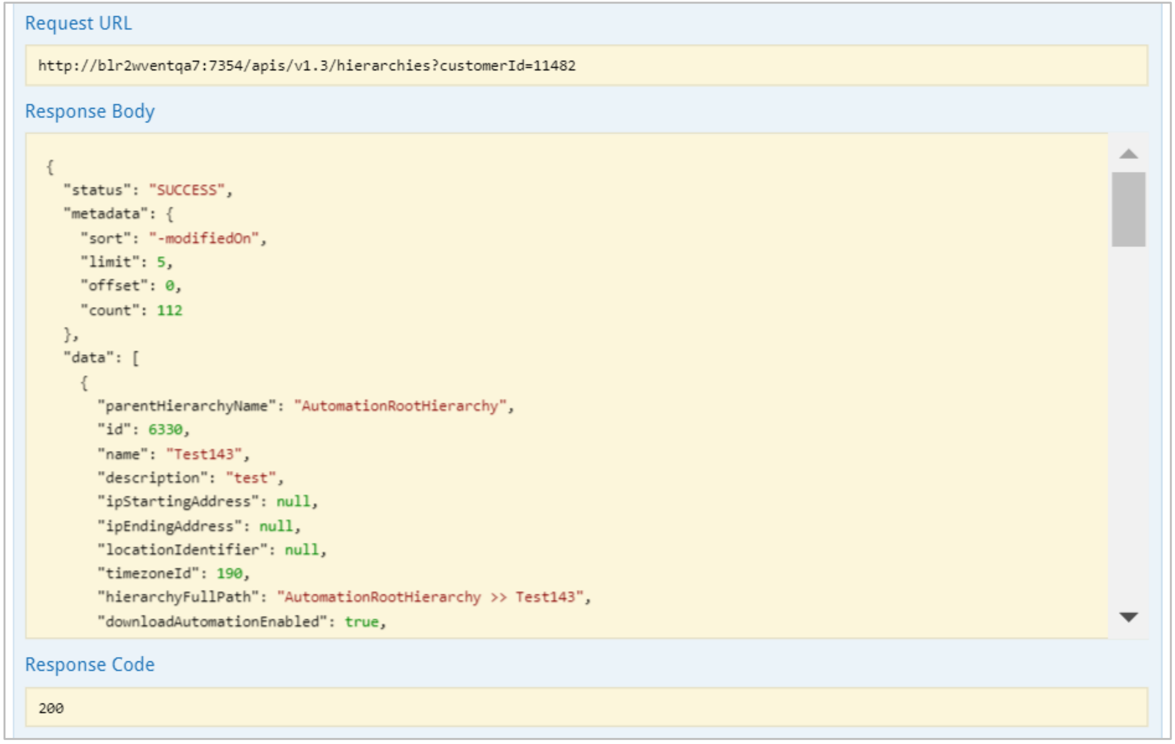
Headers

Header	Description	Type	Other
Last-Modified	Date when the hierarchy list has been last modified	string	

Parameters

Parameter	Value	Description	Par Typ
id	Provide multiple values in new lines.	Return all hierarchies with the given identifier(s)	que
customerId	11482	Return hierarchies for the given customer identifier(s)	que

Sample using Swagger - Response



Request URL

```
http://blr2wventqa7:7354/apis/v1.3/hierarchies?customerId=11482
```

Response Body

```
{
  "status": "SUCCESS",
  "metadata": {
    "sort": "-modifiedOn",
    "limit": 5,
    "offset": 0,
    "count": 112
  },
  "data": [
    {
      "parentHierarchyName": "AutomationRootHierarchy",
      "id": 6330,
      "name": "Test143",
      "description": "test",
      "ipStartingAddress": null,
      "ipEndingAddress": null,
      "locationIdentifier": null,
      "timezoneId": 190,
      "hierarchyFullPath": "AutomationRootHierarchy >> Test143",
      "downloadAutomationEnabled": true,
```

Response Code

```
200
```

Error Codes

HTTP Status Code	Application Code	Name	Severity	Message
201	S201	S_CREATED	-	Success.
401	E101	E_TOKEN_INVALID_OR_EXPIRED	2	Token is expired or invalid.
401	E102	E_UNAUTHORIZED_ACCESS	2	The user does not have the right to perform the operation.
500	E103	E_GENERAL_ERROR	3	API failed.
500	E130	E_HIEARCHIES_NOT_FOUND	3	Hierarchies not found.

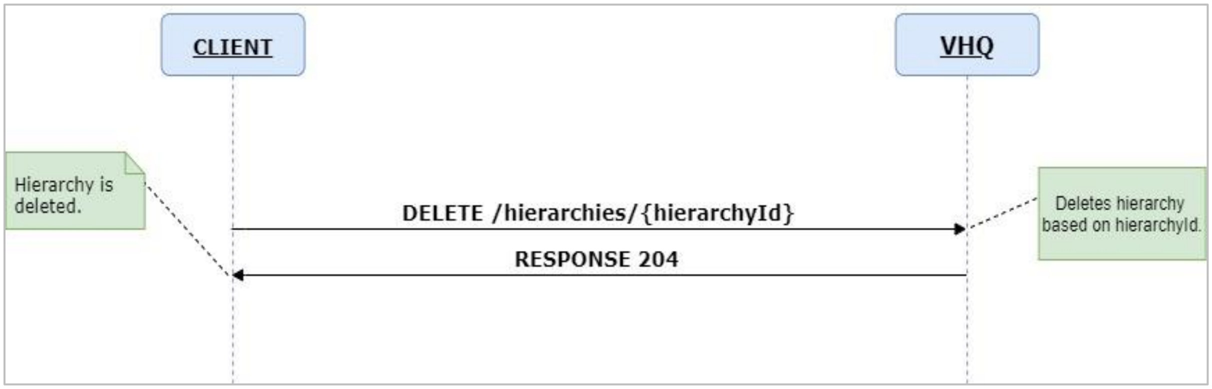
DELETE HIERARCHY

To delete the hierarchy of a specified hierarchy ID. If there are devices in the hierarchy, the hierarchy will not get deleted, but it might give an appropriate response code/message.

Go to the following path:

```
API - GET/hierarchies/{hierarchyId}
URL - https://<API Server>/apis/v1.1/hierarchies
e.g. https://vhqtest.verifone.com/apis/v1.1/hierarchies
```

Sequence Diagram


CLIENT
VHQ
Hierarchy is deleted.
DELETE /hierarchies/{hierarchyId}
RESPONSE 204
Deletes hierarchy based on hierarchyId.

Input Parameters

Input parameters must be part of the query string.

Element	Optional/Mandatory	Data Type	Description
hierarchyId	Mandatory	Int	Hierarchy identifier.

Output Parameters

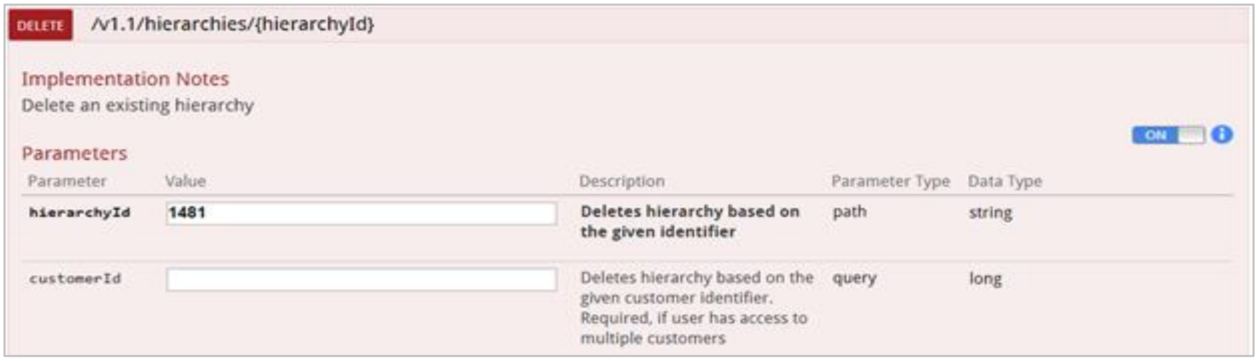
None

Delete Hierarchy API - Sample Code

SAMPLE CODE SNIPPET

```
public class DeleteHierarchy {
        private static String url = "http://<servername:port>/apis/v1.1/";
        //http://<servername:port>/apis/v1.1/hierarchies/6234?customerId=1473
        // This is the base URL, it will be appended as per the entity which is invoked
        /**
         * @param Api -> This refers to the endpoint of the URL for the particular entity e.g. for Hierarchy entity it is
hierarchies
         * @return
         */
        HttpHeaders getHeader(String Api) {
            HttpHeaders headers = new HttpHeaders();
            headers.set("Authorization", "Bearer 1c58de5b-7033-3cec-a2c4-99d2668f95c9");
            headers.setContentType(MediaType.APPLICATION_JSON);
            return headers;
        }
        /**
         * @param Api -> This refers to the endpoint of the URL for the
        particular entity e.g. for Hierarchy entity it is hierarchies
         * @param body -> The request body which is required to be patched,
        This changes as per the entity
         * @param id -> This refer to which specific Row need to change
         */
        void deleteHierarchy(String api, String body, int id,String[] queryParameter,String[] queryparamvalue) {
            String BASE_PATH = url.concat(api) + "/" + id;
            //Getting All Header Information
            //like(contentType,CustomerName,CustomerId)
            url=BASE_PATH+"?";
            if(null!=queryParameter && null!=queryparamvalue){
                int value=0;
                url = url  + queryParameter[value]+ "=" + queryparamvalue[value] ;
            }
            System.out.println(url);
            HttpHeaders headers = getHeader(api);
            int statusCode;
            try {
                HttpEntity<String> entity = new HttpEntity<String>(body, headers);
                RestTemplate = new RestTemplate();
                HttpComponentsClientHttpRequestFactory requestFactory = new HttpComponentsClientHttpRequestFactory();
                requestFactory.setConnectTimeout(18000);
                requestFactory.setReadTimeout(18000);
                restTemplate.setRequestFactory(requestFactory);
                //HttpEntity<String> result = restTemplate.exchange(BASE_PATH, HttpMethod.PATCH, entity, String.class);
                ResponseEntity<String> responseEntity = restTemplate.exchange(url, HttpMethod.PATCH, entity,
String.class);
                statusCode =  responseEntity.getStatusCode().value();
                System.out.println(statusCode);
                System.out.println(responseEntity.getBody());
            } catch (Exception ex) {
                ex.printStackTrace();
                if (ex.getMessage().equalsIgnoreCase("404 Not Found")) {
                    statusCode = 404;
                    throw new RuntimeException("Not Found " + statusCode);
                } else if (ex.getMessage().equalsIgnoreCase("400 Bad Request")) {
                    statusCode = 400;
                    throw new RuntimeException("Bad Request " + statusCode);
                } else if (ex.getMessage().equalsIgnoreCase("409 Conflict")) {
                    statusCode = 409;
                    throw new RuntimeException("Already Exist : Conflict " + statusCode);
                } else {
                    statusCode = 500;
                    throw new RuntimeException("Already Exist : Conflict " + statusCode);
                }
            }
            System.out.println("Hierarchy Deleted!");
        }
        public static void main(String[] args) {
            String [] queryParameter= {"customerId"};
            String[] queryparamvalue= {"11482"};
            new DeleteHierarchy().deleteHierarchy("hierarchies",null, 6235,queryParameter,queryparamvalue);
        }
    }
```

Sample using Swagger – Request



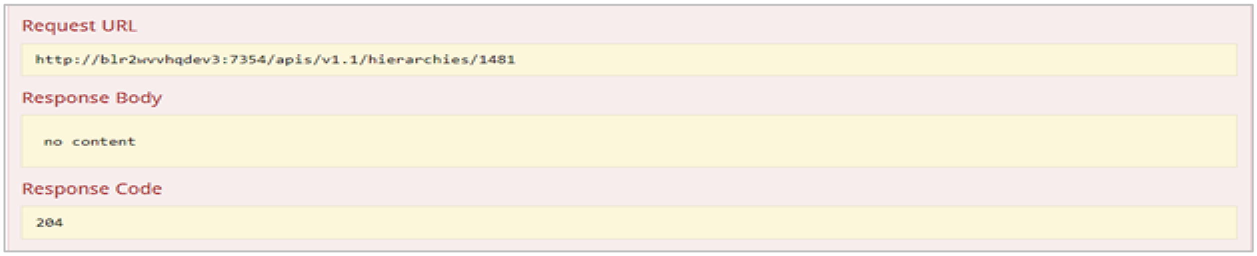
Implementation Notes

Delete an existing hierarchy

Parameters

Parameter	Value	Description	Parameter Type	Data Type
hierarchyId	1481	**Deletes hierarchy based on the given identifier**	path	string
customerId		Deletes hierarchy based on the given customer identifier. Required, if user has access to multiple customers	query	long

Sample using Swagger – Response



Request URL

```
http://blr2wvvhqdev3:7354/apis/v1.1/hierarchies/1481
```

Response Body

```
no content
```

Response Code

```
204
```

Error Codes

HTTP Status Code	Application Code	Name	Severity	Message
204	S204	S_DELETED	-	Success.
400	E100	E_DEVICE_EXIST_IN_HIERARCHY	3	Hierarchy cannot be deleted.
401	E101	E_TOKEN_INVALID_OR_EXPIRED	2	Token expired or it is invalid.
401	E102	E_UNAUTHORIZED_ACCESS	2	The user does not have the right to perform the operation.
500	E103	E_GENERAL_ERROR	3	API failed.

DOWNLOAD MANAGEMENT APIs

The download feature allows you to schedule the download of multiple packages onto multiple devices.

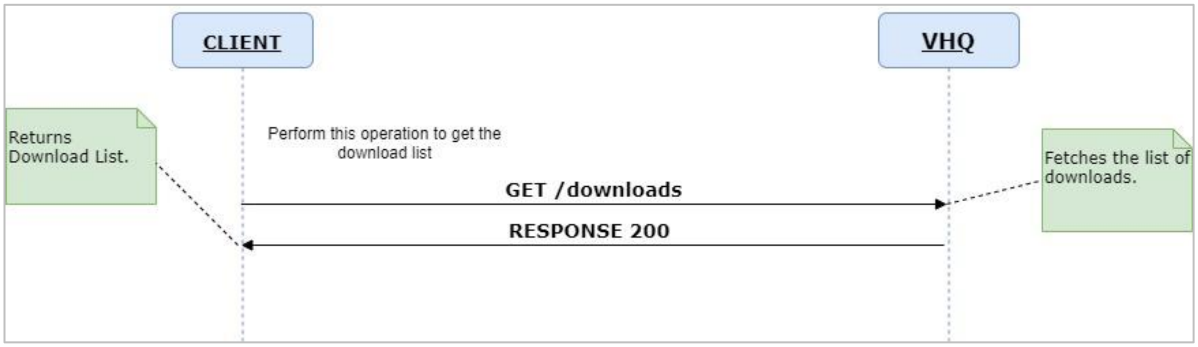
GET DOWNLOADS

This API is used to get the status of all the downloads. Scheduling the download of multiple packages onto multiple devices results in multiple download tasks. This API returns the download information for each download task that indicates the status of the download by device and package.

Go to the following path:

```
API - GET/downloads
URL- https://<APIServer>/apis/v1.3/downloads
e.g. https://vhqtest.verifone.com/apis/v1.3/downloads
```

Sequence Diagram


CLIENT
VHQ
Returns Download List.
Perform this operation to get the download list
GET /downloads
RESPONSE 200
Fetches the list of downloads.

Input Parameters

Element	Optional/Mandatory	Data Type	Description
packageId/packageName	Optional	int/String	Package Identifier/package name.

Output Parameters

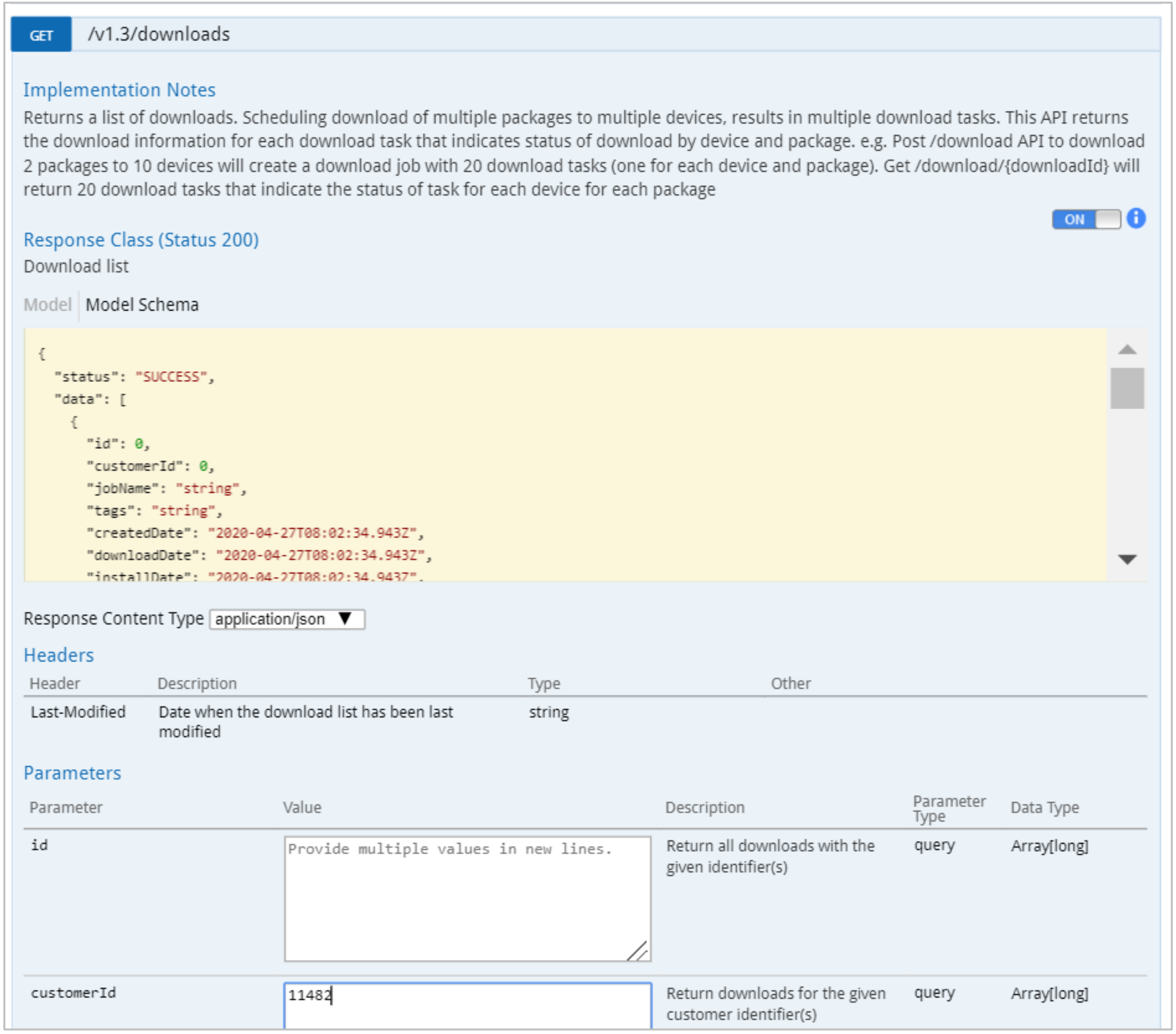
Element	Attribute	Data Type	Description
downloads (collection of downloads)	id	int	Download Identifier.
	taskStatus	array	Details of the task.

Get Downloads API - Sample Code

SAMPLE CODE SNIPPET

```
public class GetDownloads {
   private static String url = "http://<servername:port>/apis/v1.3/";
   private int statusCode;
   // This is the base URL, it will be appended as per the entity which is invoked
   /**
    * @param completeURL -> This is the URL which appended as per the
   entity which is invoked
    * @param method -> This refer to which method we are going to
   send e.g. get,post,put
    * @return -> httpCon with Content Type and Authorization token
    * @throws IOException
    */
   HttpURLConnection getConnection(URL completeURL, String method)
         throws IOException {
      HttpURLConnection httpCon = (HttpURLConnection)
            completeURL.openConnection();
      httpCon.setDoOutput(true);
      httpCon.setRequestProperty("Authorization", "Bearer 8c10910f-9228-37f7-a1d3-d73883ea2615");
      httpCon.setRequestMethod(method);
      httpCon.setRequestProperty("Content-Type", "application/json");
      return httpCon;
   }
   /**
    * This method get all data of a entity
    * @param Api -> This refers to the endpoint of the URL for the
   particular entity e.g. for Parameter entity it is parameters
    * @param id -> This refer to show specific data from Entity based
   on Id.
    * @param fields -> This refers to show specific fields, which we
   pass as a parameter.
    * @param queryparameter
    * @return response from entity
    */
   String getDownloads(String Api, int id,String[] queryparameter,String queryParamValue) {
      StringBuffer response=null;
      if(id!=0) {
         url = url.concat(Api) + "/" + id;
      }else {
         url = url.concat(Api);
      }

      String fullPath = null;
      if(fullPath!=null){
         url=url+"/".concat(fullPath);
      }
      if(null!=queryparameter){
         for(String fields: queryparameter){
            url = url +"?"+fields+"="+queryParamValue;
            System.out.println("URL :::: "+url);
         }
      }
      try {
         URL completeURL = new URL(url);
         // Getting All Header Information
         // like(contentType,CustomerName,CustomerId)
         HttpURLConnection httpCon = getConnection(completeURL, "GET");
         httpCon.getResponseCode();
         BufferedReader in = new BufferedReader(new InputStreamReader(httpCon.getInputStream()));
         String inputLine;
         response =new StringBuffer();
         while ((inputLine = in.readLine()) != null) {
            response.append(inputLine);
         }
         System.out.println(response);
         System.out.println(httpCon.getResponseCode());
         System.out.println(httpCon.getResponseMessage());
      } catch (Exception ex) {
         ex.printStackTrace();
         if (ex.getMessage().equalsIgnoreCase("404 Not Found")) {
            statusCode = 404;
            throw new RuntimeException("Not Found " + statusCode);

         } else if (ex.getMessage().equalsIgnoreCase("400 Bad Request")) {
            statusCode = 400;
            throw new RuntimeException("Bad Request " + statusCode);
         } else if (ex.getMessage().equalsIgnoreCase("409 Conflict"))
         {
            statusCode = 409;
            throw new RuntimeException("Already Exist : Conflict " +
                  statusCode);
         } else {
            statusCode = 500;
            throw new RuntimeException("Already Exist : Conflict " +
                  statusCode);
         }
      }
      return response.toString();
   }
   public static void main(String[] args) {
      String [] queryparameter= {"customerId"};
      String queryParamValue = "11482";
      new GetDownloads().getDownloads("downloads",0,queryparameter,queryParamValue);
   }
}
```

Sample using Swagger - Request



/v1.3/downloads

Implementation Notes

Returns a list of downloads. Scheduling download of multiple packages to multiple devices, results in multiple download tasks. This API returns the download information for each download task that indicates status of download by device and package. e.g. Post /download API to download 2 packages to 10 devices will create a download job with 20 download tasks (one for each device and package). Get /download/{downloadId} will return 20 download tasks that indicate the status of task for each device for each package

Response Class (Status 200)

Download list

```
{
  "status": "SUCCESS",
  "data": [
    {
      "id": 0,
      "customerId": 0,
      "jobName": "string",
      "tags": "string",
      "createdDate": "2020-04-27T08:02:34.943Z",
      "downloadDate": "2020-04-27T08:02:34.943Z",
      "installDate": "2020-04-27T08:02:34.943Z",
```

Response Content Type application/json ▼

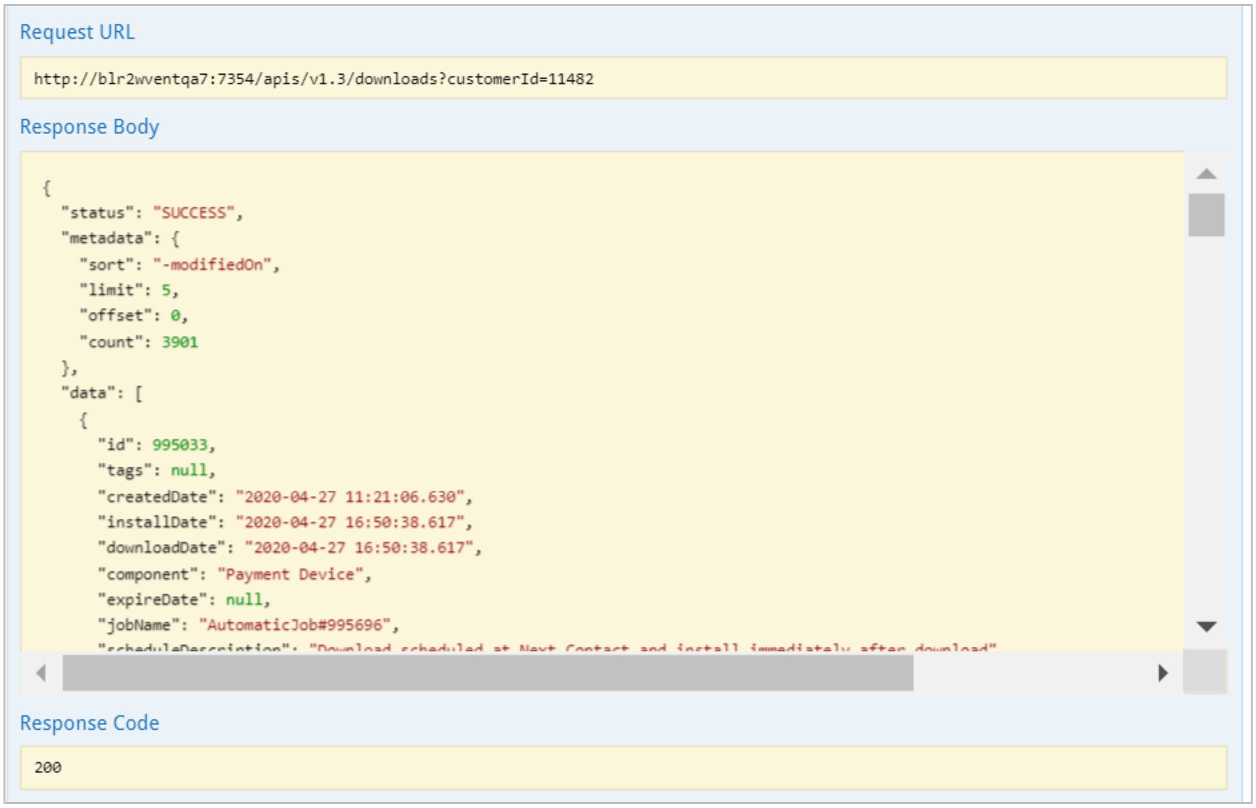
Headers

Header	Description	Type	Other
Last-Modified	Date when the download list has been last modified	string	

Parameters

Parameter	Value	Description	Parameter Type	Data Type
id	Provide multiple values in new lines.	Return all downloads with the given identifier(s)	query	Array[long]
customerId	11482	Return downloads for the given customer identifier(s)	query	Array[long]

Sample using Swagger – Response



Request URL

```
http://blr2wventqa7:7354/apis/v1.3/downloads?customerId=11482
```

Response Body

```
{
  "status": "SUCCESS",
  "metadata": {
    "sort": "-modifiedOn",
    "limit": 5,
    "offset": 0,
    "count": 3901
  },
  "data": [
    {
      "id": 995033,
      "tags": null,
      "createdDate": "2020-04-27 11:21:06.630",
      "installDate": "2020-04-27 16:50:38.617",
      "downloadDate": "2020-04-27 16:50:38.617",
      "component": "Payment Device",
      "expireDate": null,
      "jobName": "AutomaticJob#995696",
      "scheduleDescription": "Download scheduled at Next Contact and install immediately after download"
```

Response Code

```
200
```

Error Codes

HTTP Status Code	Application Code	Name	Severity	Message
200	S100	S_OK	-	Success.
401	E101	E_TOKEN_INVALID_OR_EXPIRED	2	Token expired or it is invalid.
401	E102	E_UNAUTHORIZED_ACCESS	2	The user does not have the right to perform the operation.
500	E103	E_GENERAL_ERROR	3	API failed.

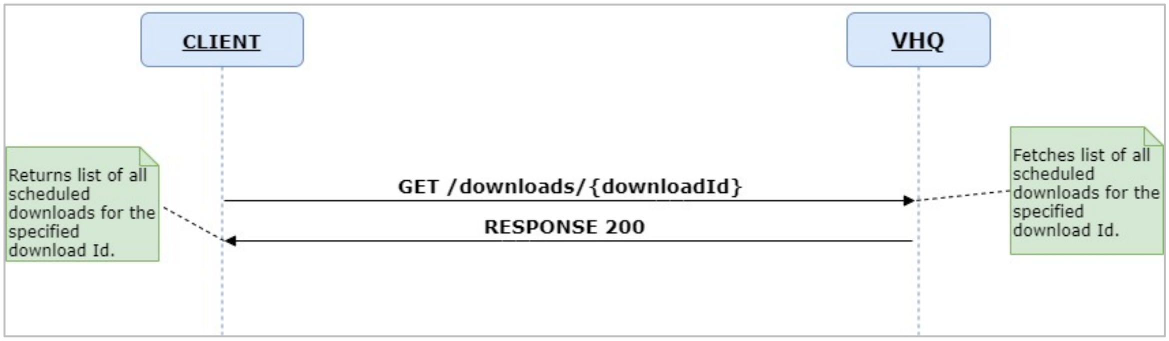
GET DOWNLOADS BY DOWNLOADID

This API is used to get the details of a particular downloadId.

Once you have the download ID, go to the following path:

```
API - GET/downloads/{downloadId}
URL - https://<API Server>/apis/v1.3/downloads/{downloadId}
e.g. https://vhqtest.verifone.com/apis/v1.3/downloads/{downloadId}
```

Sequence Diagram


CLIENT
VHQ
Returns list of all scheduled downloads for the specified download Id.
GET /downloads/{downloadId}
RESPONSE 200
Fetches list of all scheduled downloads for the specified download Id.

Input Parameters

Element	Optional/Mandatory	Data Type	Description
downloadId	Mandatory	Int	Unique device identifier to be set as the path of the URI.

Output Parameters

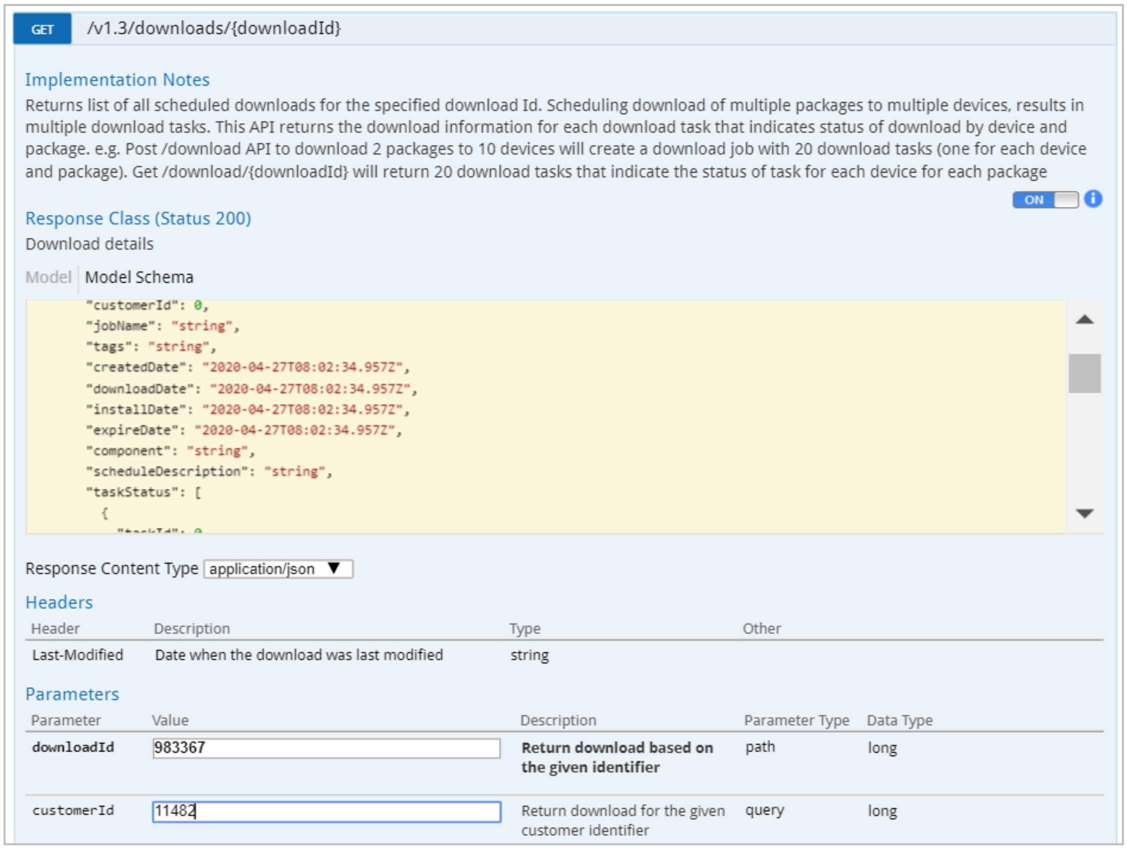
Element	Attribute	Data Type	Description
downloads (collection of downloads)	id	int	Download Identifier.
	taskStatus	array	Details of the task.

Get Downloads by DownloadId API - Sample Code

SAMPLE CODE SNIPPET

```
public class GetDownloadsById {
   private static String url = "http://<servername:port>/apis/v1.3/";
   private int statusCode;
   // This is the base URL, it will be appended as per the entity which is invoked
   /**
    * @param completeURL -> This is the URL which appended as per the
   entity which is invoked
    * @param method -> This refer to which method we are going to
   send e.g. get,post,put
    * @return -> httpCon with Content Type and Authorization token
    * @throws IOException
    */
   HttpURLConnection getConnection(URL completeURL, String method)
         throws IOException {
      HttpURLConnection httpCon = (HttpURLConnection)
            completeURL.openConnection();
      httpCon.setDoOutput(true);
      httpCon.setRequestProperty("Authorization", "Bearer dec38556-c51a-3702-82e9-b97757930023");
      httpCon.setRequestMethod(method);
      httpCon.setRequestProperty("Content-Type", "application/json");
      return httpCon;
   }
   /**
    * This method get all data of an entity
    * @param Api -> This refers to the endpoint of the URL for the
   particular entity e.g. for Download entity it is downloads
    * @param id -> This refer to show specific data from Entity based
   on Id.
    * @param fields -> This refers to show specific fields, which we
   pass as a parameter.
    * @param queryparameter
    * @return response from entity
    */
   String getById(String Api, int id,String fullPath,String[] queryparameter,String queryParamValue) {
      StringBuffer response=null;
      if(id!=0) {
         url = url.concat(Api) + "/" + id;
      }else {
         url = url.concat(Api);
      }
      if(fullPath!=null){
         url=url+"/".concat(fullPath);
      }
      if(null!=queryparameter){
         for(String fields: queryparameter){
            url = url +"?"+fields+"="+queryParamValue;
            System.out.println("URL :::: "+url);
         }
      }
      try {
         URL completeURL = new URL(url);
         // Getting All Header Information
         // like(contentType,CustomerName,CustomerId)
         HttpURLConnection httpCon = getConnection(completeURL, "GET");
         httpCon.getResponseCode();
         BufferedReader in = new BufferedReader(new InputStreamReader(httpCon.getInputStream()));
         String inputLine;
         response =new StringBuffer();
         while ((inputLine = in.readLine()) != null) {
            response.append(inputLine);
         }
         System.out.println(response);
         System.out.println(httpCon.getResponseCode());
         System.out.println(httpCon.getResponseMessage());
      } catch (Exception ex) {
         ex.printStackTrace();
         if (ex.getMessage().equalsIgnoreCase("404 Not Found")) {
            statusCode = 404;
            throw new RuntimeException("Not Found " + statusCode);

         } else if (ex.getMessage().equalsIgnoreCase("400 Bad Request")) {
            statusCode = 400;
            throw new RuntimeException("Bad Request " + statusCode);
         } else if (ex.getMessage().equalsIgnoreCase("409 Conflict"))
         {
            statusCode = 409;
            throw new RuntimeException("Already Exist : Conflict " +
                  statusCode);
         } else {
            statusCode = 500;
            throw new RuntimeException("Already Exist : Conflict " +
                  statusCode);
         }
      }
      return response.toString();
   }
   public static void main(String[] args) {
      String [] queryparameter= {"customerId"};
      String queryParamValue = "11482";
      new GetDownloadsById().getById("downloads",983363,null,queryparameter,queryParamValue);
   }
}
```

Sample using Swagger – Request



GET /v1.3/downloads/{downloadId}

Implementation Notes

Returns list of all scheduled downloads for the specified download Id. Scheduling download of multiple packages to multiple devices, results in multiple download tasks. This API returns the download information for each download task that indicates status of download by device and package. e.g. Post /download API to download 2 packages to 10 devices will create a download job with 20 download tasks (one for each device and package). Get /download/{downloadId} will return 20 download tasks that indicate the status of task for each device for each package

ON

Response Class (Status 200)

Download details

Model | Model Schema

```
      "customerId": 0,
      "jobName": "string",
      "tags": "string",
      "createdDate": "2020-04-27T08:02:34.957Z",
      "downloadDate": "2020-04-27T08:02:34.957Z",
      "installDate": "2020-04-27T08:02:34.957Z",
      "expireDate": "2020-04-27T08:02:34.957Z",
      "component": "string",
      "scheduleDescription": "string",
      "taskStatus": [
        {
          "taskId": 0
```

Response Content Type: application/json

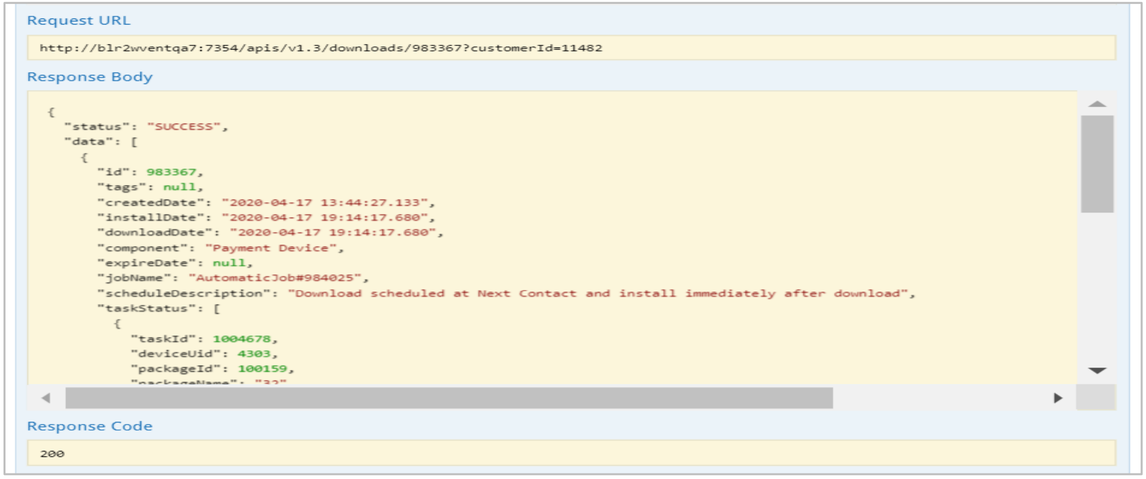
Headers

Header	Description	Type	Other
Last-Modified	Date when the download was last modified	string	

Parameters

Parameter	Value	Description	Parameter Type	Data Type
downloadId	983367	**Return download based on the given identifier**	path	long
customerId	11482	Return download for the given customer identifier	query	long

Sample using Swagger - Response



Request URL

```
http://blr2wventqa7:7354/apis/v1.3/downloads/983367?customerId=11482
```

Response Body

```
{
  "status": "SUCCESS",
  "data": [
    {
      "id": 983367,
      "tags": null,
      "createdDate": "2020-04-17 13:44:27.133",
      "installDate": "2020-04-17 19:14:17.680",
      "downloadDate": "2020-04-17 19:14:17.680",
      "component": "Payment Device",
      "expireDate": null,
      "jobName": "AutomaticJob#984025",
      "scheduleDescription": "Download scheduled at Next Contact and install immediately after download",
      "taskStatus": [
        {
          "taskId": 1004678,
          "deviceUid": 4303,
          "packageId": 100159,
          "packageName": "32"
```

Response Code

```
200
```

Error Codes

HTTP Status Code	Application Code	Name	Severity	Message
200	S100	S_OK	-	Success.

401	E101	E_TOKEN_INVALID_OR_EXPIRED	2	The token is expired or it is invalid.
401	E102	E_UNAUTHORIZED_ACCESS	2	The user does not have the right to perform the operation.
500	E103	E_GENERAL_ERROR	3	API failed.

GET DOWNLOADS BY DEVICEID

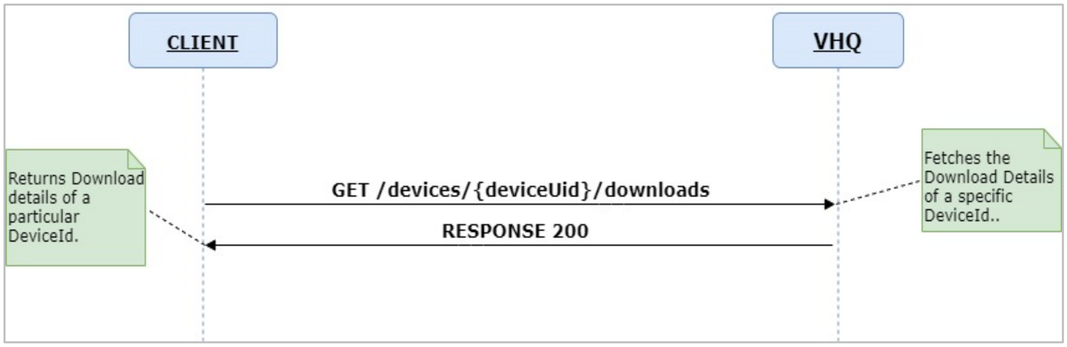
To get the download details for a specified device ID.

As a pre-requisite for calling get downloads, *Get Device* API is used to get the **deviceUid**. Refer to the *Get Device* section for steps.

Once you have the device Uid, go to the following path:

```
API - GET/devices/{deviceUid}/downloads
URL - https://<API Server>/apis/v1.3/devices/{deviceUid}/downloads
e.g. https://vhqtest.verifone.com/apis/v1.3/devices/{deviceUid}/downloads
```

Sequence Diagram

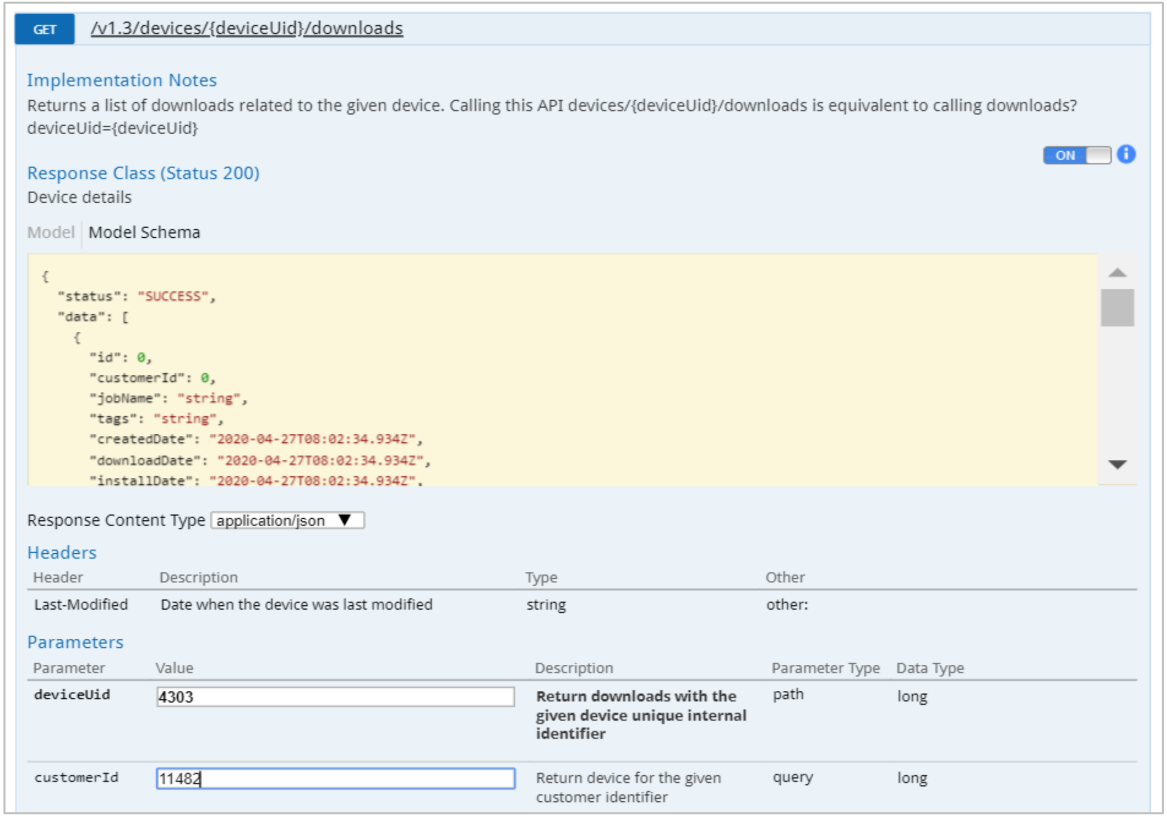

CLIENT
VHQ
Returns Download details of a particular DeviceId.
GET /devices/{deviceUid}/downloads
RESPONSE 200
Fetches the Download Details of a specific DeviceId..

Get Downloads by DeviceId API - Sample Code

SAMPLE CODE SNIPPET

```
public class GetDownloadByDevice {
   private static String url = "http://<servername:port>/apis/v1.3/";
   private int statusCode;
   private static String fullPath="downloads";
   // This is the base URL, it will be appended as per the entity which is invoked
   /**
    * @param completeURL -> This is the URL which appended as per the
   entity which is invoked
    * @param method -> This refer to which method we are going to
   send e.g. get,post,put
    * @return -> httpCon with Content Type and Authorization token
    * @throws IOException
    */
   HttpURLConnection getConnection(URL completeURL, String method)
         throws IOException {
      HttpURLConnection httpCon = (HttpURLConnection)
            completeURL.openConnection();
      httpCon.setDoOutput(true);
      httpCon.setRequestProperty("Authorization", "Bearer dec38556-c51a-3702-82e9-b97757930023");
      httpCon.setRequestMethod(method);
      httpCon.setRequestProperty("Content-Type", "application/json");
      return httpCon;
   }
   /**
    * This method get all data of a entity
    * @param Api -> This refers to the endpoint of the URL for the
   particular entity e.g. for Download entity it is downloads
    * @param id -> This refer to show specific data from Entity based
   on Id.
    * @param fields -> This refers to show specific fields, which we
   pass as a parameter.
    * @param queryparameter
    * @return response from entity
    */
   String getById(String Api, int id,String fullPath,String[] queryparameter,String queryParamValue) {
      StringBuffer response=null;
      if(id!=0) {
         url = url.concat(Api) + "/" + id;
      }else {
         url = url.concat(Api);
      }

      if(fullPath!=null){
         url=url+"/".concat(fullPath);
      }
      if(null!=queryparameter){
         for(String fields: queryparameter){
            url = url +"?"+fields+"="+queryParamValue;
            System.out.println("URL :::: "+url);
         }
      }
      try {
         URL completeURL = new URL(url);
         // Getting All Header Information
         // like(contentType,CustomerName,CustomerId)
         HttpURLConnection httpCon = getConnection(completeURL, "GET");
         httpCon.getResponseCode();
         BufferedReader in = new BufferedReader(new InputStreamReader(httpCon.getInputStream()));
         String inputLine;
         response =new StringBuffer();
         while ((inputLine = in.readLine()) != null) {
            response.append(inputLine);
         }
         System.out.println(response);
         System.out.println(httpCon.getResponseCode());
         System.out.println(httpCon.getResponseMessage());
      } catch (Exception ex) {
         ex.printStackTrace();
         if (ex.getMessage().equalsIgnoreCase("404 Not Found")) {
            statusCode = 404;
            throw new RuntimeException("Not Found " + statusCode);

         } else if (ex.getMessage().equalsIgnoreCase("400 Bad Request")) {
            statusCode = 400;
            throw new RuntimeException("Bad Request " + statusCode);
         } else if (ex.getMessage().equalsIgnoreCase("409 Conflict"))
         {
            statusCode = 409;
            throw new RuntimeException("Already Exist : Conflict " +
                  statusCode);
         } else {
            statusCode = 500;
            throw new RuntimeException("Already Exist : Conflict " +
                  statusCode);
         }
      }
      return response.toString();
   }
   public static void main(String[] args) {
      String [] queryparameter= {"customerId"};
      String queryParamValue = "11482";
      new GetDownloadByDevice().getById("devices",4266,fullPath,queryparameter,queryParamValue);
   }
}
```

Sample using Swagger - Request



GET /v1.3/devices/{deviceUid}/downloads

Implementation Notes

Returns a list of downloads related to the given device. Calling this API devices/{deviceUid}/downloads is equivalent to calling downloads?deviceUid={deviceUid}

Response Class (Status 200)

Device details

Model | Model Schema

```
{
  "status": "SUCCESS",
  "data": [
    {
      "id": 0,
      "customerId": 0,
      "jobName": "string",
      "tags": "string",
      "createdDate": "2020-04-27T08:02:34.934Z",
      "downloadDate": "2020-04-27T08:02:34.934Z",
      "installDate": "2020-04-27T08:02:34.934Z",
```

Response Content Type application/json ▼

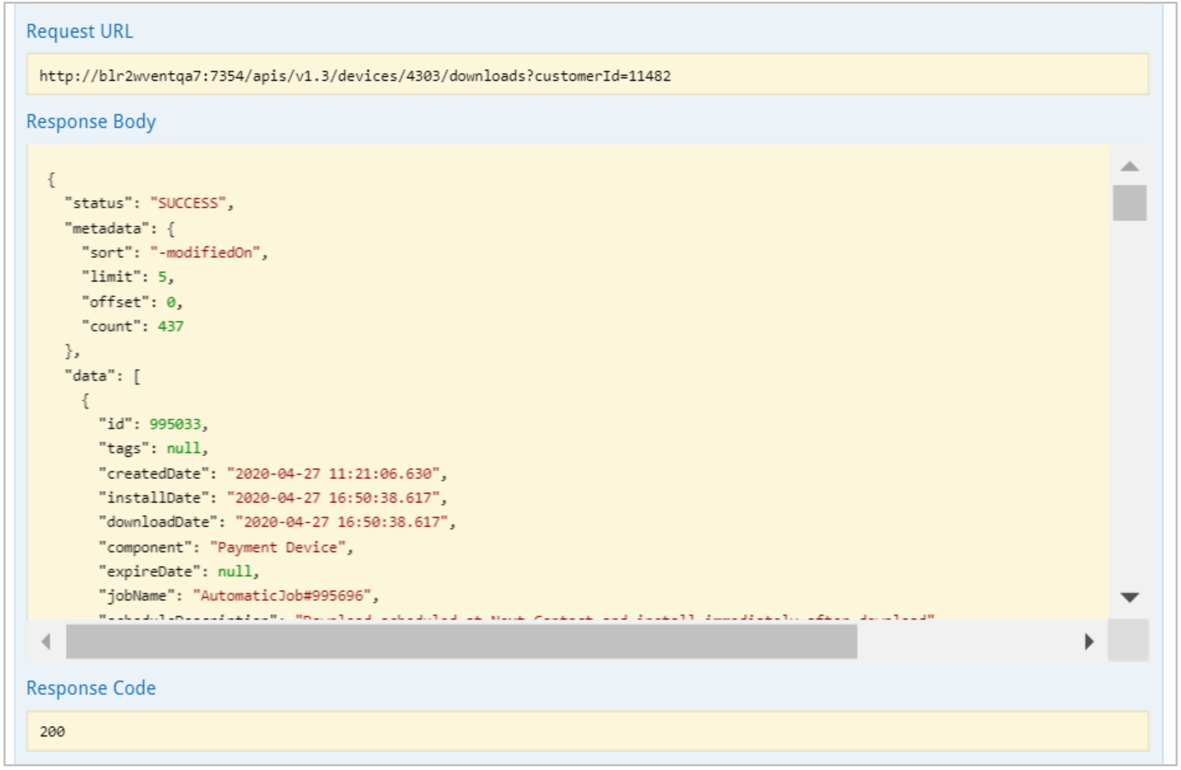
Headers

Header	Description	Type	Other
Last-Modified	Date when the device was last modified	string	other:

Parameters

Parameter	Value	Description	Parameter Type	Data Type
deviceUid	4303	**Return downloads with the given device unique internal identifier**	path	long
customerId	11482	Return device for the given customer identifier	query	long

Sample using Swagger - Response



Request URL

```
http://blr2wventqa7:7354/apis/v1.3/devices/4303/downloads?customerId=11482
```

Response Body

```
{
  "status": "SUCCESS",
  "metadata": {
    "sort": "-modifiedOn",
    "limit": 5,
    "offset": 0,
    "count": 437
  },
  "data": [
    {
      "id": 995033,
      "tags": null,
      "createdDate": "2020-04-27 11:21:06.630",
      "installDate": "2020-04-27 16:50:38.617",
      "downloadDate": "2020-04-27 16:50:38.617",
      "component": "Payment Device",
      "expireDate": null,
      "jobName": "AutomaticJob#995696",
      "scheduleDescription": "Download scheduled at Next Contact and install immediately after download"
```

Response Code

```
200
```

SOFTWARE PACKAGES MANAGEMENT APIs

The package represents the collection of one or more software applications.

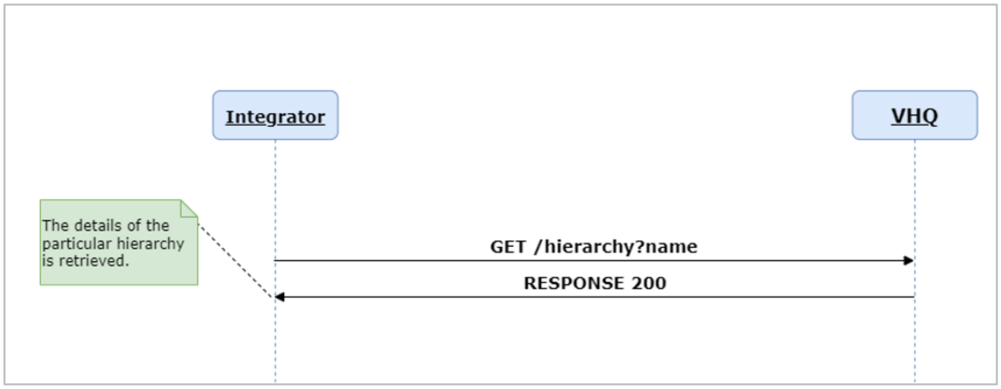
GET SOFTWARE PACKAGES-1.4

This API returns the list of all packages with details. Pagination is supported for this API.

Go to the following path:

```
API - GET/devices/packages
URL - https://<API Server>/apis/v1.4/packages
e.g.  https://vhqtest.verifone.com/apis/v1.4/packages
```

Sequence Diagram



Input Parameters

Element	Optional/Mandatory	Data Type	Description
source	Optional	String	Source contains values from PackageSources table Ex: • MarketApp • PayWare • Weaver All packages are rendered if the source is not sent in headers.
modelId/modelName	Optional	Array	Supported model identifier.
limit	Optional	Int	Rows per request.
offset	Optional	Int	Offset required to calculate pageNumber.

platform	Optional	String	Platform like: eVo or V/OS or Vx or Engage.
appId	Optional	Int	Application Id. Additional information or Tag to identify the Package. Client Application can use this to add and retrieve the package.
locationId	Optional	String	Device location identifier.

Output Parameters

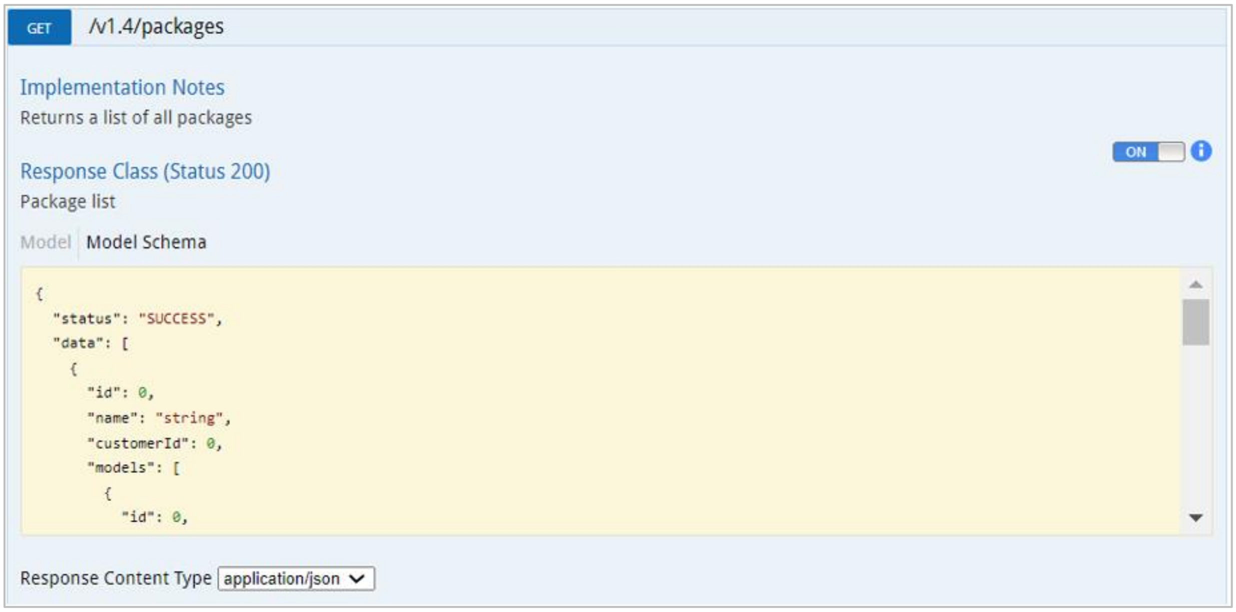
Element	Attribute	Data Type	Description
package (collection of packages)	id	Int	Id of the package.
	name	String	Name of the Package.
	version	String	Version of the Package.

Get Software Packages API - Sample Code

SAMPLE CODE SNIPPET

```
public class GetPackages {
   private static String url = "http://<servername:port>/apis/v1.4/";
   private int statusCode;
   // This is the base URL, it will be appended as per the entity which is invoked
   /**
    * @param completeURL -> This is the URL which appended as per the
   entity which is invoked
    * @param method -> This refer to which method we are going to
   send e.g. get,post,put
    * @return -> httpCon with Content Type and Authorization token
    * @throws IOException
    */
   HttpURLConnection getConnection(URL completeURL, String method)
         throws IOException {
      HttpURLConnection httpCon = (HttpURLConnection)
            completeURL.openConnection();
      httpCon.setDoOutput(true);
      httpCon.setRequestProperty("Authorization", "Bearer 4eafc052-4d05-377b-a53c-cb76e68a3d48");
      httpCon.setRequestMethod(method);
      httpCon.setRequestProperty("Content-Type", "application/json");
      return httpCon;
   }
   /**
    * This method get all data of a entity
    * @param Api -> This refers to the endpoint of the URL for the
   particular entity e.g. for software entity it is softwares
    * @param id -> This refer to show specific data from Entity based
   on Id.
    * @param fields -> This refers to show specific fields, which we
   pass as a parameter.
    * @param queryparameter
    * @return response from entity
    */
   String getPackages(String Api, int id,String fullPath,String[] queryparameter,String queryParamValue) {
      StringBuffer response=null;
      if(id!=0) {
         url = url.concat(Api) + "/" + id;
      }else {
         url = url.concat(Api);
      }
      if(fullPath!=null){
         url=url+"/".concat(fullPath);
      }
      if(null!=queryparameter){
         for(String fields: queryparameter){
            url = url +"?"+fields+"="+queryParamValue;
            System.out.println("URL :::: "+url);
         }
      }
      try {
         URL completeURL = new URL(url);
         // Getting All Header Information
         // like(contentType,CustomerName,CustomerId)
         HttpURLConnection httpCon = getConnection(completeURL, "GET");
         httpCon.getResponseCode();
         BufferedReader in = new BufferedReader(new InputStreamReader(httpCon.getInputStream()));
         String inputLine;
         response =new StringBuffer();
         while ((inputLine = in.readLine()) != null) {
            response.append(inputLine);
         }
         System.out.println(response);
         System.out.println(httpCon.getResponseCode());
         System.out.println(httpCon.getResponseMessage());
      } catch (Exception ex) {
         ex.printStackTrace();
         if (ex.getMessage().equalsIgnoreCase("404 Not Found")) {
            statusCode = 404;
            throw new RuntimeException("Not Found " + statusCode);

         } else if (ex.getMessage().equalsIgnoreCase("400 Bad Request")) {
            statusCode = 400;
            throw new RuntimeException("Bad Request " + statusCode);
         } else if (ex.getMessage().equalsIgnoreCase("409 Conflict"))
         {
            statusCode = 409;
            throw new RuntimeException("Already Exist : Conflict " +
                  statusCode);
         } else {
            statusCode = 500;
            throw new RuntimeException("Already Exist : Conflict " +
                  statusCode);
         }
      }
      return response.toString();
   }
   public static void main(String[] args) {
      String [] queryparameter= {"customerId"};
      String queryParamValue = "11482";
      new GetPackages().getPackages("packages",0,null,queryparameter,queryParamValue);
   }
}
```

Sample using Swagger – Request



/v1.4/packages

Implementation Notes

Returns a list of all packages

Response Class (Status 200)

Package list

```
{
  "status": "SUCCESS",
  "data": [
    {
      "id": 0,
      "name": "string",
      "customerId": 0,
      "models": [
        {
          "id": 0,
```

Response Content Type application/json

Error Codes

HTTP Status Code	Application Code	Name	Severity	Message
200	S100	S_OK	-	Success.
401	E101	E_TOKEN_INVALID_OR_EXPIRED	2	Token expired or it is invalid.
401	E102	E_UNAUTHORIZED_ACCESS	2	The user does not have the right to perform the operation.
500	E103	E_GENERAL_ERROR	3	API failed.

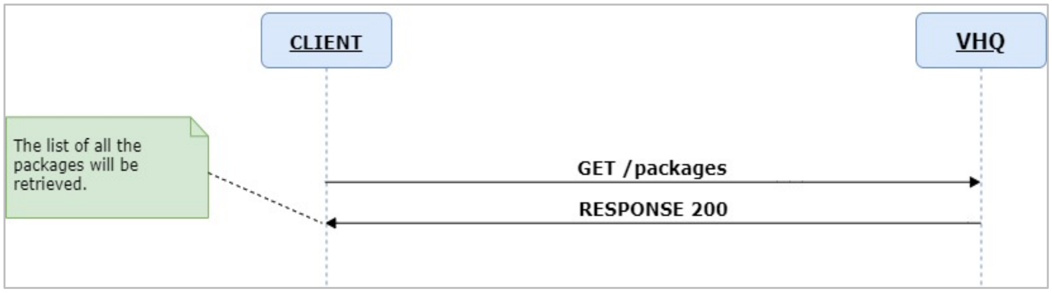
GET SOFTWARE PACKAGES

This API returns the list of all the packages. The API supports pagination.

Go to the following path:

```
API - GET/devices/packages
URL - https://<API Server>/apis/v1.3/packages
e.g. https://vhqtest.verifone.com/apis/v1.3/packages
```

Sequence Diagram



Input Parameters

Element	Optional/Mandatory	Data Type	Description
source	Optional	string	Source contains values from PackageSources table Ex: • MarketApp • PayWare • Weaver All packages are rendered if the source is not sent in headers.
modelId/modelName	Optional	Array	Supported model identifier.
limit	Optional	Int	Rows per request.
offset	Optional	Int	Offset required to calculate pageNumber.
platform	Optional	String	Platform like eVo or V/OS or Vx or Engage.
appId	Optional	Int	Application Id. Additional information or Tag to identify the Package. Client Application can use this to add and retrieve the package.

Output Parameters

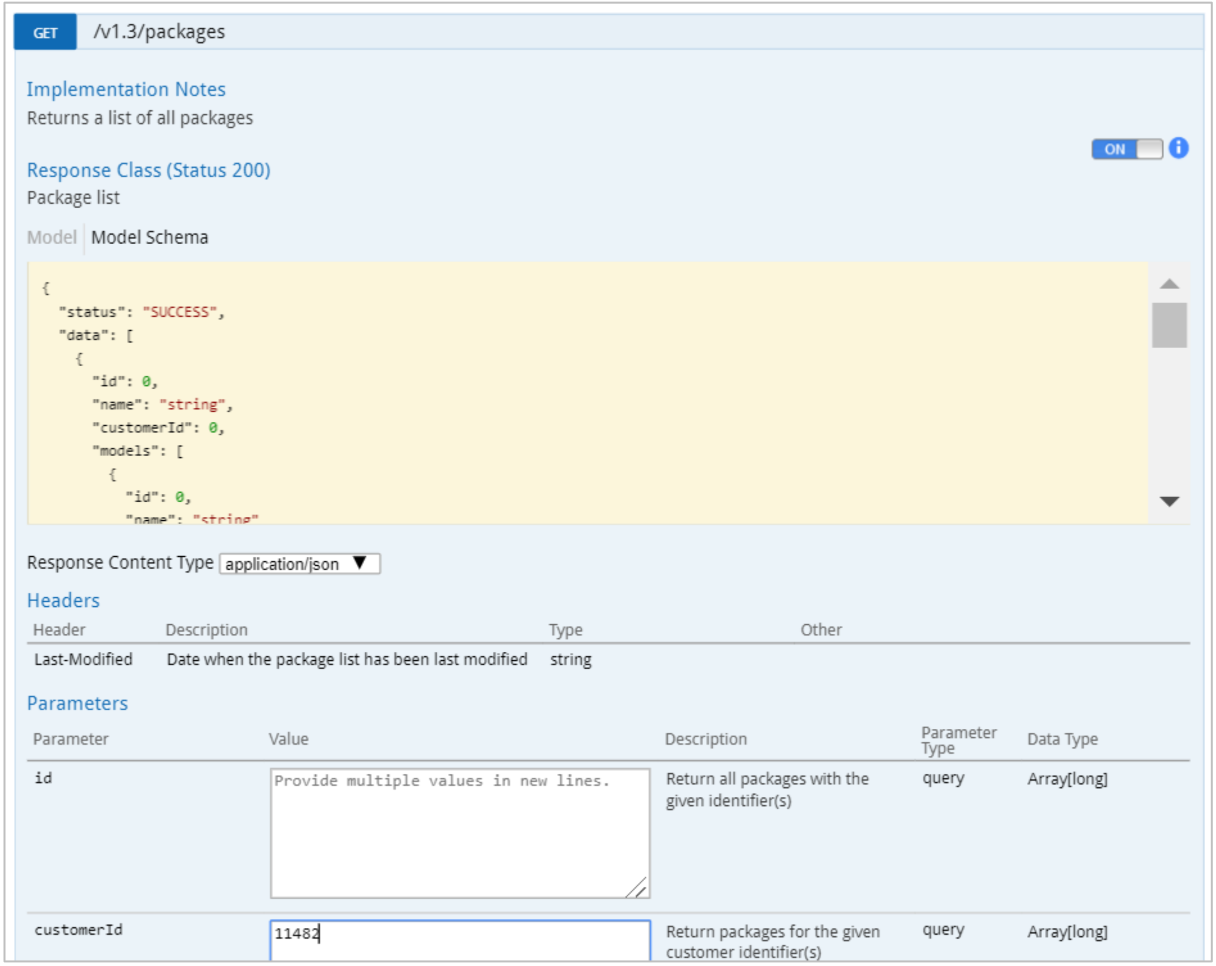
Element	Attribute	Data Type	Description
package (collection of packages)	id	Int	Id of the package.
	name	String	Name of the Package.
	version	String	Version of the Package.

Get Software Packages API - Sample Code

SAMPLE CODE SNIPPET

```
public class GetPackages {
   private static String url = "http://<servername:port>/apis/v1.3/";
   private int statusCode;
   // This is the base URL, it will be appended as per the entity which is invoked
   /**
    * @param completeURL -> This is the URL which appended as per the
   entity which is invoked
    * @param method -> This refer to which method we are going to
   send e.g. get,post,put
    * @return -> httpCon with Content Type and Authorization token
    * @throws IOException
    */
   HttpURLConnection getConnection(URL completeURL, String method)
         throws IOException {
      HttpURLConnection httpCon = (HttpURLConnection)
            completeURL.openConnection();
      httpCon.setDoOutput(true);
      httpCon.setRequestProperty("Authorization", "Bearer 4eafc052-4d05-377b-a53c-cb76e68a3d48");
      httpCon.setRequestMethod(method);
      httpCon.setRequestProperty("Content-Type", "application/json");
      return httpCon;
   }
   /**
    * This method get all data of a entity
    * @param Api -> This refers to the endpoint of the URL for the
   particular entity e.g. for software entity it is softwares
    * @param id -> This refer to show specific data from Entity based
   on Id.
    * @param fields -> This refers to show specific fields, which we
   pass as a parameter.
    * @param queryparameter
    * @return response from entity
    */
   String getPackages(String Api, int id,String fullPath,String[] queryparameter,String queryParamValue) {
      StringBuffer response=null;
      if(id!=0) {
         url = url.concat(Api) + "/" + id;
      }else {
         url = url.concat(Api);
      }

      if(fullPath!=null){
         url=url+"/".concat(fullPath);
      }
      if(null!=queryparameter){
         for(String fields: queryparameter){
            url = url +"?"+fields+"="+queryParamValue;
            System.out.println("URL :::: "+url);
         }
      }
      try {
         URL completeURL = new URL(url);
         // Getting All Header Information
         // like(contentType,CustomerName,CustomerId)
         HttpURLConnection httpCon = getConnection(completeURL, "GET");
         httpCon.getResponseCode();
         BufferedReader in = new BufferedReader(new InputStreamReader(httpCon.getInputStream()));
         String inputLine;
         response =new StringBuffer();
         while ((inputLine = in.readLine()) != null) {
            response.append(inputLine);
         }
         System.out.println(response);
         System.out.println(httpCon.getResponseCode());
         System.out.println(httpCon.getResponseMessage());
      } catch (Exception ex) {
         ex.printStackTrace();
         if (ex.getMessage().equalsIgnoreCase("404 Not Found")) {
            statusCode = 404;
            throw new RuntimeException("Not Found " + statusCode);

         } else if (ex.getMessage().equalsIgnoreCase("400 Bad Request")) {
            statusCode = 400;
            throw new RuntimeException("Bad Request " + statusCode);
         } else if (ex.getMessage().equalsIgnoreCase("409 Conflict"))
         {
            statusCode = 409;
            throw new RuntimeException("Already Exist : Conflict " +
                  statusCode);
         } else {
            statusCode = 500;
            throw new RuntimeException("Already Exist : Conflict " +
                  statusCode);
         }
      }
      return response.toString();
   }
   public static void main(String[] args) {
      String [] queryparameter= {"customerId"};
      String queryParamValue = "11482";
      new GetPackages().getPackages("packages",0,null,queryparameter,queryParamValue);
   }
}
```

Sample using Swagger - Request



Implementation Notes

Returns a list of all packages

Response Class (Status 200)

Package list

```
{
  "status": "SUCCESS",
  "data": [
    {
      "id": 0,
      "name": "string",
      "customerId": 0,
      "models": [
        {
          "id": 0,
          "name": "string"
```

Response Content Type application/json ▼

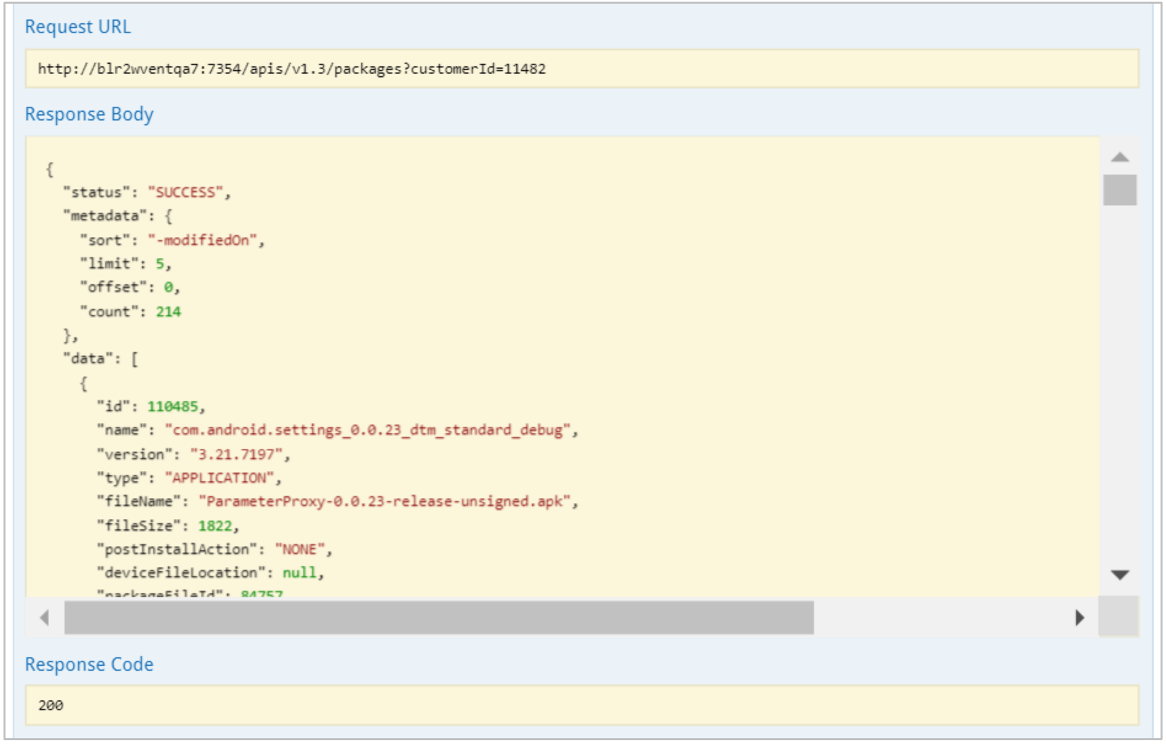
Headers

Header	Description	Type	Other
Last-Modified	Date when the package list has been last modified	string	

Parameters

Parameter	Value	Description	Parameter Type	Data Type
id	Provide multiple values in new lines.	Return all packages with the given identifier(s)	query	Array[long]
customerId	11482	Return packages for the given customer identifier(s)	query	Array[long]

Sample using Swagger - Response



Request URL

```
http://blr2wventqa7:7354/apis/v1.3/packages?customerId=11482
```

Response Body

```
{
  "status": "SUCCESS",
  "metadata": {
    "sort": "-modifiedOn",
    "limit": 5,
    "offset": 0,
    "count": 214
  },
  "data": [
    {
      "id": 110485,
      "name": "com.android.settings_0.0.23_dtm_standard_debug",
      "version": "3.21.7197",
      "type": "APPLICATION",
      "fileName": "ParameterProxy-0.0.23-release-unsigned.apk",
      "fileSize": 1822,
      "postInstallAction": "NONE",
      "deviceFileLocation": null,
      "packageFileId": 84757
```

Response Code

```
200
```

Error Codes

HTTP Status Code	Application Code	Name	Severity	Message
200	S100	S_OK	-	Success
401	E101	E_TOKEN_INVALID_OR_EXPIRED	2	Token expired or it is invalid.
401	E102	E_UNAUTHORIZED_ACCESS	2	The user does not have the right to perform the operation.
500	E103	E_GENERAL_ERROR	3	API failed.

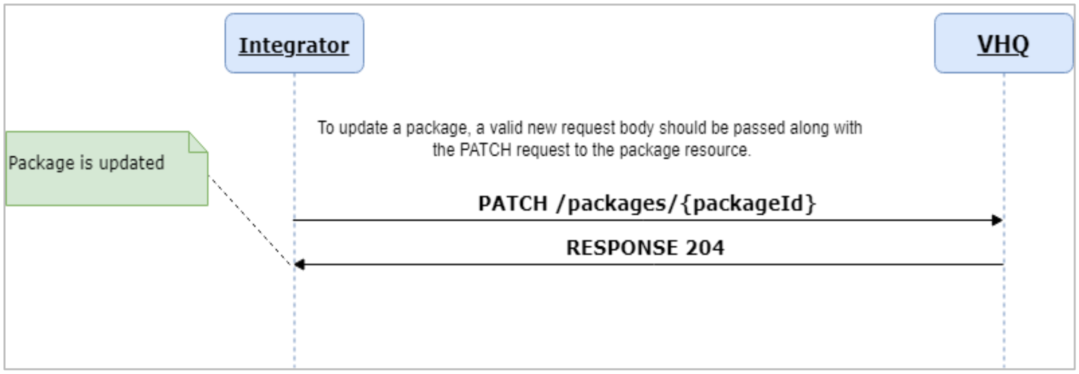
UPDATE SOFTWARE PACKAGES-1.4

This API is used to update the package details of a specified packageId.

Go to the following path:

```
API - PATCH/packages/{packageId}
URL - https://<API Server>/apis/v1.4/packages/{packageId}
e.g. https://vhqtest.verifone.com/apis/v1.4/packages/{packageId}
```

Sequence Diagram


Integrator
VHQ
Package is updated
To update a package, a valid new request body should be passed along with the PATCH request to the package resource.
PATCH /packages/{packageId}
RESPONSE 204

Request Body

```
{
  "data": {
    "id": 0,
    "name": "string",
    "customerId": 0,
    "models": [
      {
        "id": 0,
        "name": "string"
      }
    ],
    "type": "APPLICATION",
    "version": "string",
    "tags": "string",
    "postInstallAction": "NONE",
    "fileName": "string",
    "fileSize": "string",
    "downloadAutomationEnabled": "string",
    "packageFileId": 0,
    "thumbNailLocationURL": "string",
    "previewURL": "string",
    "fileNameOnDevice": "string",
    "deviceFileLocation": "string",
    "targetUser": "usr1",
    "containerize": true,
    "folderId": 0,
    "folderName": "string"
  }
}
```

Input Parameters

Element	Optional/Mandatory	Data Type	Description
name	Optional	String	Friendly name of the package. Required to add the entity. Value cannot be modified after the entity is created.
models	Optional	Array	Supported model identifier.
type	Optional	String	Package type. Required to add the entity. Value can be modified after the entity is created.

			= ['APPLICATION', 'BUNDLE', 'FEATURE_ENABLEMENT_LICENSE', 'FORM', 'OS', 'PARAMETER_FILE', 'SERVICE_PACK']
folderId	Optional	Int	Id of the folder.
folderName	Optional	String	Name of the folder.
containerize	Optional	Boolean	Indicates the VHQ to maintain the uniqueness of the applications contained in the Package.
postInstallAction	Optional	String	Post-installation action of the package. Required to add the entity. Value can be modified after the entity is created. = ['NONE', 'REBOOT', 'RESTART_APPLICATIONS']
downloadAutomationEnabled	Optional	String	Indicates if the Package is enabled for download automation. The package is enabled for download automation based on the content of the associated file. Read-only.
FileNameOnDevice	Optional	String	Name of the file on the device. Required to add the content file. Value can be modified after the entity is created.
deviceFileLocation	Optional	String	Location of the file on the device. Required to add the content file. Value can be modified after the entity is created.
targetUser	Optional	String	User location on the device. Required to add the content file. Value can be modified after the entity is created. = ['usr1', 'usr2', 'usr3', 'usr4', 'usr5', 'usr6', 'usr7', 'usr8', 'usr9', 'usr10']
limit	Optional	Int	Rows per request.
offset	Optional	Int	Offset required to calculate page Number.

Output Parameters

None

Update Package API - Sample Code

SAMPLE CODE SNIPPET

```
public class UpdatePackagev14 {
        private static String url = "http://blr2wventqa7:7354/apis/v1.4/";
//This is the base URL, it will be appended as per the entity which is invoked along with the
corresponding packageId
        private static String requestBody = "{\r\n" + " \"data\":  {\r\n" +
                                                                                          "
\"customerId\": 1473,\r\n" +
                                                                                          "
\"postInstallAction\": \"REBOOT\",\r\n" +
                                                                                          " }\r\n" +
                                                                                          "}"; // Patch
Request body in JSON format



/**
 * @param Api -> This refers to the endpoint of the URL for the particular entity e.g. for Hierarchy
entity it is hierarchies
 * @return
 */
   HttpHeaders getHeader(String Api) {
   HttpHeaders headers = new HttpHeaders();
   headers.set("Authorization", "Bearer b57e17a1-b50d-325a-9566-ad24c9973fe0");
   headers.setContentType(MediaType.APPLICATION_JSON);
   return headers;
   }
/**
 * @param Api ->  This refers to the endpoint of the URL for the particular entity e.g. for Hierarchy
entity it is hierarchies
 * @param body -> The request body which is required to be patched, This changes as per the entity
 * @param id   -> This refer to which specific Row need to change
 * @param queryParamValue
 * @param queryparameter
 */
   void patch(String api, String body, int id, String[] queryparameter, String queryParamValue) {
    int statusCode;

     url = url.concat(api) + "/" + id;
    if(null!=queryparameter){
        for(String fields: queryparameter){
            url = url +"?"+fields+"="+queryParamValue;
            System.out.println("URL :::: "+url);
        }
    }
    HttpHeaders headers = getHeader(api);

   try {

            System.out.println("Url :::::::: " +url);
            URL completeURL = new URL(url);
    // Getting All Header Information
    // like(contentType,CustomerName,CustomerId)
      HttpEntity<String> entity = new HttpEntity<String>(body, headers);
      RestTemplate restTemplate = new RestTemplate();
      HttpComponentsClientHttpRequestFactory requestFactory = new
HttpComponentsClientHttpRequestFactory();
      requestFactory.setConnectTimeout(18000);
      requestFactory.setReadTimeout(18000);
      restTemplate.setRequestFactory(requestFactory);
      //HttpEntity<String> result = restTemplate.exchange(BASE_PATH, HttpMethod.PATCH, entity,
String.class);
       ResponseEntity<String> responseEntity = restTemplate.exchange(url, HttpMethod.PATCH, entity,
String.class);
      statusCode =  responseEntity.getStatusCode().value();
      System.out.println(statusCode);

   } catch (Exception ex) {
      ex.printStackTrace();
      if (ex.getMessage().equalsIgnoreCase("404 Not Found")) {
         statusCode = 404;
         throw new RuntimeException("Not Found " + statusCode);
      } else if (ex.getMessage().equalsIgnoreCase("400 Bad Request")) {
         statusCode = 400;
         throw new RuntimeException("Bad Request " + statusCode);
      } else if (ex.getMessage().equalsIgnoreCase("409 Conflict")) {
         statusCode = 409;
         throw new RuntimeException("Already Exist : Conflict " + statusCode);
      } else {
         statusCode = 500;
         throw new RuntimeException("Already Exist : Conflict " + statusCode);
      }
   }
}
   public static void main(String[] args) {
              String [] queryparameter= {"customerId"};
           String queryParamValue = "11482";
       new UpdatePackagev14().patch("packages", requestBody,98342,queryparameter,queryParamValue);

       System.out.println("Updated Suceesfully");
    }
}
```

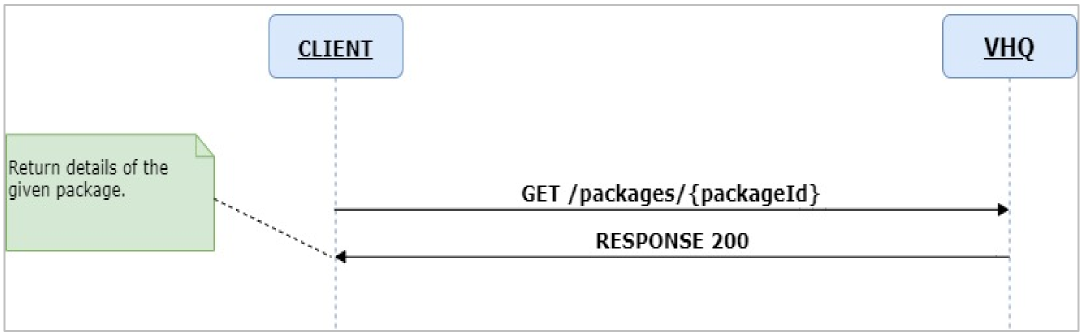
Get Software Packages by PackageId

This API is used to get the list of all packages of a packageId.

Go to the following path:

```
API - GET/packages/{packageId}
URL - https://<API Server>/apis/v1.3/packages/{packageId}
e.g. https://vhqtest.verifone.com/apis/v1.3/packages/{packageId}
```

Sequence Diagram


CLIENT
VHQ
Return details of the given package.
GET /packages/{packageId}
RESPONSE 200

Input Parameters

Element	Optional/Mandatory	Data Type	Description
packageId	Mandatory	Int	Package identifier.
limit	Optional	Int	Rows per request.
offset	Optional	Int	Offset required to calculate the page number.
platform	Optional	String	Platform like eVo or V/OS or Vx or Engage.

Output Parameters

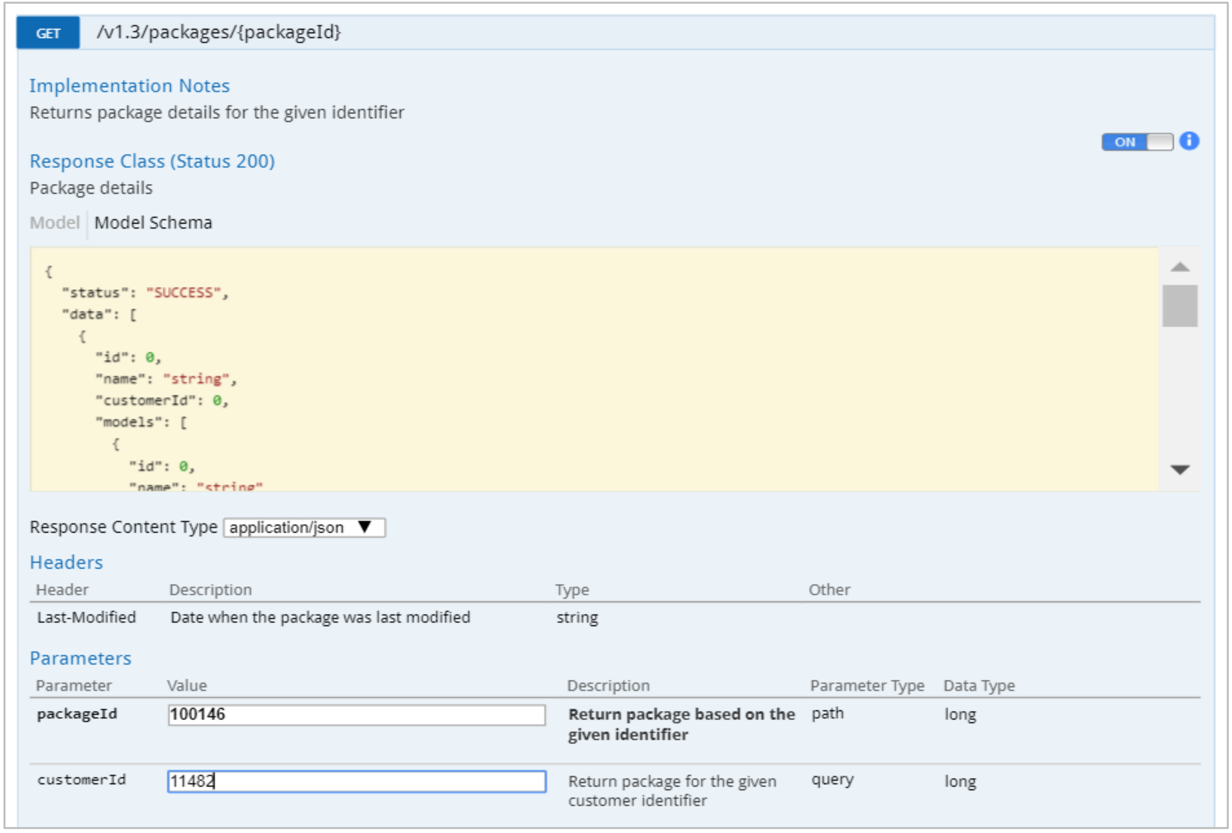
Element	Attribute	Data Type	Description
package (collection of packages)	id	Int	Id of the package.
	name	String	Name of the Package.
	version	String	Version of the Package.
	models	Array	Supported model identifier(s).

Get Software Packages by PackageId API - Sample Code

SAMPLE CODE SNIPPET

```
public class GetPackages {

   private static String url = "http://<servername:port>/apis/v1.3/";
   private int statusCode;
   // This is the base URL, it will be appended as per the entity which is invoked

   /**
    * @param completeURL -> This is the URL which appended as per the
   entity which is invoked
    * @param method -> This refer to which method we are going to
   send e.g. get,post,put
    * @return -> httpCon with Content Type and Authorization token
    * @throws IOException
    */
   HttpURLConnection getConnection(URL completeURL, String method)
         throws IOException {
      HttpURLConnection httpCon = (HttpURLConnection)
            completeURL.openConnection();
      httpCon.setDoOutput(true);
      httpCon.setRequestProperty("Authorization", "Bearer 4eafc052-4d05-377b-a53c-cb76e68a3d48");
      httpCon.setRequestMethod(method);
      httpCon.setRequestProperty("Content-Type", "application/json");
      return httpCon;
   }
   /**
    * This method get all data of a entity
    * @param Api -> This refers to the endpoint of the URL for the
   particular entity e.g. for software entity it is softwares
    * @param id -> This refer to show specific data from Entity based
   on Id.
    * @param fields -> This refers to show specific fields, which we
   pass as a parameter.
    * @param queryparameter
    * @return response from entity
    */
   String getPackages(String Api, int id,String fullPath,String[] queryparameter,String queryParamValue) {
      StringBuffer response=null;
      if(id!=0) {
         url = url.concat(Api) + "/" + id;
      }else {
         url = url.concat(Api);
      }

      if(fullPath!=null){
         url=url+"/".concat(fullPath);
      }
      if(null!=queryparameter){
         for(String fields: queryparameter){
            url = url +"?"+fields+"="+queryParamValue;
            System.out.println("URL :::: "+url);
         }
      }
      try {
         URL completeURL = new URL(url);
         // Getting All Header Information
         // like(contentType,CustomerName,CustomerId)
         HttpURLConnection httpCon = getConnection(completeURL, "GET");
         httpCon.getResponseCode();
         BufferedReader in = new BufferedReader(new InputStreamReader(httpCon.getInputStream()));
         String inputLine;
         response =new StringBuffer();
         while ((inputLine = in.readLine()) != null) {
            response.append(inputLine);
         }
         System.out.println(response);
         System.out.println(httpCon.getResponseCode());
         System.out.println(httpCon.getResponseMessage());
      } catch (Exception ex) {
         ex.printStackTrace();
         if (ex.getMessage().equalsIgnoreCase("404 Not Found")) {
            statusCode = 404;
            throw new RuntimeException("Not Found " + statusCode);

         } else if (ex.getMessage().equalsIgnoreCase("400 Bad Request")) {
            statusCode = 400;
            throw new RuntimeException("Bad Request " + statusCode);
         } else if (ex.getMessage().equalsIgnoreCase("409 Conflict"))
         {
            statusCode = 409;
            throw new RuntimeException("Already Exist : Conflict " +
                  statusCode);
         } else {
            statusCode = 500;
            throw new RuntimeException("Already Exist : Conflict " +
                  statusCode);
         }
      }
      return response.toString();
   }
   public static void main(String[] args) {
      String [] queryparameter= {"customerId"};
      String queryParamValue = "11482";
      new GetPackages().getPackages("packages",100339,null,queryparameter,queryParamValue);
   }
}
```

Sample using Swagger - Request



GET /v1.3/packages/{packageId}

Implementation Notes

Returns package details for the given identifier

Response Class (Status 200)

Package details

Model | Model Schema

```
{
  "status": "SUCCESS",
  "data": [
    {
      "id": 0,
      "name": "string",
      "customerId": 0,
      "models": [
        {
          "id": 0,
          "name": "string"
```

Response Content Type application/json ▼

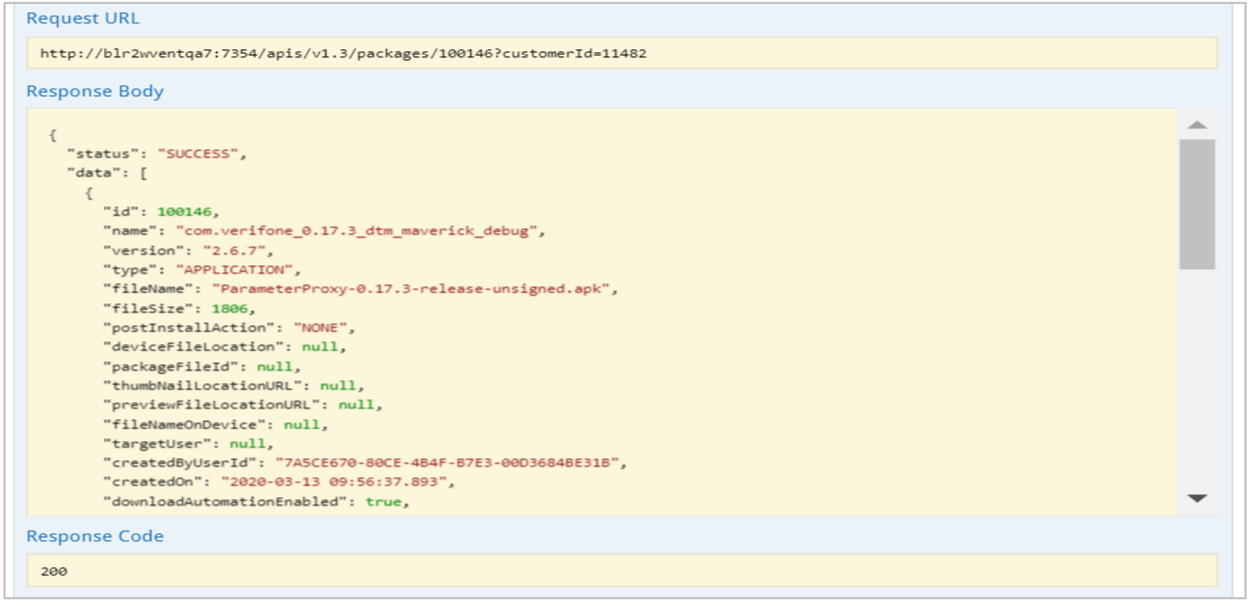
Headers

Header	Description	Type	Other
Last-Modified	Date when the package was last modified	string	

Parameters

Parameter	Value	Description	Parameter Type	Data Type
packageId	100146	**Return package based on the given identifier**	path	long
customerId	11482	Return package for the given customer identifier	query	long

Sample using Swagger – Response



Request URL

```
http://blr2wventqa7:7354/apis/v1.3/packages/100146?customerId=11482
```

Response Body

```
{
  "status": "SUCCESS",
  "data": [
    {
      "id": 100146,
      "name": "com.verifone_0.17.3_dtm_maverick_debug",
      "version": "2.6.7",
      "type": "APPLICATION",
      "fileName": "ParameterProxy-0.17.3-release-unsigned.apk",
      "fileSize": 1806,
      "postInstallAction": "NONE",
      "deviceFileLocation": null,
      "packageFileId": null,
      "thumbNailLocationURL": null,
      "previewFileLocationURL": null,
      "fileNameOnDevice": null,
      "targetUser": null,
      "createdByUserId": "7A5CE670-80CE-4B4F-B7E3-00D3684BE31B",
      "createdOn": "2020-03-13 09:56:37.893",
      "downloadAutomationEnabled": true,
```

Response Code

```
200
```

Error Codes

HTTP Status Code	Application Code	Name	Severity	Message
200	S100	S_OK	-	Success.
401	E101	E_TOKEN_INVALID_OR_EXPIRED	2	Token expired or invalid.
401	E102	E_UNAUTHORIZED_ACCESS	2	The user does not have the right to perform the operation.
500	E103	E_GENERAL_ERROR	3	API failed.

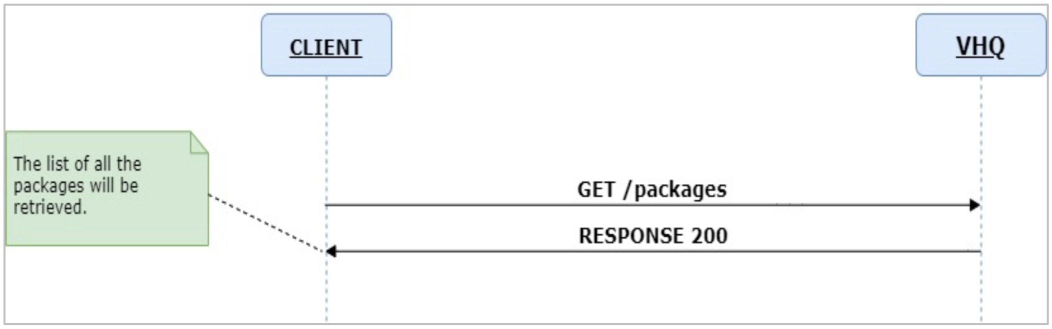
GET DEVICE SOFTWARE PACKAGES

This API is used to get all the packages of a device.

Go to the following path:

```
API - GET/devices/{deviceUid}packages
URL - https://<API Server>/apis/v1.3/devices/{deviceUid}/packages
e.g. https://vhqtest.verifone.com/apis/v1.3/devices/{deviceUid}packages
```

Sequence Diagram



Input Parameters

Element	Optional/Mandatory	Data Type	Description
deviceUid	Mandatory	Int	Unique device identifier.
limit	Optional	Int	Rows per request.
offset	Optional	Int	Offset required to calculate page number.
platform	Optional	String	Platform like eVo or V/OS or Vx or Engage.

Output Parameters

Element	Attribute	Data Type	Description
packages (collection of packages)	id	Int	Id of the package.
	name	String	Name of the Package.

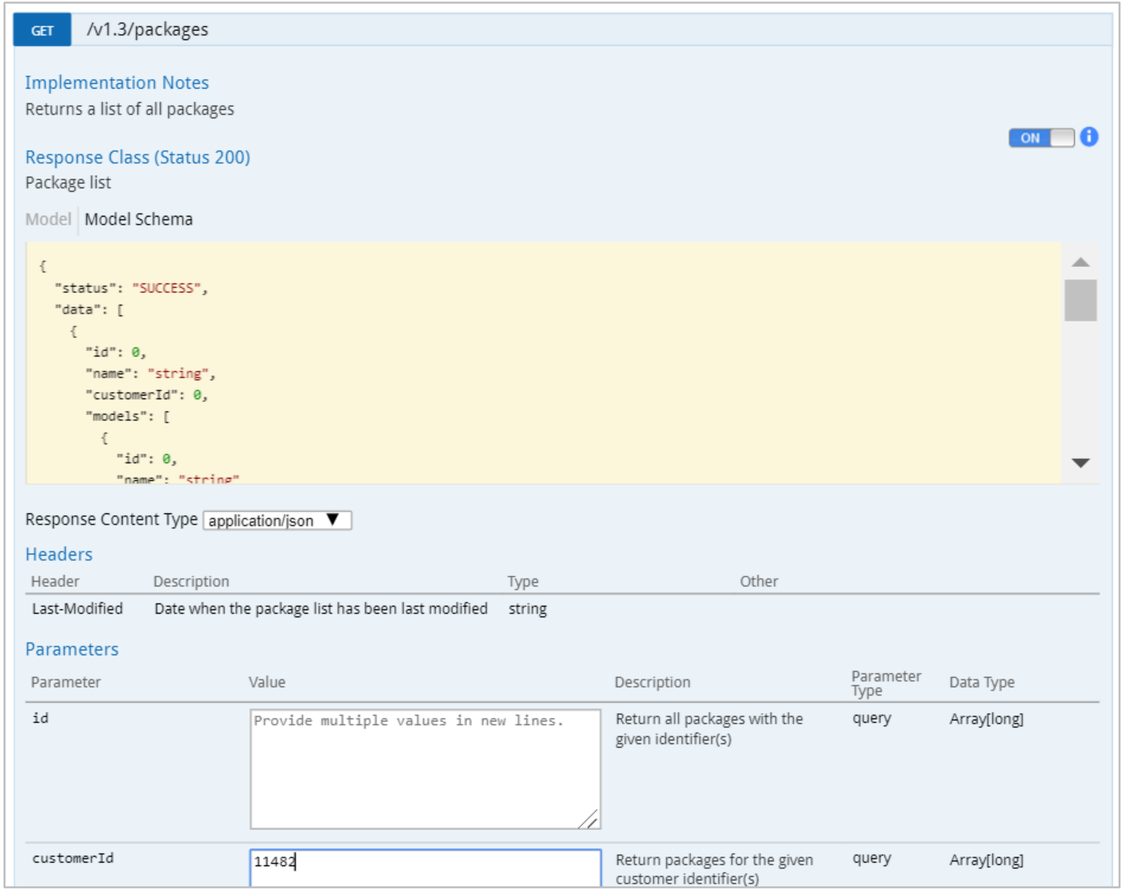
	version	String	Version of the Package.
	models	Array	Supported model identifier(s).

Get Device Software Packages API - Sample Code

SAMPLE CODE SNIPPET

```
public class GetPackages {
   private static String url = "http://<servername:port>/apis/v1.3/";
   private int statusCode;
   // This is the base URL, it will be appended as per the entity which is invoked
   /**
    * @param completeURL -> This is the URL which appended as per the
   entity which is invoked
    * @param method -> This refer to which method we are going to
   send e.g. get,post,put
    * @return -> httpCon with Content Type and Authorization token
    * @throws IOException
    */
   HttpURLConnection getConnection(URL completeURL, String method)
         throws IOException {
      HttpURLConnection httpCon = (HttpURLConnection)
            completeURL.openConnection();
      httpCon.setDoOutput(true);
      httpCon.setRequestProperty("Authorization", "Bearer 4eafc052-4d05-377b-a53c-cb76e68a3d48");
      httpCon.setRequestMethod(method);
      httpCon.setRequestProperty("Content-Type", "application/json");
      return httpCon;
   }
   /**
    * This method get all data of a entity
    * @param Api -> This refers to the endpoint of the URL for the
   particular entity e.g. for software entity it is softwares
    * @param id -> This refer to show specific data from Entity based
   on Id.
    * @param fields -> This refers to show specific fields, which we
   pass as a parameter.
    * @param queryparameter
    * @return response from entity
    */
   String getPackages(String Api, int id,String fullPath,String[] queryparameter,String queryParamValue) {
      StringBuffer response=null;
      if(id!=0) {
         url = url.concat(Api) + "/" + id;
      }else {
         url = url.concat(Api);
      }

      if(fullPath!=null){
         url=url+"/".concat(fullPath);
      }
      if(null!=queryparameter){
         for(String fields: queryparameter){
            url = url +"?"+fields+"="+queryParamValue;
            System.out.println("URL :::: "+url);
         }
      }
      try {
         URL completeURL = new URL(url);
         // Getting All Header Information
         // like(contentType,CustomerName,CustomerId)
         HttpURLConnection httpCon = getConnection(completeURL, "GET");
         httpCon.getResponseCode();
         BufferedReader in = new BufferedReader(new InputStreamReader(httpCon.getInputStream()));
         String inputLine;
         response =new StringBuffer();
         while ((inputLine = in.readLine()) != null) {
            response.append(inputLine);
         }
         System.out.println(response);
         System.out.println(httpCon.getResponseCode());
         System.out.println(httpCon.getResponseMessage());
      } catch (Exception ex) {
         ex.printStackTrace();
         if (ex.getMessage().equalsIgnoreCase("404 Not Found")) {
            statusCode = 404;
            throw new RuntimeException("Not Found " + statusCode);

         } else if (ex.getMessage().equalsIgnoreCase("400 Bad Request")) {
            statusCode = 400;
            throw new RuntimeException("Bad Request " + statusCode);
         } else if (ex.getMessage().equalsIgnoreCase("409 Conflict"))
         {
            statusCode = 409;
            throw new RuntimeException("Already Exist : Conflict " +
                  statusCode);
         } else {
            statusCode = 500;
            throw new RuntimeException("Already Exist : Conflict " +
                  statusCode);
         }
      }
      return response.toString();
   }
   public static void main(String[] args) {
      String [] queryparameter= {"customerId"};
      String queryParamValue = "11482";
      new GetPackages().getPackages("packages",0,null,queryparameter,queryParamValue);
   }
}
```

Sample using Swagger - Request



/v1.3/packages

Implementation Notes

Returns a list of all packages

Response Class (Status 200)

Package list

```
{
  "status": "SUCCESS",
  "data": [
    {
      "id": 0,
      "name": "string",
      "customerId": 0,
      "models": [
        {
          "id": 0,
          "name": "string"
```

Response Content Type application/json

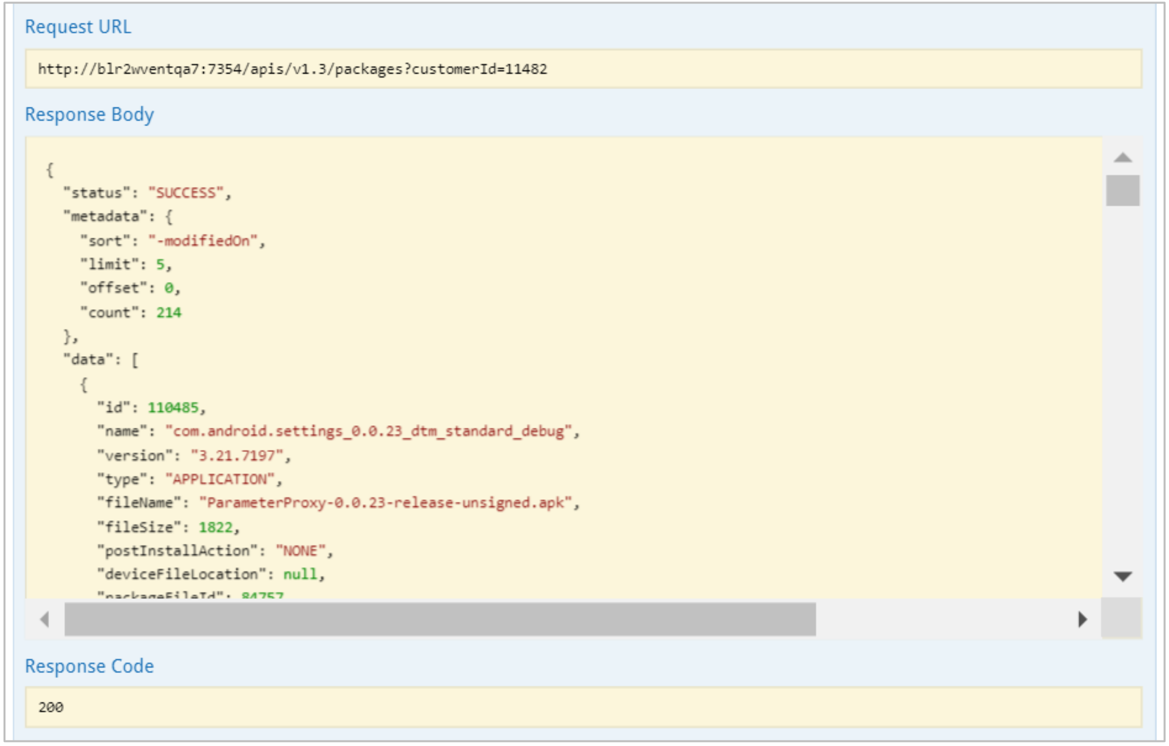
Headers

Header	Description	Type	Other
Last-Modified	Date when the package list has been last modified	string	

Parameters

Parameter	Value	Description	Parameter Type	Data Type
id	Provide multiple values in new lines.	Return all packages with the given identifier(s)	query	Array[long]
customerId	11482	Return packages for the given customer identifier(s)	query	Array[long]

Sample using Swagger – Response



Request URL

```
http://blr2wventqa7:7354/apis/v1.3/packages?customerId=11482
```

Response Body

```
{
  "status": "SUCCESS",
  "metadata": {
    "sort": "-modifiedOn",
    "limit": 5,
    "offset": 0,
    "count": 214
  },
  "data": [
    {
      "id": 110485,
      "name": "com.android.settings_0.0.23_dtm_standard_debug",
      "version": "3.21.7197",
      "type": "APPLICATION",
      "fileName": "ParameterProxy-0.0.23-release-unsigned.apk",
      "fileSize": 1822,
      "postInstallAction": "NONE",
      "deviceFileLocation": null,
      "packageFileId": 84757
```

Response Code

```
200
```

Error Codes

HTTP Status Code	Application Code	Name	Severity	Message
200	S100	S_OK	-	Success.
401	E101	E_TOKEN_INVALID_OR_EXPIRED	2	Token expired or invalid.
401	E102	E_UNAUTHORIZED_ACCESS	2	The user does not have the right to perform the operation.
500	E103	E_GENERAL_ERROR	3	API failed.

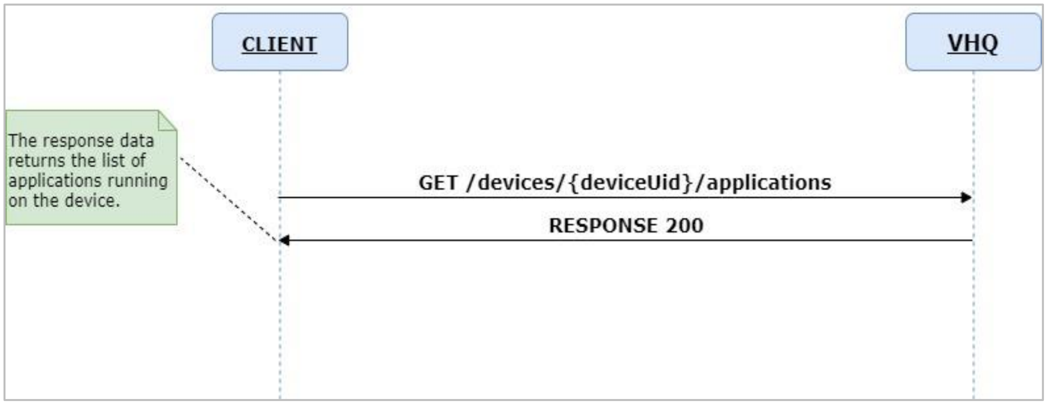
GET DEVICE APPLICATIONS

This API gets the details of applications for a specified device.

Go to the following path:

```
API - GET/devices/{deviceUid}applications
URL - https://<API Server>/apis/v1.3/devices/{deviceUid}/applications
e.g. https://vhqtest.verifone.com/apis/v1.3/devices/{deviceUid}applications
```

Sequence Diagram


CLIENT
VHQ
The response data returns the list of applications running on the device.
GET /devices/{deviceUid}/applications
RESPONSE 200

Input Parameters

Element	Optional/Mandatory	Data Type	Description
udi	Mandatory	Int	Unique device identifier to be sent in the path of the URI.

Output Parameters

Element	Attribute	Data Type	Description
applications (collection of applications)	id	Int	Application Identifier.
	name	String	Name of the Application.

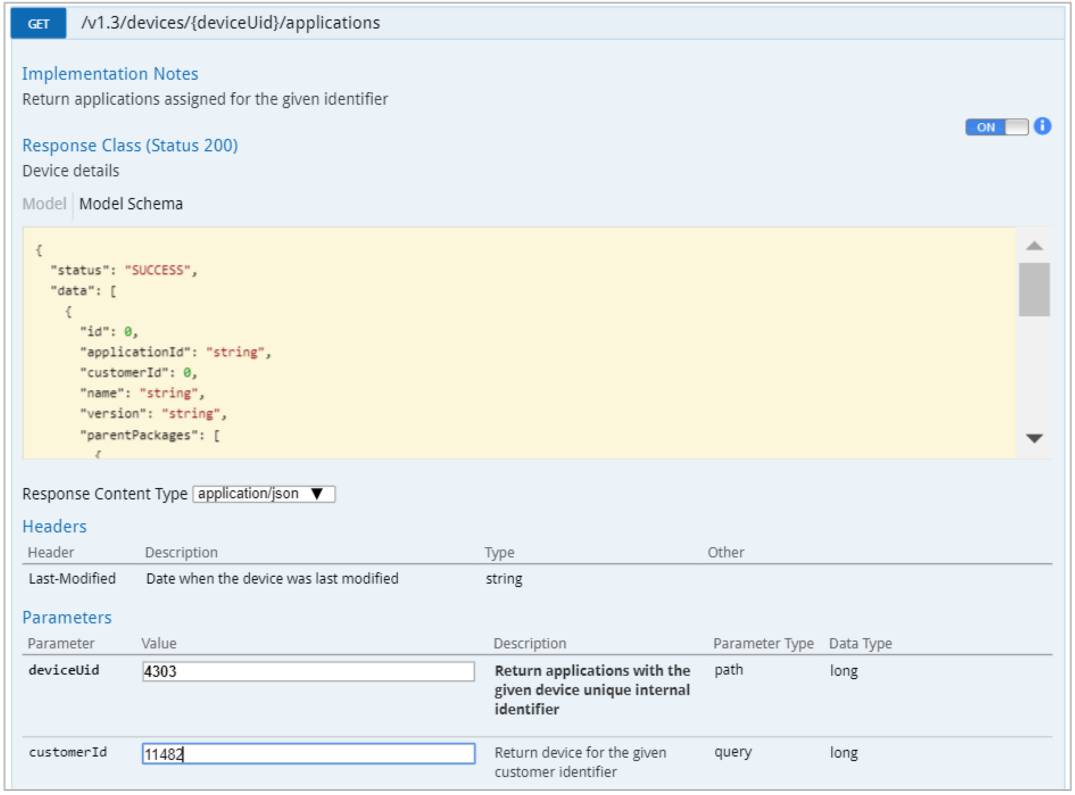
Get Device Applications API - Sample Code

SAMPLE CODE SNIPPET

```
public class GetDeviceApplication {
   private static String url = "http://<servername:port>/apis/v1.3/";
   private int statusCode;
   private static String fullPath="applications";
   // This is the base URL, it will be appended as per the entity which is invoked
   /**
    * @param completeURL -> This is the URL which appended as per the
   entity which is invoked
    * @param method -> This refer to which method we are going to
   send e.g. get,post,put
    * @return -> httpCon with Content Type and Authorization token
    * @throws IOException
    */
   HttpURLConnection getConnection(URL completeURL, String method)
         throws IOException {
      HttpURLConnection httpCon = (HttpURLConnection)
            completeURL.openConnection();
      httpCon.setDoOutput(true);
      httpCon.setRequestProperty("Authorization", "Bearer 8c10910f-9228-37f7-a1d3-d73883ea2615");
      httpCon.setRequestMethod(method);
      httpCon.setRequestProperty("Content-Type", "application/json");
      return httpCon;
   }
   /**
    * This method get all data of a entity
    * @param Api -> This refers to the endpoint of the URL for the
   particular entity e.g. for Device entity it is devices
    * @param id -> This refer to show specific data from Entity based
   on Id.
```

```
    * @param fields -> This refers to show specific fields, which we
   pass as a parameter.
    * @param queryparameter
    * @return response from entity
    */
   String getById(String Api, int id,String fullPath,String[] queryparameter,String queryParamValue) {
      StringBuffer response=null;
      if(id!=0) {
         url = url.concat(Api) + "/" + id;
      }else {
         url = url.concat(Api);
      }

      if(fullPath!=null){
         url=url+"/".concat(fullPath);
      }
      if(null!=queryparameter){
         for(String fields: queryparameter){
            url = url +"?"+fields+"="+queryParamValue;
            System.out.println("URL :::: "+url);
         }
      }
      try {
         URL completeURL = new URL(url);
         // Getting All Header Information
         // like(contentType,CustomerName,CustomerId)
         HttpURLConnection httpCon = getConnection(completeURL, "GET");
         httpCon.getResponseCode();
         BufferedReader in = new BufferedReader(new InputStreamReader(httpCon.getInputStream()));
         String inputLine;
         response =new StringBuffer();
         while ((inputLine = in.readLine()) != null) {
            response.append(inputLine);
         }
         System.out.println(response);
         System.out.println(httpCon.getResponseCode());
         System.out.println(httpCon.getResponseMessage());
      } catch (Exception ex) {
         ex.printStackTrace();
         if (ex.getMessage().equalsIgnoreCase("404 Not Found")) {
            statusCode = 404;
            throw new RuntimeException("Not Found " + statusCode);

         } else if (ex.getMessage().equalsIgnoreCase("400 Bad Request")) {
            statusCode = 400;
            throw new RuntimeException("Bad Request " + statusCode);
         } else if (ex.getMessage().equalsIgnoreCase("409 Conflict"))
         {
            statusCode = 409;
            throw new RuntimeException("Already Exist : Conflict " +
                  statusCode);
         } else {
            statusCode = 500;
            throw new RuntimeException("Already Exist : Conflict " +
                  statusCode);
         }
      }
      return response.toString();
   }
   public static void main(String[] args) {
      String [] queryparameter= {"customerId"};
      String queryParamValue = "11482";
      new GetDeviceApplication().getById("devices",4266,fullPath,queryparameter,queryParamValue);
   }
}
```

Sample using Swagger - Request



/v1.3/devices/{deviceUid}/applications

Implementation Notes

Return applications assigned for the given identifier

Response Class (Status 200)

Device details

Model | Model Schema

```
{
  "status": "SUCCESS",
  "data": [
    {
      "id": 0,
      "applicationId": "string",
      "customerId": 0,
      "name": "string",
      "version": "string",
      "parentPackages": [
        {
```

Response Content Type application/json ▼

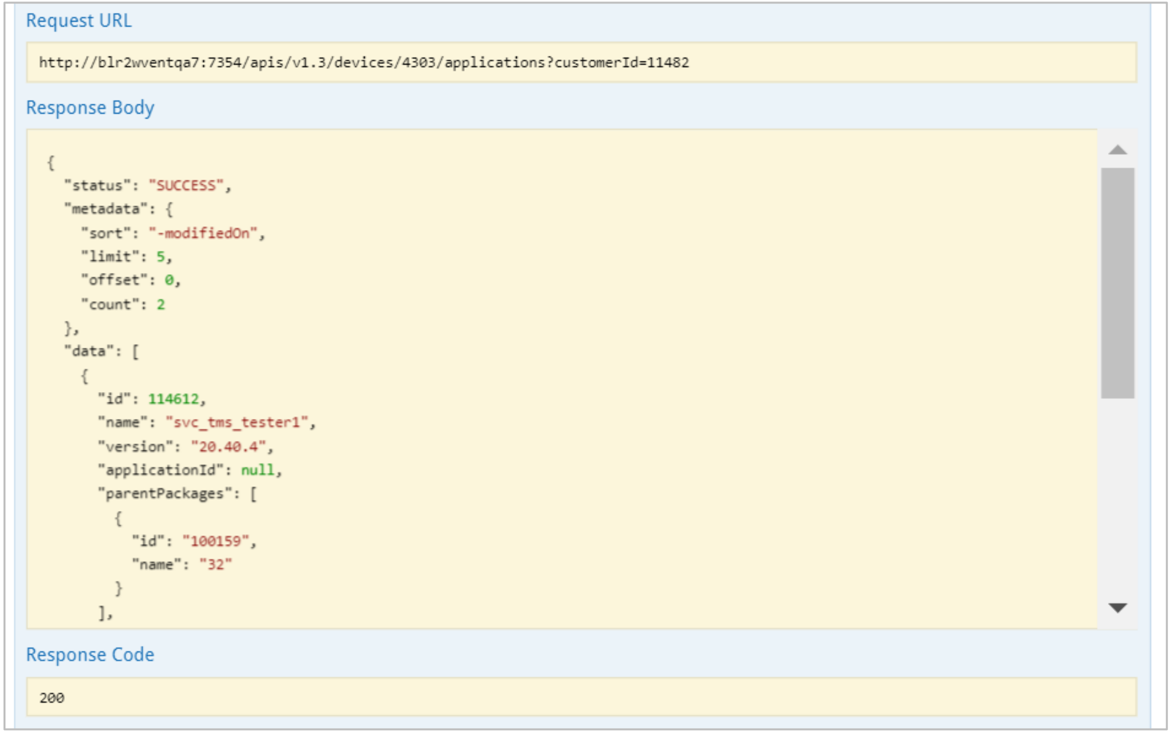
Headers

Header	Description	Type	Other
Last-Modified	Date when the device was last modified	string	

Parameters

Parameter	Value	Description	Parameter Type	Data Type
deviceUid	4303	**Return applications with the given device unique internal identifier**	path	long
customerId	11482	Return device for the given customer identifier	query	long

Sample using Swagger - Response



Request URL

```
http://blr2wventqa7:7354/apis/v1.3/devices/4303/applications?customerId=11482
```

Response Body

```
{
  "status": "SUCCESS",
  "metadata": {
    "sort": "-modifiedOn",
    "limit": 5,
    "offset": 0,
    "count": 2
  },
  "data": [
    {
      "id": 114612,
      "name": "svc_tms_tester1",
      "version": "20.40.4",
      "applicationId": null,
      "parentPackages": [
        {
          "id": "100159",
          "name": "32"
        }
      ],
```

Response Code

```
200
```

Error Codes

HTTP Status Code	Application Code	Name	Severity	Message
200	S100	S_OK	-	Success.
401	E101	E_TOKEN_INVALID_OR_EXPIRED	2	Token is expired or it is invalid.
401	E102	E_UNAUTHORIZED_ACCESS	2	The user does not have the right to perform the operation.
500	E103	E_GENERAL_ERROR	3	API failed.

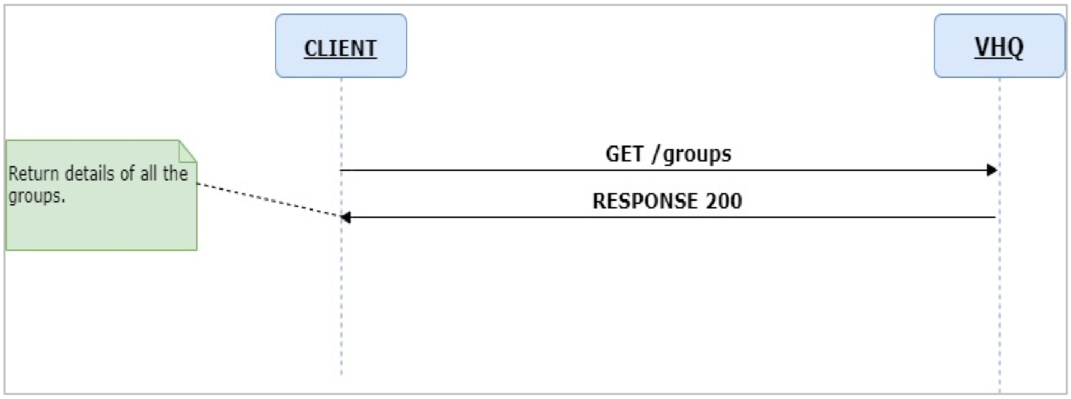
GET GROUPS

This API gets all the Device Groups in the VHQ. The API supports pagination.

Go to the following path:

```
API - GET/groups
URL – https://<API Server>/apis/v1.1/groups
e.g. https://vhqtest.verifone.com/apis/v1.1/groups
```

Sequence Diagram



Input Parameters

Input parameters has to pass in query string.

Element	Optional/Mandatory	Data Type	Description
limit	Optional	Int	Rows per request.
offset	Optional	Int	Offset is required to calculate the page number.

Output Parameters

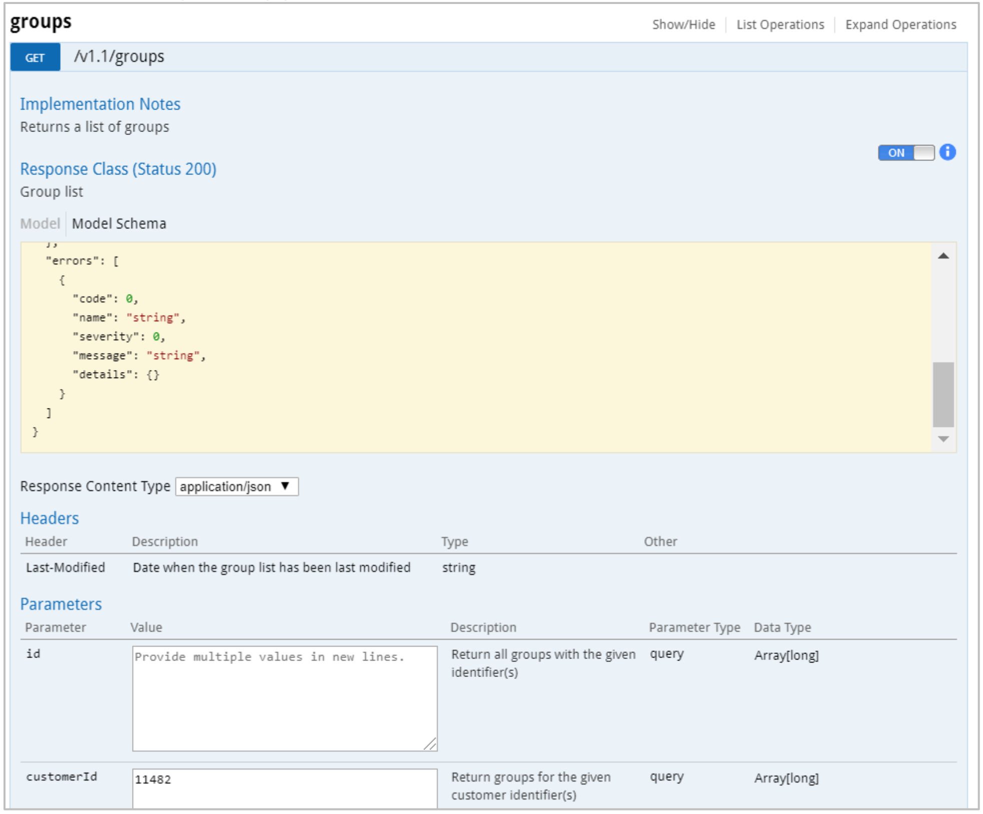
Element	Attribute	Data Type	Description
group (collection of Group)	id	Int	Id of the group.
	name	String	Name of the group.

Get Groups API - Sample Code

SAMPLE CODE SNIPPET

```
public class GetGroups {

   private static String url = "http://<servername:port>/apis/v1.1/";
   private int statusCode;
   // This is the base URL, it will be appended as per the entity which is invoked

   /**
    * @param completeURL -> This is the URL which appended as per the
   entity which is invoked
    * @param method -> This refer to which method we are going to
   send e.g. get,post,put
    * @return -> httpCon with Content Type and Authorization token
    * @throws IOException
    */
   HttpURLConnection getConnection(URL completeURL, String method)
         throws IOException {
      HttpURLConnection httpCon = (HttpURLConnection)
            completeURL.openConnection();
      httpCon.setDoOutput(true);
      httpCon.setRequestProperty("Authorization", "Bearer 019e3fbe-3512-3d7c-b7f6-c05787f08ac0");
      httpCon.setRequestMethod(method);
      httpCon.setRequestProperty("Content-Type", "application/json");
      return httpCon;
   }
   /**
    * This method get all data of a entity
    * @param Api -> This refers to the endpoint of the URL for the
   particular entity e.g. for Group entity it is groups
    * @param id -> This refer to show specific data from Entity based
   on Id.
    * @param fields -> This refers to show specific fields, which we
   pass as a parameter.
    * @param queryparameter
    * @return response from entity
    */
   String getGroups(String Api, int id,String fullPath,String[] queryparameter,String queryParamValue) {
      StringBuffer response=null;
      if(id!=0) {
         url = url.concat(Api) + "/" + id;
      }else {
         url = url.concat(Api);
      }

      if(fullPath!=null){
         url=url+"/".concat(fullPath);
      }
      if(null!=queryparameter){
         for(String fields: queryparameter){
            url = url +"?"+fields+"="+queryParamValue;
            System.out.println("URL :::: "+url);
         }
      }
      try {
         URL completeURL = new URL(url);
         // Getting All Header Information
         // like(contentType,CustomerName,CustomerId)
         HttpURLConnection httpCon = getConnection(completeURL, "GET");
         httpCon.getResponseCode();
         BufferedReader in = new BufferedReader(new InputStreamReader(httpCon.getInputStream()));
         String inputLine;
         response =new StringBuffer();
         while ((inputLine = in.readLine()) != null) {
            response.append(inputLine);
         }
         System.out.println(response);
         System.out.println(httpCon.getResponseCode());
         System.out.println(httpCon.getResponseMessage());
      } catch (Exception ex) {
         ex.printStackTrace();
         if (ex.getMessage().equalsIgnoreCase("404 Not Found")) {
            statusCode = 404;
            throw new RuntimeException("Not Found " + statusCode);

         } else if (ex.getMessage().equalsIgnoreCase("400 Bad Request")) {
            statusCode = 400;
            throw new RuntimeException("Bad Request " + statusCode);
         } else if (ex.getMessage().equalsIgnoreCase("409 Conflict"))
         {
            statusCode = 409;
            throw new RuntimeException("Already Exist : Conflict " +
                  statusCode);
         } else {
            statusCode = 500;
            throw new RuntimeException("Already Exist : Conflict " +
                  statusCode);
         }
      }
      return response.toString();
   }
   public static void main(String[] args) {
      String [] queryparameter= {"customerId"};
      String queryParamValue = "11482";
      new GetGroups().getGroups("groups",0,null,queryparameter,queryParamValue);
   }
}
```

Sample using Swagger - Request



Implementation Notes

Returns a list of groups

Response Class (Status 200)

Group list

```
  }],
  "errors": [
    {
      "code": 0,
      "name": "string",
      "severity": 0,
      "message": "string",
      "details": {}
    }
  ]
}
```

Response Content Type application/json ▼

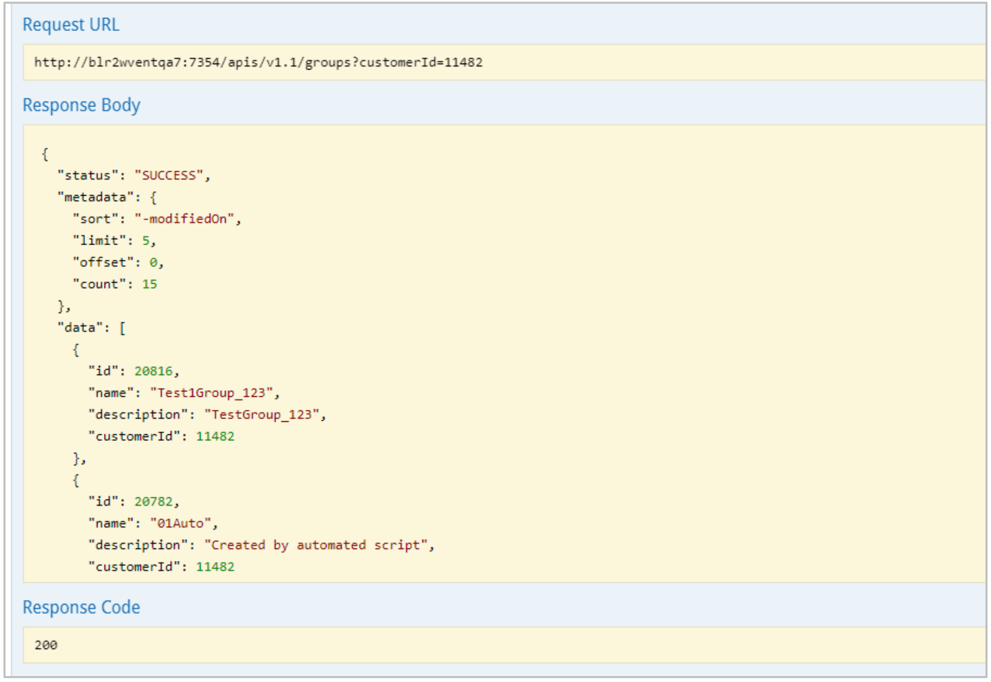
Headers

Header	Description	Type	Other
Last-Modified	Date when the group list has been last modified	string	

Parameters

Parameter	Value	Description	Parameter Type	Data Type
id	Provide multiple values in new lines.	Return all groups with the given identifier(s)	query	Array[long]
customerId	11482	Return groups for the given customer identifier(s)	query	Array[long]

Sample using Swagger – Response



Request URL

```
http://blr2wventqa7:7354/apis/v1.1/groups?customerId=11482
```

Response Body

```
{
  "status": "SUCCESS",
  "metadata": {
    "sort": "-modifiedOn",
    "limit": 5,
    "offset": 0,
    "count": 15
  },
  "data": [
    {
      "id": 20816,
      "name": "Test1Group_123",
      "description": "TestGroup_123",
      "customerId": 11482
    },
    {
      "id": 20782,
      "name": "01Auto",
      "description": "Created by automated script",
      "customerId": 11482
```

Response Code

```
200
```

Error Codes

HTTP Status Code	Code	Name	Severity	Message
200	S100	S_OK	-	Success.
401	E101	E_TOKEN_INVALID_OR_EXPIRED	2	Token is expired or invalid.
401	E102	E_UNAUTHORIZED_ACCESS	2	The user does not have the right to perform the operation.
500	E103	E_GENERAL_ERROR	3	API failed.

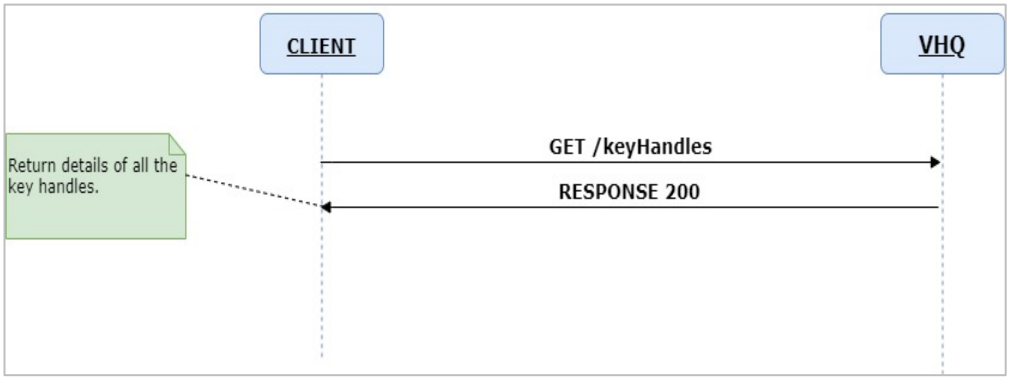
GET KEYHANDLES

This API is used to get the key handles.

Go to the following path:

```
API - GET/KeyHandles
URL - https://<API.Server>/apis/v1.1/KeyHandles
e.g. https://vhqtest.verifone.com/apis/v1.1/KeyHandles
```

Sequence Diagram



Input Parameters

Element	Optional/Mandatory	Data Type	Description
criaCustomerId	Optional	String	CRIA customer id that stores keys for a given key handle.
keyProfileName	Optional	String	Key profile name.
signerCertificateName	Optional	String	Signer certificate name.
criaEndpoint	Optional	String	CRIA endpoint that stores keys for a given key handle.

Output Parameters

Element	Data Type	Description
criaCustomerId	String	CRIA customer id that stores keys for a given key handle.
keyProfileName	String	Key profile name.
signerCertificateName	String	Signer certificate name.
criaEndpoint	String	CRIA endpoint that stores keys for the given key handle.

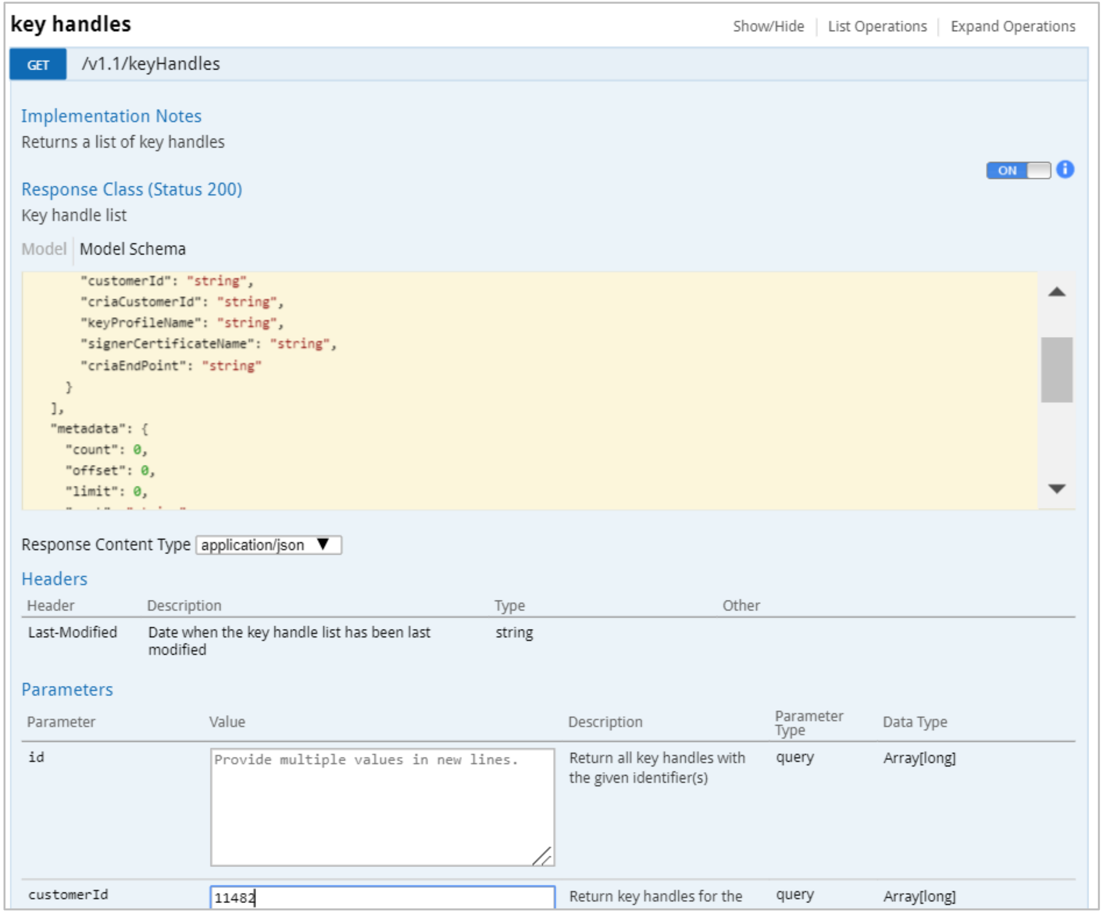
Get Key Handles API - Sample Code

SAMPLE CODE SNIPPET

```
public class GetKeyHandles {

   private static String url = "http://<servername:port>/apis/v1.1/";
   // This is the base URL, it will be appended as per the entity which is invoked

   private int statusCode;

   / * @param completeURL -> This is the URL which appended as per the
   entity which is invoked
    * @param method -> This refer to which method we are going to
   send e.g. get,post,put
    * @return -> httpCon with Content Type and Authorization token
    * @throws IOException
    */
   HttpURLConnection getConnection(URL completeURL, String method)
         throws IOException {
      HttpURLConnection httpCon = (HttpURLConnection)
```

```
            completeURL.openConnection();
      httpCon.setDoOutput(true);
      httpCon.setRequestProperty("Authorization", "Bearer dd7cd592-bf9f-3ec2-9239-6e6d729a657c");
      httpCon.setRequestMethod(method);
      httpCon.setRequestProperty("Content-Type", "application/json");
      return httpCon;
   }
   /**
    * This method get all data of a entity
    * @param Api -> This refers to the endpoint of the URL for the
   particular entity e.g. for Keyhandles entity it is keyhandles
    * @param id -> This refer to show specific data from Entity based
   on Id.
    * @param fields -> This refers to show specific fields, which we
   pass as a parameter.
    * @param queryparameter
    * @return response from entity
    */
   String getKeyHandles(String Api, int id,String fullPath,String[] queryparameter,String queryParamValue) {
      StringBuffer response=null;
      if(id!=0) {
         url = url.concat(Api) + "/" + id;
      }else {
         url = url.concat(Api);
      }

      if(fullPath!=null){
         url=url+"/".concat(fullPath);
      }
      if(null!=queryparameter){
         for(String fields: queryparameter){
            url = url +"?"+fields+"="+queryParamValue;
            System.out.println("URL :::: "+url);
         }
      }
      try {
         URL completeURL = new URL(url);
         // Getting All Header Information
         // like(contentType,CustomerName,CustomerId)
         HttpURLConnection httpCon = getConnection(completeURL, "GET");
         httpCon.getResponseCode();
         BufferedReader in = new BufferedReader(new InputStreamReader(httpCon.getInputStream()));
         String inputLine;
         response =new StringBuffer();
         while ((inputLine = in.readLine()) != null) {
            response.append(inputLine);
         }
         System.out.println(response);
         System.out.println(httpCon.getResponseCode());
         System.out.println(httpCon.getResponseMessage());
      } catch (Exception ex) {
         ex.printStackTrace();
         if (ex.getMessage().equalsIgnoreCase("404 Not Found")) {
            statusCode = 404;
            throw new RuntimeException("Not Found " + statusCode);

         } else if (ex.getMessage().equalsIgnoreCase("400 Bad Request")) {
            statusCode = 400;
            throw new RuntimeException("Bad Request " + statusCode);
         } else if (ex.getMessage().equalsIgnoreCase("409 Conflict"))
         {
            statusCode = 409;
            throw new RuntimeException("Already Exist : Conflict " +
                  statusCode);
         } else {
            statusCode = 500;
            throw new RuntimeException("Already Exist : Conflict " +
                  statusCode);
         }
      }
      return response.toString();
   }
   public static void main(String[] args) {
      String [] queryparameter= {"customerId"};
      String queryParamValue = "11482";
      new GetKeyHandles().getKeyHandles("keyHandles",0,null,queryparameter,queryParamValue);
   }
}
```

Sample using Swagger - Request



/v1.1/keyHandles

Implementation Notes

Returns a list of key handles

Response Class (Status 200)

Key handle list

Model | Model Schema

```
      "customerId": "string",
      "criaCustomerId": "string",
      "keyProfileName": "string",
      "signerCertificateName": "string",
      "criaEndPoint": "string"
    }
  ],
  "metadata": {
    "count": 0,
    "offset": 0,
    "limit": 0,
```

Response Content Type application/json ▼

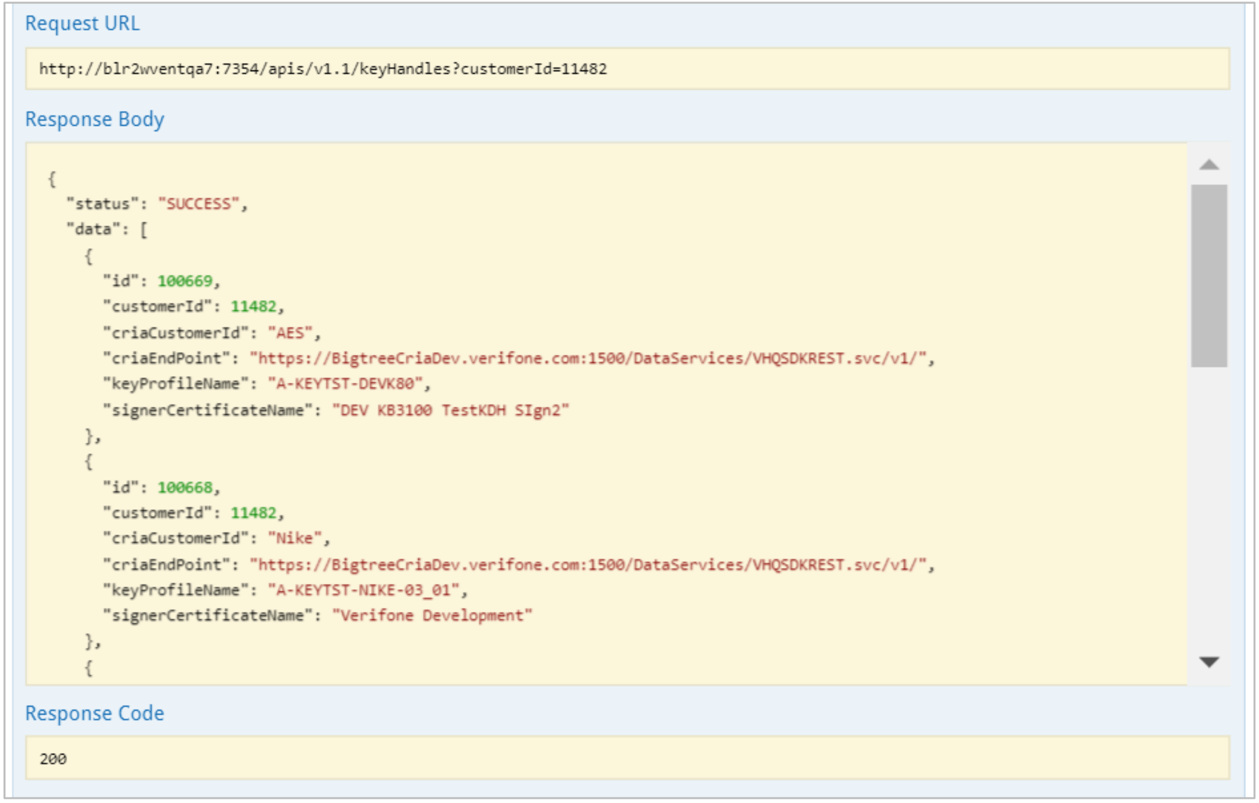
Headers

Header	Description	Type	Other
Last-Modified	Date when the key handle list has been last modified	string	

Parameters

Parameter	Value	Description	Parameter Type	Data Type
id	Provide multiple values in new lines.	Return all key handles with the given identifier(s)	query	Array[long]
customerId	11482	Return key handles for the	query	Array[long]

Sample using Swagger – Response



```
http://blr2wventqa7:7354/apis/v1.1/keyHandles?customerId=11482
```

```
{
  "status": "SUCCESS",
  "data": [
    {
      "id": 100669,
      "customerId": 11482,
      "criaCustomerId": "AES",
      "criaEndPoint": "https://BigtreeCriaDev.verifone.com:1500/DataServices/VHQSDKREST.svc/v1/",
      "keyProfileName": "A-KEYTST-DEVK80",
      "signerCertificateName": "DEV KB3100 TestKDH SIgn2"
    },
    {
      "id": 100668,
      "customerId": 11482,
      "criaCustomerId": "Nike",
      "criaEndPoint": "https://BigtreeCriaDev.verifone.com:1500/DataServices/VHQSDKREST.svc/v1/",
      "keyProfileName": "A-KEYTST-NIKE-03_01",
      "signerCertificateName": "Verifone Development"
    },
    {
```

```
200
```

Error Codes

HTTP Status Code	Application Code	Name	Severity	Message
200	S100	S_OK	-	Success.
401	E101	E_TOKEN_INVALID_OR_EXPIRED	2	Token is expired or invalid.
401	E102	E_UNAUTHORIZED_ACCESS	2	The user does not have the right to perform the operation.
500	E103	E_GENERAL_ERROR	3	API failed.

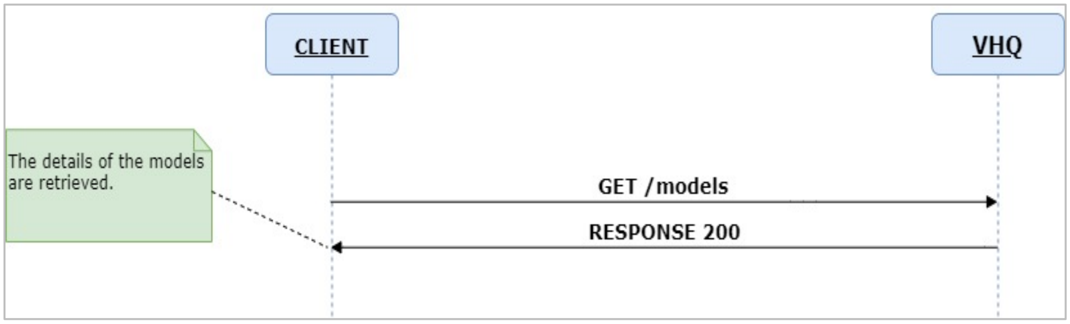
GET MODELS

This API is used to fetch a list of all Device Models supported by the VHQ.

Go to the following path:

```
API - GET/models
URL - https://<API Server>/apis/v1.1/models
e.g. https://vhqtest.verifone.com/apis/v1.1/models
```

Sequence Diagram



Input Parameters

None

Output Parameters

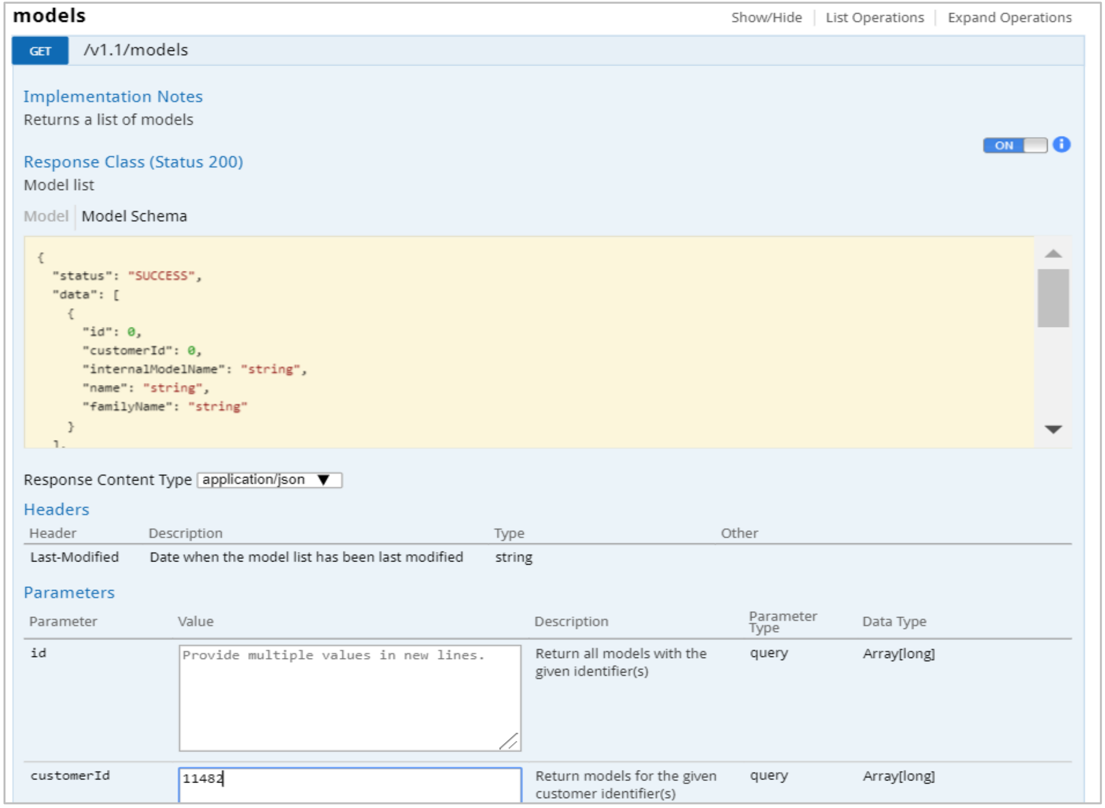
Element	Attribute	Data Type	Description
Models (collection of models)	id	Int	Model identifier
	name	String	To display on GUI, e.g. MX 915
	internalName	String	Internal Use, e.g. Mx915

Get Models API - Sample Code

SAMPLE CODE SNIPPET

```
public class GetModels {
   private static String url = "http://<servername:port>/apis/v1.1/";
   private int statusCode;
   // This is the base URL, it will be appended as per the entity which is invoked
   /**
    * @param completeURL -> This is the URL which appended as per the
   entity which is invoked
    * @param method -> This refer to which method we are going to
   send e.g. get,post,put
    * @return -> httpCon with Content Type and Authorization token
    * @throws IOException
    */
   HttpURLConnection getConnection(URL completeURL, String method)
         throws IOException {
      HttpURLConnection httpCon = (HttpURLConnection)
            completeURL.openConnection();
      httpCon.setDoOutput(true);
      httpCon.setRequestProperty("Authorization", "Bearer dd7cd592-bf9f-3ec2-9239-6e6d729a657c");
      httpCon.setRequestMethod(method);
      httpCon.setRequestProperty("Content-Type", "application/json");
      return httpCon;
   }
   /**
    * This method get all data of an entity
    * @param Api -> This refers to the endpoint of the URL for the
   particular entity e.g. for models entity it is modules
    * @param id -> This refer to show specific data from Entity based
   on Id.
    * @param fields -> This refers to show specific fields, which we
   pass as a parameter.
    * @param queryparameter
    * @return response from entity
    */
   String getModels(String Api, int id,String fullPath,String[] queryparameter,String queryParamValue) {
      StringBuffer response=null;
      if(id!=0) {
         url = url.concat(Api) + "/" + id;
      }else {
         url = url.concat(Api);
      }

      if(fullPath!=null){
         url=url+"/".concat(fullPath);
      }
      if(null!=queryparameter){
         for(String fields: queryparameter){
            url = url +"?"+fields+"="+queryParamValue;
            System.out.println("URL :::: "+url);
         }
      }
      try {
         URL completeURL = new URL(url);
         // Getting All Header Information
         // like(contentType,CustomerName,CustomerId)
         HttpURLConnection httpCon = getConnection(completeURL, "GET");
         httpCon.getResponseCode();
         BufferedReader in = new BufferedReader(new InputStreamReader(httpCon.getInputStream()));
         String inputLine;
         response =new StringBuffer();
         while ((inputLine = in.readLine()) != null) {
            response.append(inputLine);
         }
         System.out.println(response);
         System.out.println(httpCon.getResponseCode());
         System.out.println(httpCon.getResponseMessage());
      } catch (Exception ex) {
         ex.printStackTrace();
         if (ex.getMessage().equalsIgnoreCase("404 Not Found")) {
            statusCode = 404;
            throw new RuntimeException("Not Found " + statusCode);

         } else if (ex.getMessage().equalsIgnoreCase("400 Bad Request")) {
            statusCode = 400;
            throw new RuntimeException("Bad Request " + statusCode);
         } else if (ex.getMessage().equalsIgnoreCase("409 Conflict"))
         {
            statusCode = 409;
            throw new RuntimeException("Already Exist : Conflict " +
                  statusCode);
         } else {
            statusCode = 500;
            throw new RuntimeException("Already Exist : Conflict " +
                  statusCode);
         }
      }
      return response.toString();
   }
   public static void main(String[] args) {
      String [] queryparameter= {"customerId"};
      String queryParamValue = "11482";
      new GetModels().getModels("models",0,null,queryparameter,queryParamValue);
   }
}
```

Sample using Swagger - Request



Implementation Notes

Returns a list of models

Response Class (Status 200)

Model list

```
{
  "status": "SUCCESS",
  "data": [
    {
      "id": 0,
      "customerId": 0,
      "internalModelName": "string",
      "name": "string",
      "familyName": "string"
    }
  ],
```

Response Content Type: application/json

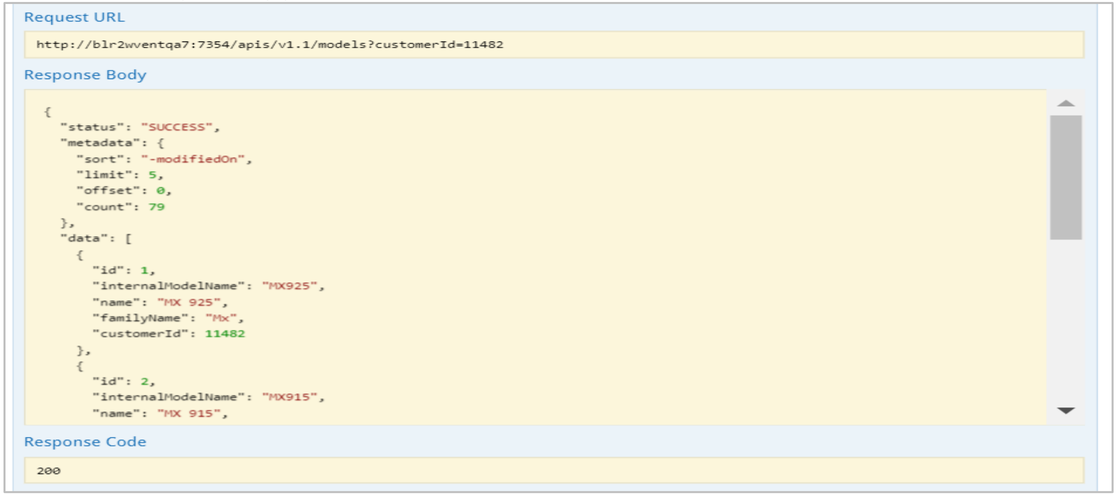
Headers

Header	Description	Type	Other
Last-Modified	Date when the model list has been last modified	string	

Parameters

Parameter	Value	Description	Parameter Type	Data Type
id	Provide multiple values in new lines.	Return all models with the given identifier(s)	query	Array[long]
customerId	11482	Return models for the given customer identifier(s)	query	Array[long]

Sample using Swagger - Response



Request URL

```
http://blr2wventqa7:7354/apis/v1.1/models?customerId=11482
```

Response Body

```
{
  "status": "SUCCESS",
  "metadata": {
    "sort": "-modifiedOn",
    "limit": 5,
    "offset": 0,
    "count": 79
  },
  "data": [
    {
      "id": 1,
      "internalModelName": "MX925",
      "name": "MX 925",
      "familyName": "Mx",
      "customerId": 11482
    },
    {
      "id": 2,
      "internalModelName": "MX915",
      "name": "MX 915",
```

Response Code

```
200
```

Error Codes

HTTP Status Code	Code	Name	Severity	Message
200	S100	S_OK	-	Success
401	E101	E_TOKEN_INVALID_OR_EXPIRED	2	Token is expired or invalid.

401	E102	E_UNAUTHORIZED_ACCESS	2	User does not have the right to perform the operation.
500	E103	E_GENERAL_ERROR	3	API failed.

GET REFERENCESETS

This API is used to get the list of reference sets.

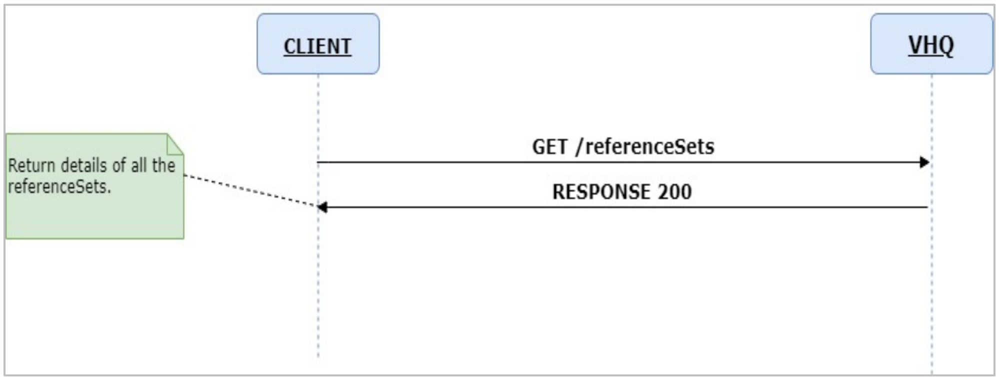
Go to the following path:

API - GET/referencesets

URL – https://<APIError! Hyperlink reference not valid.Server>/apis/v1.1/referencesets

e.g. https://vhqtest.verifone.com/apis/v1.1/referencesets

Sequence Diagram



Input Parameters

None

Output Parameters

Element	Attribute	Data Type	Description
refrenceSets	id	Int	Identifier
modelId		String	This value is required to add a new reference set.
packageId		Array [Integer]	Package identifier.
active		boolean	Indicates if the Reference set is active.
deviceAttributes		String	FIQL expression using the attributes.

Get Reference sets API - Sample Code

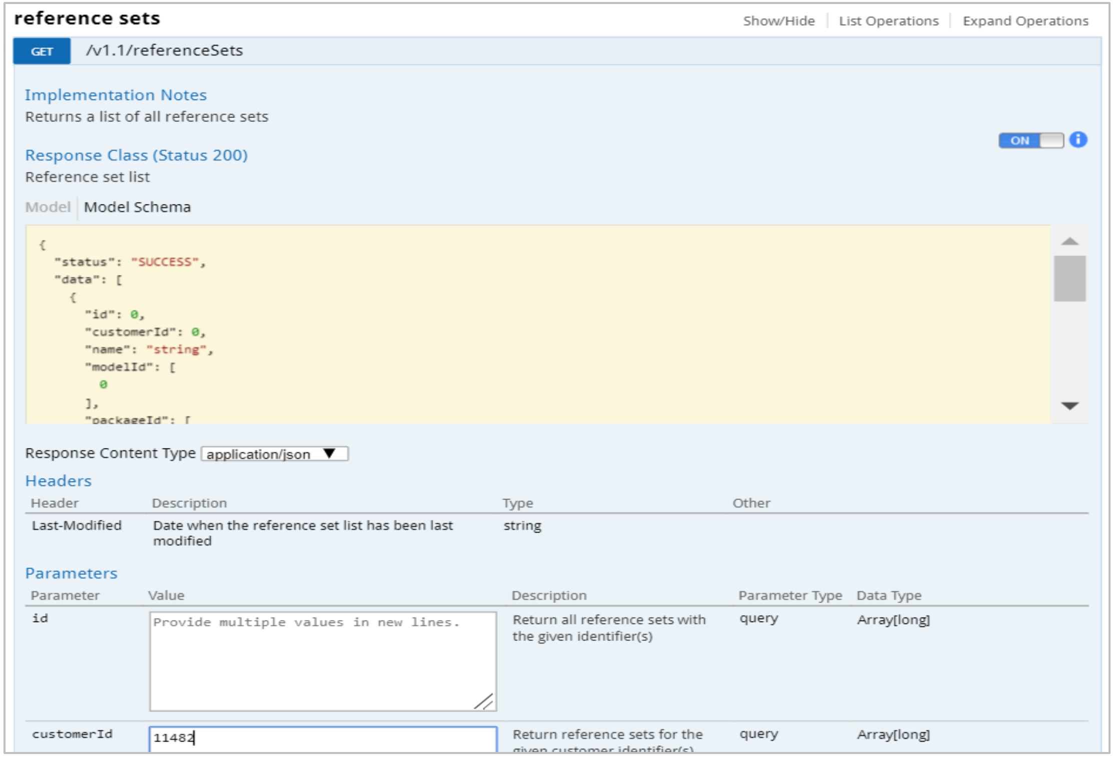
SAMPLE CODE SNIPPET

```
public class GetReferenceSets {

   private static String url = "http://<servername:port>/apis/v1.1/";
   private int statusCode;
   // This is the base URL, it will be appended as per the entity which is invoked
```

```
/**
 * @param completeURL -> This is the URL which appended as per the
entity which is invoked
 * @param method -> This refer to which method we are going to
send e.g. get,post,put
 * @return -> httpCon with Content Type and Authorization token
 * @throws IOException
 */
HttpURLConnection getConnection(URL completeURL, String method)
      throws IOException {
   HttpURLConnection httpCon = (HttpURLConnection)
         completeURL.openConnection();
   httpCon.setDoOutput(true);
   httpCon.setRequestProperty("Authorization", "Bearer 3a3a7b88-b0a8-3891-bc98-b39868301be1");
   httpCon.setRequestMethod(method);
   httpCon.setRequestProperty("Content-Type", "application/json");
   return httpCon;
}
/**
 * This method get all data of a entity
 * @param Api -> This refers to the endpoint of the URL for the
particular entity e.g. for referenceset entity it is referencesets
 * @param id -> This refer to show specific data from Entity based
on Id.
 * @param fields -> This refers to show specific fields, which we
pass as a parameter.
 * @param queryparameter
 * @return response from entity
 */
String getById(String Api,String[] queryparameter,String queryParamValue) {
   StringBuffer response=null;

   url = url.concat(Api);
   if(null!=queryparameter){
      for(String fields: queryparameter){
         url = url +"?"+fields+"="+queryParamValue;
         System.out.println("URL :::: "+url);
      }
   }
   try {
      URL completeURL = new URL(url);
      // Getting All Header Information
      // like(contentType,CustomerName,CustomerId)
      HttpURLConnection httpCon = getConnection(completeURL, "GET");
      httpCon.getResponseCode();
      BufferedReader in = new BufferedReader(new InputStreamReader(httpCon.getInputStream()));
      String inputLine;
      response =new StringBuffer();
      while ((inputLine = in.readLine()) != null) {
         response.append(inputLine);
      }
      System.out.println(response);
      System.out.println(httpCon.getResponseCode());
      System.out.println(httpCon.getResponseMessage());
   } catch (Exception ex) {
      ex.printStackTrace();
      if (ex.getMessage().equalsIgnoreCase("404 Not Found")) {
         statusCode = 404;
         throw new RuntimeException("Not Found " + statusCode);

      } else if (ex.getMessage().equalsIgnoreCase("400 Bad Request")) {
         statusCode = 400;
         throw new RuntimeException("Bad Request " + statusCode);
      } else if (ex.getMessage().equalsIgnoreCase("409 Conflict"))
      {
         statusCode = 409;
         throw new RuntimeException("Already Exist : Conflict " +
               statusCode);
      } else {
         statusCode = 500;
         throw new RuntimeException("Already Exist : Conflict " +
               statusCode);
      }
   }
   return response.toString();
}
public static void main(String[] args) {
   String [] queryparameter= {"customerId"};
   String queryParamValue = "11482";
```

```
        new GetReferenceSets().getById("referenceSets",queryparameter,queryParamValue);
    }
}
```

Sample using Swagger - Request



reference sets

/v1.1/referenceSets

Implementation Notes

Returns a list of all reference sets

Response Class (Status 200)

Reference set list

Model | Model Schema

```
{
  "status": "SUCCESS",
  "data": [
    {
      "id": 0,
      "customerId": 0,
      "name": "string",
      "modelId": [
        0
      ],
      "packageId": [
```

Response Content Type application/json ▼

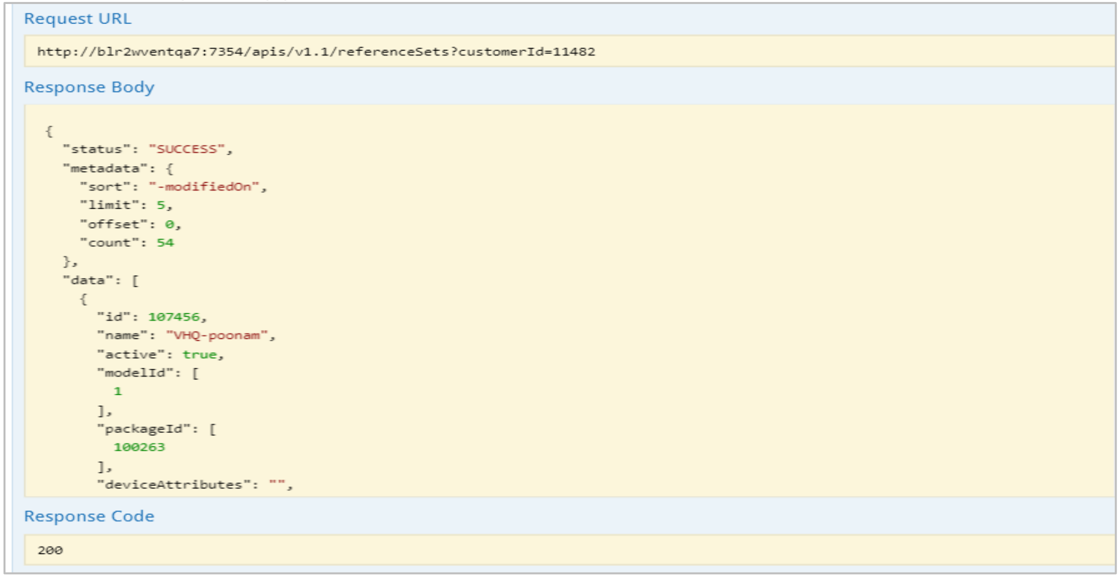
Headers

Header	Description	Type	Other
Last-Modified	Date when the reference set list has been last modified	string	

Parameters

Parameter	Value	Description	Parameter Type	Data Type
id	Provide multiple values in new lines.	Return all reference sets with the given identifier(s)	query	Array[long]
customerId	11482	Return reference sets for the given customer identifier(s)	query	Array[long]

Sample using Swagger – Response



Request URL

```
http://blr2wventqa7:7354/apis/v1.1/referenceSets?customerId=11482
```

Response Body

```
{
  "status": "SUCCESS",
  "metadata": {
    "sort": "-modifiedOn",
    "limit": 5,
    "offset": 0,
    "count": 54
  },
  "data": [
    {
      "id": 107456,
      "name": "VHQ-poonam",
      "active": true,
      "modelId": [
        1
      ],
      "packageId": [
        100263
      ],
      "deviceAttributes": "",
```

Response Code

```
200
```

Error Codes

HTTP Status Code	Application Code	Name	Severity	Message
200	S100	S_OK	-	Success.
401	E101	E_TOKEN_INVALID_OR_EXPIRED	2	Token is expired or invalid.
401	E102	E_UNAUTHORIZED_ACCESS	2	User does not have the right to perform the operation.
500	E103	E_GENERAL_ERROR	3	API failed.

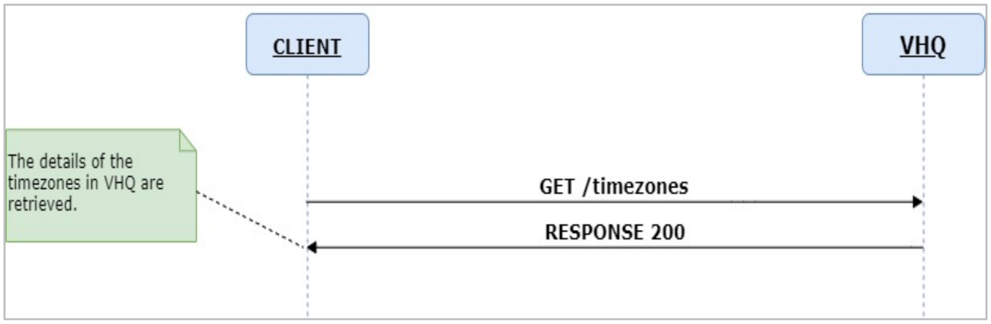
GET TIMEZONES

The Time Zone feature allows you to view all the supported time zones. This API is used to get a list of the time zones supported by the VHQ.

Go to the following path to get a list of the time zones supported by the VHQ:

```
API - GET/timezones
URL - https://<API Server>/apis/v1.3/timezones
e.g. https://vhqtest.verifone.com/apis/v1.3/timezones
```

Sequence Diagram



Input Parameters

None

Output Parameters

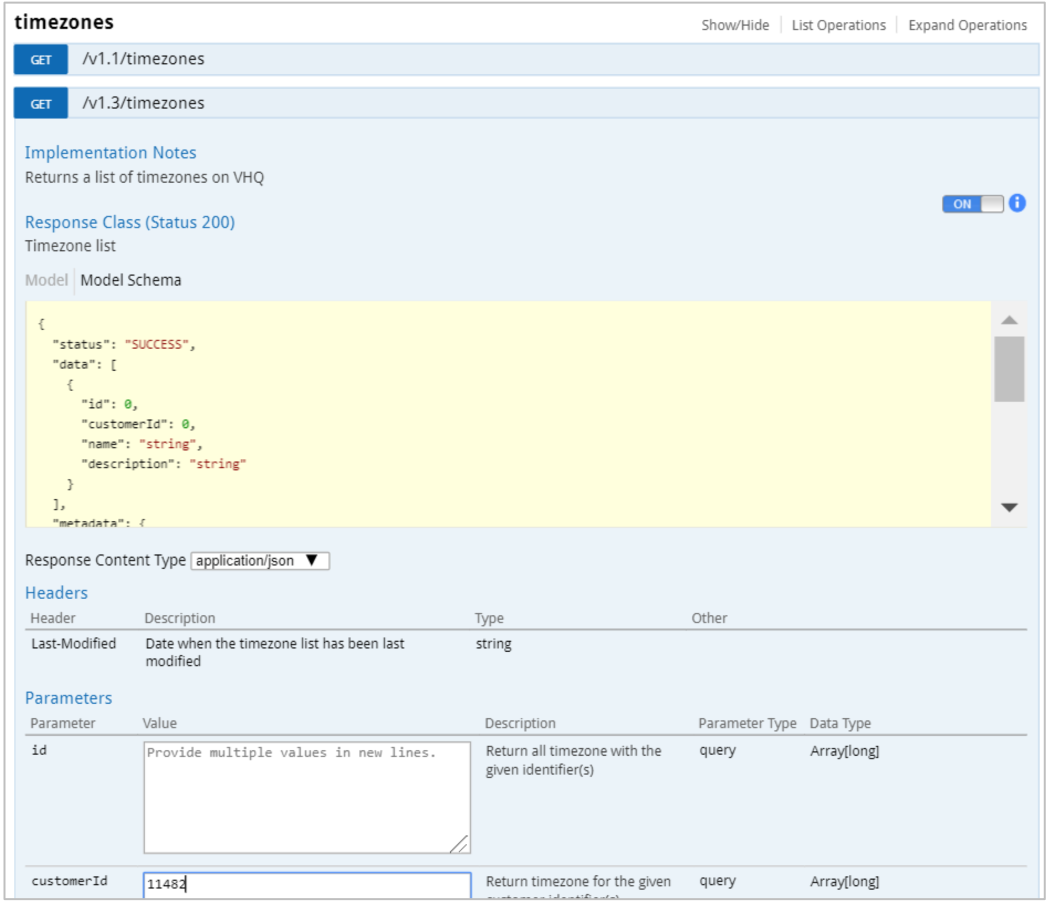
Element	Data Type	Description
id	Integer	Timezone identifier.
name	String	Timezone name. e.g. Pacific Standard Time.
description	String	Timezone description.

Get TimeZones API - Sample Code

SAMPLE CODE SNIPPET

```java
public class GetTimezones {
    private static String url = "http://<servername:port>/apis/v1.3/"; // This is the base URL, it will be appended as 
per the entity which is invoked
    private int statusCode;
    /**
     * @param completeURL -> This is the URL which appended as per the 
    entity which is invoked
     * @param method -> This refer to which method we are going to 
    send e.g. get,post,put
     * @return -> httpCon with Content Type and Authorization token
     * @throws IOException
     */
    HttpURLConnection getConnection(URL completeURL, String method) throws IOException {
        HttpURLConnection httpCon = (HttpURLConnection) 
                completeURL.openConnection();
        httpCon.setDoOutput(true);
        httpCon.setRequestProperty("Authorization", "Bearer c256d098-20f7-3109-9310-03c4e7d77d93");
        httpCon.setRequestMethod(method);
        httpCon.setRequestProperty("Content-Type", "application/json");
        return httpCon;
    }
    /**
     * This method get all data of a entity
     * @param Api -> This refers to the endpoint of the URL for the 
    particular entity e.g. for Timezone entity it is timezones
     * @param fields -> This refers to show specific fields, which we 
    have paas as a parameter.
     * @return response from entity
     */
    String getTimezones(String Api,String[] fields) {
        url = url.concat(Api);
        StringBuffer response=null;
        if(null!=fields){
            for(String field: fields){
                url = url + "?fields=" + field;
            }
        }
        try {
            URL completeURL = new URL(url);
            // Getting All Header Information
            // like(contentType,CustomerName,CustomerId)
            HttpURLConnection httpCon = getConnection(completeURL, 
                    "GET");
            httpCon.getResponseCode();
            BufferedReader in = new BufferedReader(new 
                    InputStreamReader(httpCon.getInputStream()));
            String inputLine;
            response=new StringBuffer();
            while ((inputLine = in.readLine()) != null) {
                response.append(inputLine);
            }
            System.out.println(response);
            System.out.println(httpCon.getResponseCode());
            System.out.println(httpCon.getResponseMessage());
        } catch (Exception ex) {
            ex.printStackTrace();
            if (ex.getMessage().equalsIgnoreCase("404 Not Found")) {
                statusCode = 404;
                throw new RuntimeException("Not Found " + statusCode);
            } else if (ex.getMessage().equalsIgnoreCase("400 Bad Request")) {
                statusCode = 400;
                throw new RuntimeException("Bad Request " + statusCode);
            } else if (ex.getMessage().equalsIgnoreCase("409 Conflict")) 
            {
                statusCode = 409;
                throw new RuntimeException("Already Exist : Conflict " + 
                        statusCode);
            } else {
                statusCode = 500;
                throw new RuntimeException("Already Exist : Conflict " + 
                        statusCode);
            }
        }
        return response.toString();
    }
    public static void main(String[] args) {
        new GetTimezones().getTimezones("timezones",null);
    }
}
```

Sample using Swagger – Request



Implementation Notes

Returns a list of timezones on VHQ

Response Class (Status 200)

Timezone list

```
{
  "status": "SUCCESS",
  "data": [
    {
      "id": 0,
      "customerId": 0,
      "name": "string",
      "description": "string"
    }
  ],
  "metadata": {
```

Response Content Type application/json ▼

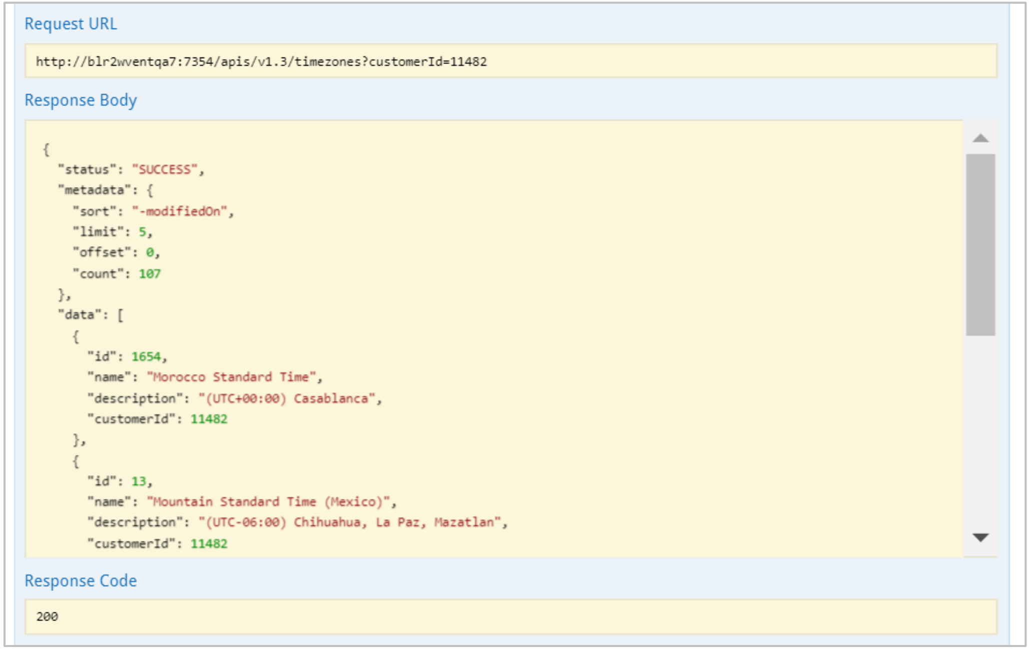
Headers

Header	Description	Type	Other
Last-Modified	Date when the timezone list has been last modified	string	

Parameters

Parameter	Value	Description	Parameter Type	Data Type
id	Provide multiple values in new lines.	Return all timezone with the given identifier(s)	query	Array[long]
customerId	11482	Return timezone for the given	query	Array[long]

Sample using Swagger – Response



```
http://blr2wventqa7:7354/apis/v1.3/timezones?customerId=11482
```

```
{
  "status": "SUCCESS",
  "metadata": {
    "sort": "-modifiedOn",
    "limit": 5,
    "offset": 0,
    "count": 107
  },
  "data": [
    {
      "id": 1654,
      "name": "Morocco Standard Time",
      "description": "(UTC+00:00) Casablanca",
      "customerId": 11482
    },
    {
      "id": 13,
      "name": "Mountain Standard Time (Mexico)",
      "description": "(UTC-06:00) Chihuahua, La Paz, Mazatlan",
      "customerId": 11482
```

```
200
```

Error Codes

HTTP Status Code	Application Code	Name	Severity	Message
200	S100	S_OK	-	Success.
401	E101	E_TOKEN_INVALID_OR_EXPIRED	2	Token is expired or invalid.
401	E102	E_UNAUTHORIZED_ACCESS	2	The user does not have the right to perform the operation.
500	E103	E_GENERAL_ERROR	3	API failed.

VERIFONE SUPPORT

Please contact VCS/Support team if the credentials are not provided/working.

To request a service, please send an email to: global.vcsvhq.request@VERIFONE.com with detailed requirements along with the issue description.

Once the request is sent by e-mail, an e-request is created and assigned to VCS Application Support group.