# AVEND Remote API

v1.0.3 – AAEON Electronics

# Contents

Overview ..... 3
- API Syntax ..... 3
- Data Array ..... 4

API Details ..... 6
- Push Inventory ..... 6
- Pull Inventory ..... 7
- Push Device Info ..... 8
- Pull Device Information ..... 9
- Pull Product ..... 10
- Event Reporting ..... 12
- Push Transaction ..... 14

Appendix ..... 16
- Appendix A: Event Codes ..... 16
- Appendix B: Time zones ..... 19
- Appendix C: Pay Type IDs ..... 20
- Appendix D: Currency IDs ..... 21
- Appendix E: Vend Modes ..... 22

# Overview

AAEON provides the AVEND Remote Portal API (“API”) for comprehensive vending machine management. The API implements the HTTP REST model and uses JSON as the request and response format. To interact with the API you will need an “App Key” and a valid “Device ID”. The “App Key” will be used for all machines across your account. The “Device ID” will be unique for each machine.

It is recommended to contact the API endpoint on a regular basis to ensure a timely synchronization between the vending machines and the portal. A typical interval is around once every ten minutes.

## API Syntax

### Endpoint

https://api2.avend.us/update/

This is the endpoint that will be used to access the AVEND API. You will be required to send the app key and device ID to this endpoint. Each API will require additional information to be included in the “data” array. The general format for the JSON is specified below.

Request Schema: application/json

### Request Syntax

```
{
    "appKey":"appKey",
    "deviceId":"deviceId",
    "data":[
        {"function":"push_inventory",...:...},
        {"function":"push_device_info",...:...}
    ]
}
```

The AVEND Portal will respond to this request with any information that has been modified since the last update. This means that any new information will be sent as a response regardless of the information sent in the “data” tag.

### Response Syntax

```
{
    "response":{
        "pull_inventory":{...},
        "pull_device_info":{...},
        "push_inventory":{"status":"success"},
        "push_device_info":{"status":"success"}
    }
}
```

You will also receive a success or failure response for any function you have sent. In the above example, we received a success response from the AVEND Portal.

## Data Array

The data array will contain fully formed JSON objects known in this document as “function objects.” A function object consists of the API function and its relevant data. Note that multiple function objects can be included in a single request. If the data array is empty, the response will only include any changes performed on the portal since last update.

**Here is an example of a Request where we are sending both “push_inventory” and “event”**

### Request Example

```
{
    "deviceId":"deviceId",
    "appKey":"appKey",
    "data":[{
        "function":"push_inventory",
        "items":[{
            "slot":"1",
            "slotCode":"1",
            "skuid":"100",
            "qty":"75",
            "price":"0.5",
            "depth":"4"
        },{
            "slot":"2",
            "slotCode":"2",
            "skuid":"100",
            "qty":"75",
            "price":"0.5",
            "depth":"4"
        }]
    },{
        "function":"event",
        "eventCode":"9002",
        "occurred":"1970-01-01 00:00:00",
        "metaData":[
            {
                "Skuid":"100",
                "Column #":"10",
                "Product Name":"Sample Drink",
                "Quantity Remaining":"4"
            }
        ]
    }]
}
```

The response for the above endpoint will look similar to the below.

Response Example

```
{
    "response": {
        "pull_inventory": {"status": "No items to pull."},
        "pull_product": {"status": "No products to pull."},
        "pull_device_info": {"status": "No changes for this device."},
        "push_inventory":{"status":"success"},
        "event":{"status":"success"}
    }
}
```

**Here is an example of a request with an empty data array.**

Request Example

```
{
    "appKey":"appKey",
    "deviceId":"deviceId",
    "data":[]
}
```

The response will include all “pull” function objects and a response for any “push” function objects you updated information for. For the above request example, you might receive the following response.

Response Example

```
{
    "response": {
        "pull_inventory": {"status": "No items to pull."},
        "pull_product": {"status": "No products to pull."},
        "pull_device_info": {"status": "No changes for this device."}
    }
}
```

Please see the relevant sections in API Details for more information on the details of each function object and the function tag they use, and the return format for the different pull function objects.

# API Details

## Push Inventory

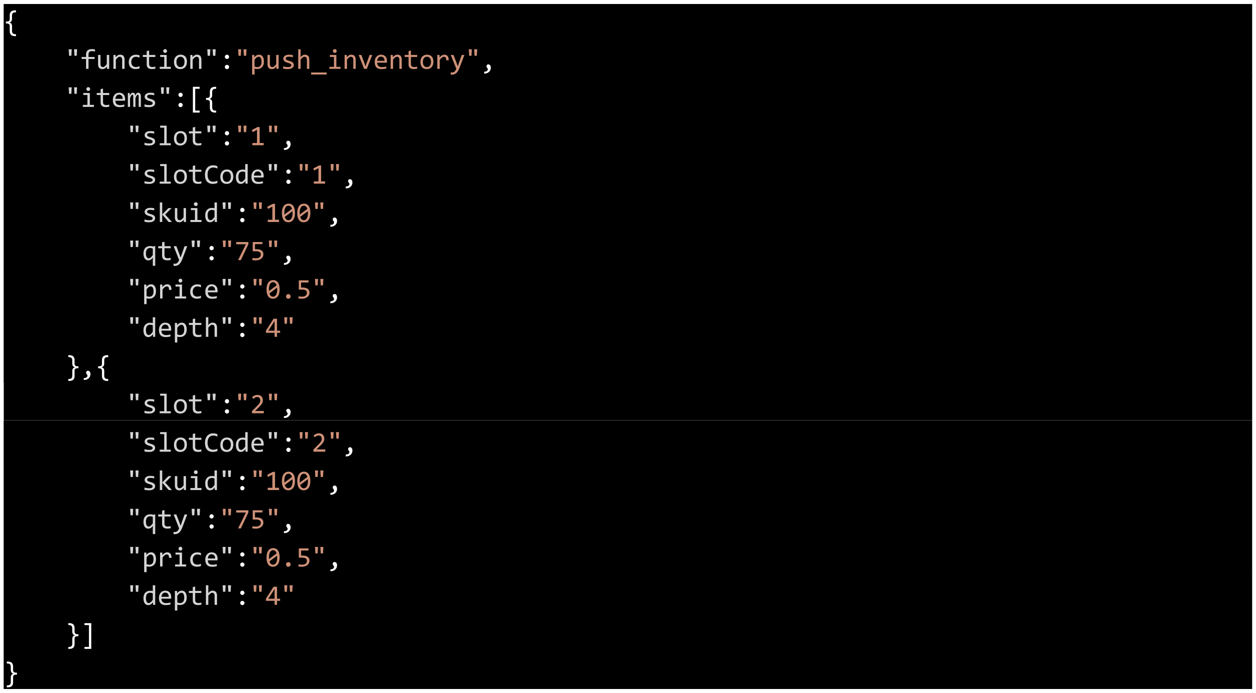
This function is used to update the portal inventory information. The items array will contain a list of all slots that need to be updated. It is good to update all columns on startup to ensure that the AVEND Portal contains the latest up to date information for all columns.

The function tag for updating the inventory is “push_inventory”

### Mandatory Key/Value Pairs

<table>
<tr><th>Key</th><th>Description</th></tr>
<tr><td><b>items</b></td><td>An array of all different slots that need to be changed.
<table>
<tr><th>Key</th><th>Description</th></tr>
<tr><td>slot</td><td>This is the numeric slot number of slot. This should only be numeric.</td></tr>
<tr><td>slotCode</td><td>This is the visible slot identifier and can be alphanumeric.</td></tr>
<tr><td>skuid</td><td>The skuid of the item in the slot</td></tr>
<tr><td>qty</td><td>The current quantity of product in this slot</td></tr>
<tr><td>price</td><td>Price of the slot</td></tr>
<tr><td>depth</td><td>Depth of the slot. This is used for the hardware configuration of the V21i machine.</td></tr>
</table>
</td></tr>
</table>

### Example



```
{
    "function":"push_inventory",
    "items":[{
        "slot":"1",
        "slotCode":"1",
        "skuid":"100",
        "qty":"75",
        "price":"0.5",
        "depth":"4"
    },{
        "slot":"2",
        "slotCode":"2",
        "skuid":"100",
        "qty":"75",
        "price":"0.5",
        "depth":"4"
    }]
}
```

## Pull Inventory

If there is any inventory change from the AVEND Portal, the changes will be sent as a response.

### Response Values

| Key | Description |
|---|---|
| **items** | An array of all different slots that have changed. |
| **slot** | This is the numeric slot number of slot. This should only be numeric and should be used to number the slots. |
| **slotCode** | This is the visible slot identifier and can be alphanumeric. |
| **skuid** | The skuid of the item in the slot |
| **qty** | The current quantity of product in this slot |
| **price** | Price of the slot |
| **depth** | Depth of the slot. This is used for the hardware configuration of the V21i machine. |

### Example

```
"pull_inventory":{
    "items":[{
        "slot":"1",
        "slotCode":"1",
        "skuid":"100",
        "qty":"75",
        "price":"0.5",
        "depth":"4"
    }]
}
```

## Push Device Info

This function is used to update miscellaneous device information to the AVEND Portal. Data should be sent following the template below. The payment device information may not be applicable to your use case, but please include all data specified below in your POST request.

The function tag for updating the device info is “push_device_info”

### Mandatory Key/Value Pairs

| Key | Description |
|---|---|
| **devTypeId** | Dev Type ID will be provided by AAEON. It will be different for each specific type of machine. This should be an int. |
| **timezone** | See Time zone chart. |
| **name** | Name of the device |
| **desc** | Description of the device |
| **location** | Location of device. |
| **locLat** | Latitude of device. This should be a double. |
| **locLng** | Longitude of device. This should be a double. |
| **ccReader** | 1 if card reader is attached. This should be an int. |
| **3rdParty** | 1 if using 3rd party payment device. This should be an int. |
| **billValidator** | 1 if bill validator is attached. This should be an int. |
| **coinMech** | 1 if coin mech is attached. This should be an int. |

### Example

```
{
    "function":"push_device_info",
    "devTypeId":"devTypeId",
    "timezone":"1",
    "name":"Device Name",
    "desc":"Device Description",
    "location":"Device Location",
    "locLat":"latitude",
    "locLng":"longitude",
    "ccreader":0,
    "3rdParty":1,
    "billValidator":0,
    "coinMech":0
}
```

## Pull Device Information

The AVEND Portal will push down any information that has been changed on the AVEND portal.

### Response Values

| Key | Description |
|---|---|
| **devTypeId** | Dev Type ID will be provided by AAEON. It will be different for each specific type of machine. This should be an int. |
| **timezone** | See Time zone chart. |
| **name** | Name of the device |
| **desc** | Description of the device |
| **location** | Location of device |
| **locLat** | Latitude of device. This should be a double. |
| **locLng** | Longitude of device. This should be a double. |
| **ccReader** | 1 if card reader is attached. This should be an int. |
| **3rdParty** | 1 if using 3rd party payment device. This should be an int. |
| **billValidator** | 1 if bill validator is attached. This should be an int. |
| **coinMech** | 1 if coin mech is attached. This should be an int. |

### Example

```
"pull_device_info":{
    "devTypeId":"devTypeId",
    "timezone":"1",
    "name":"Device Name",
    "desc":"Device Description",
    "location":"Device Location",
    "locLat":"latitude",
    "locLng":"longitude",
    "ccreader":0,
    "3rdParty":1,
    "billValidator":0,
    "coinMech":0
}
```

Device Type Id will be provided by AAEON and is used for record keeping purposes.

## Pull Product

The AVEND Portal will return all products that have been updated by the AVEND portal that the machine has not already pulled down yet. This includes new products and modified products. This function allows the deactivation and deletion of products from the local kiosk.

Note that imageUrl and nutriUrl will be url locations to where to retrieve the image.

### Response Values

| Key | Description |
|---|---|
| **skuid** | The skuid of the new or existing product |
| **name** | The name of the product for display to the customer. |
| **size** | The size of the product to be displayed for the customer. |
| **referencePrice** | The expected price of the product. This is a reference price and only sets the default price that the product should be. This actual price can be changed when assigning the sku to a slot. |
| **time** | Timestamp of when this product was last updated. |
| **category** | Not currently in use. |
| **desc** | The description of the product for display to the customer. |
| **imageUrl** | Product image location. See above for more info. |
| **nutriUrl** | Product Nutritional information image location. See above for more info. |
| **spacer** | Spacer information for V21i |
| **depth** | Depth information for V21i |
| **action** | “new” for new product or update to existing product based on skuid.<br>“deactivate” for product deactivation.<br>“delete” for product deletion.<br>“activate” for activation of a previously deactivated product. |

Example

```
"pull_product":{
    "skus":[{
        "skuid":"100",
        "name":"Product Name",
        "size":"12 fl oz",
        "referencePrice":"2.50",
        "time":"1970-01-01 00:00:00",
        "category":"2001",
        "desc":"Item Description",
        "imageUrl":"imageLocation",
        "nutriUrl":"imageLocation",
        "spacer":"12345",
        "depth":"4",
        "action":"new"
    }]
}
```

## Event Reporting

The AVEND Portal will handle this event and then push an email notification to the respective recipients based on the information provided. More information for valid event codes is located at the end of this document.

The function tag for event reporting is “event”

### Mandatory Key/Value Pairs

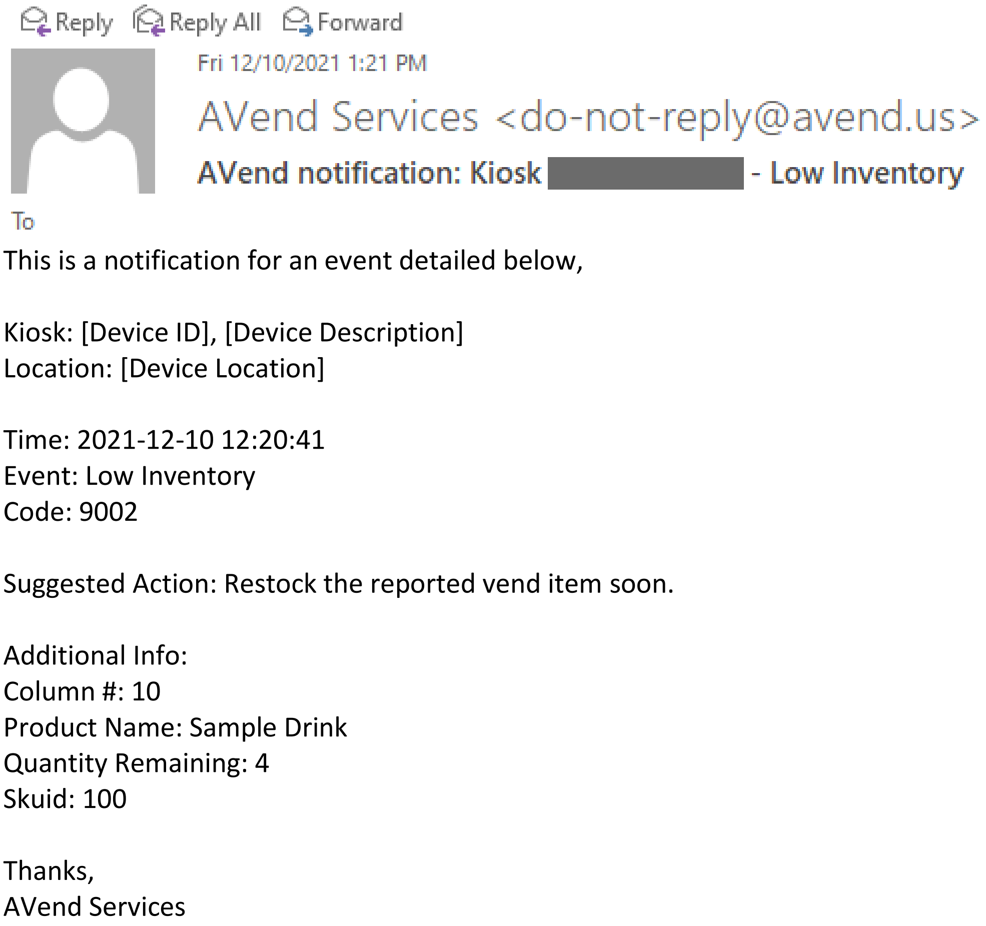
| Key | Description |
|---|---|
| **eventCode** | Is specified in the appendix |
| **occurred** | Is the timestamp of when the event occurred. This can differ from the time this is sent. |
| **metaData** | Contains any other information you may want to be displayed in the alert email. Any tags and data are acceptable here. If there is no meta data to be included in the event, the “metaData” tag should still be included with an empty array. |

### Example

```
{
    "function":"event",
    "eventCode":"9002",
    "occurred":"1970-01-01 00:00:00",
    "metaData":[
        {
            "Skuid":"100",
            "Column #":"10",
            "Product Name":"Sample Drink",
            "Quantity Remaining":"4"
        }
    ]
}
```

Below is a sample email you would receive from the AVEND Portal for the JSON provided above. Note below how the metaData tag is displayed in the email.

### Email Notification Example



Fri 12/10/2021 1:21 PM

AVend Services <do-not-reply@avend.us>

**AVend notification: Kiosk [redacted] - Low Inventory**

To

This is a notification for an event detailed below,

Kiosk: [Device ID], [Device Description]
Location: [Device Location]

Time: 2021-12-10 12:20:41
Event: Low Inventory
Code: 9002

Suggested Action: Restock the reported vend item soon.

Additional Info:
Column #: 10
Product Name: Sample Drink
Quantity Remaining: 4
Skuid: 100

Thanks,
AVend Services

## Push Transaction

This API is used to send transaction data to the AVEND Portal. This is used to update the AVEND Portal of any new transactions that have occurred on a specified vending machine.

The function tag for event reporting is “push_transaction”

### Mandatory Key/Value Pairs

<table>
<tr><th>Key</th><th colspan="2">Description</th></tr>
<tr><td><b>occurred</b></td><td colspan="2">This should be the time that the transaction occurred.</td></tr>
<tr><td><b>payType</b></td><td colspan="2">The type of payment used for this transaction. See Pay Type ID Table for information on valid IDs</td></tr>
<tr><td><b>vendMode</b></td><td colspan="2">Specifies the vend mode that this vend was conducted in. See the Vend Mode Table for more details on the valid vend modes.</td></tr>
<tr><td><b>currencyId</b></td><td colspan="2">The currency used for this transaction.<br>See Currency ID Table for information on valid IDs</td></tr>
<tr><td rowspan="7"><b>items</b></td><td colspan="2">This is an array of products that were vended during this transaction. Only successfully vended products should be included.</td></tr>
<tr><th>Key</th><th>Description</th></tr>
<tr><td><b>skuid</b></td><td>Skuid of the product that was dispensed.</td></tr>
<tr><td><b>slot</b></td><td>Slot number of the product that was dispensed.</td></tr>
<tr><td><b>name</b></td><td>Name of the product that was dispensed.</td></tr>
<tr><td><b>price</b></td><td>The price at which the product was vended for.</td></tr>
<tr><td><b>qtyRemain</b></td><td>The number of products remaining in this column after the vend.</td></tr>
</table>

Example

```
{
    "function":"push_transaction",
    "occurred":"1970-01-01 00:00:00",
    "payType":1,
    "vendMode":"REGULAR",
    "currencyId":1,
    "items":[{
        "skuid":"100",
        "slot":"10",
        "name":"Sample Drink",
        "price":"1.00",
        "qtyRemain":10
    },{
        "skuid":"200",
        "slot":"2",
        "name":"Sample Drink 2",
        "price":"1.00",
        "qtyRemain":12
    }]
}
```

# Appendix

## Appendix A: Event Codes

Provided here is the current list of “Event Codes” that are accepted by this endpoint. Certain events in this table may not be applicable to your use case and can be ignored.

| Code | Type | Description | Extra Messages |
|---|---|---|---|
| **1099** | INFO | Sold Out | Replenish the soldout stock. |
| **1700** | INFO | Timezone Changed | The timezone on the local kiosk has changed. |
| **1801** | INFO | Scheduled Sales Block Mode On | |
| **1802** | INFO | Scheduled Sales Block Mode Off | |
| **1803** | INFO | Scheduled Refrigeration Mode On | |
| **1804** | INFO | Scheduled Refrigeration Mode Off | |
| **1805** | INFO | Scheduled Lights Mode On | |
| **1806** | INFO | Scheduled Lights Mode Off | |
| **1910** | INFO | Discount Mode OFF | |
| **1911** | INFO | Discount Mode On | |
| **2099** | ERROR | Dispense Error | Check Additional Info section. The item position having the dispense error is now not available for vend selection until it's reset. |
| **2300** | INFO | Receipt | |
| **2999** | ERROR | Dispense Error | Per requirements the system has pushed "makeup" sales ticket(s) through for the items failed to dispense. Please make sure the items are given to the customer so the inventory and transaction will reconcile. |
| **4099** | INFO | Door Open | |
| **4100** | INFO | Door Close | |
| **6999** | INFO | Abandoned Cart | |
| **7009** | INFO | Refund Fail | Check Additional Info section and handle accordingly. |
| **7099** | INFO | Payment Not Authorized | |

| | | | |
|---|---|---|---|
| **7200** | ERROR | Incomplete Transaction | |
| **7201** | ERROR | No Payment Device - Sales Block mode ON | Check to resolve the payment device not detected issue. |
| **7202** | ERROR | Payment Device Not Detected | Another detection attempt is initiated. |
| **7203** | INFO | Payment Device Detected | |
| **7309** | ERROR | Cash Box FULL | Clear out the cash box. |
| **7903** | INFO | Payment Authorized | |
| **7999** | ERROR | Refund Fail | Check Additional Info section and handle accordingly. |
| **9001** | WARNING | Machine Restart Notification | |
| **9002** | WARNING | Low Inventory | Restock the reported vend item soon. |
| **9003** | INFO | Scheduled System Restart | The system is restarting according to preset schedule. |
| **9004** | ERROR | System Restart Issue | Contact AVend Services for support. |
| **9006** | INFO | Kiosk App Starts | The kiosk app has started and the machine is now in normal operation mode. |
| **9100** | WARNING | OnSignTV Issue | |
| **9200** | ERROR | Product Temperature High Notification | Cooling cabinet Temperature exceeded 60F for longer than 4 hours. Check and fix for refrigeration failure and restock items affected. |
| **9201** | ERROR | MDB watchdog action | App self-recovery for MDB issue initiated. |
| **9202** | INFO | MDB issue cleared | |
| **9203** | ERROR | MDB error - Sales Block mode ON | Check to resolve the MDB issue between VMC & computer in the kiosk. |
| **9204** | ERROR | Door Opened for More Than 1 Hour | The door switch is faulty or door not closed. |
| **9205** | ERROR | Coin Acceptor Malfunction | Coin Acceptor has self reported a malfunction. Please send someone check if the machine is still operational. |

| | | | |
|---|---|---|---|
| **9206** | ERROR | Bill Validator Malfunction | Bill Validator has self reported a malfunction. Please send someone check if the machine is still operational. |
| **9207** | ERROR | VMC Reported Error | Check Additional Info section and resolve the specific issue if persisted. |
| **9208** | ERROR | VMC Reported Error Cleared | VMC indicated that the prior reported issue(s) listed in the Additional Info section have been cleared. |
| **9209** | ERROR | Hardware Issue - Freezer Locks | |
| **9301** | ERROR | Double Vend | Check Additional Info section and handle accordingly. |
| **9402** | ERROR | Low Disk Space | This kiosk is running low in disk space. Further loss of disk space can result in unexpected behavior. |

## Appendix B: Time zones

These are the specified time zones for use with certain functions objects in this document.

| ID | Time zone |
|---:|---|
| **1** | US Eastern Time |
| **2** | US Central Time |
| **3** | US Mountain Time |
| **4** | US Pacific Time |
| **5** | US Alaska Time |
| **6** | US Hawaii-Aleutian Time |

## Appendix C: Pay Type IDs

| ID | Name |
|---|---|
| **1** | Cash |
| **2** | Credit Card |
| **3** | Promotion |

## Appendix D: Currency IDs

| ID | ISO | Currency |
|---|---|---|
| **1** | USD | US Dollar |
| **2** | CAD | Canadian Dollar |
| **3** | EUR | Euro |

## Appendix E: Vend Modes

| ID | Description |
|---|---|
| **REGULAR** | Vending in normal mode |
| **PROMOTION** | Vending in a promotional mode |
| **TEST** | Vending in a test mode. Vends in this mode will not be tallied during data reconciliation. |