## Restful API Design – Online Store

## Base URL

The base URL of the API is [**https://www.onlinestore.com/api/v1/**](https://www.onlinestore.com/api/v1/)

## Access Control

The Online Store can only be accessed by only one user, the administrator, as of now. The Administrator can add, edit, search or delete products and other resources in the system.

## Authentication

Authentication shall happen using HTTP Basic Authentication over SSL. The user will have to provide his username and password as authorization credentials:

$ curl -u username:password https://www.onlinesite.com/api/v1/products

**HTTP Verbs**

Standard HTTP verbs are used to indicate different types of requests:

* GET - To retrieve a resource or a collection of resources
* POST - To create a resource
* PUT - To set/modify a resource
* DELETE - To delete a resource

## Resources in the System

There are six kinds of resources in the system.

## 1. Products: This resource type is for managing products in the store.

**2. Categories:** This resource type is for managing categories of products in the store. Categories can have child categories. A product may belong to multiple categories.

**3. Attributes –** This resource type is for managing the various attributes that a product may have. For example – A product may have different sizes and different colors. Therefore, ‘size’ and ‘color’ will be the attributes for the corresponding product.

**4. Variants:** This resource type is for managing Variants in the system. A product can have multiple Variants. A Variant has to necessarily belong to a Product. For example: A Leather Jacket which comes in two colors for 3 different sizes will have a total of 2x3 = 6 variants. Each variant has a unique SKU name which uniquely distinguishes itself from other variants. A variant has its corresponding price, available quantity and the attribute-value pair array. For example- A Leather Jacket of color ‘Brown’ and size ‘32’ will be linked to two Attribute Resource Instance – Value Resource Instance Pairs. In this case they will be:

{“color”:”Green”}

{“size”:”32”}

**5. Values** – This resource type is for storing the different values a product has for the different applicable attributes of the corresponding product. A value resource instance will always correspond to an attribute resource instance.

**6. Custom Attributes:** This resource type is for managing custom attributes in the system. A product may or may not have multiple custom attributes. A custom attribute resource has to necessarily belong to a product. A custom attribute resource instance has a name and the corresponding value.

## Difference between Attributes and Custom Attributes

Values of attributes will help in distinguishing a variant from other variants of a product. Ideally, attributes are those attributes which have different values for different variants. Whereas custom attributes are those attributes which are specific to a product, and cannot help in distinguishing a variant from another variant.

## C:\Users\Kanishk\Downloads\miltechin.com\How attributes and values are stored for variants.jpg

## Above is a pictorial representation about how attributes and their values are stored for a product variant.

## Requests

All POST and PUT requests are JSON encoded and must have content type of ‘application/json’.

## Responses

All response bodies are encoded in JSON.

### HTTP Response Status Codes

HTTP response status codes shall be used to indicate the result of a request.

Success codes:

* 200 OK - Request succeeded. Response included.
* 201 Created - Resource created. URL to the new resource in Location header.
* 204 No Content - Request succeeded, but no response body

Error codes:

* 401 Unauthorized- No authentication credentials provided or authentication failed
* 404 Not Found- Resource not found
* 415 Unsupported Media Type - POST/PUT request occurred without a application/json content type

**Query Parameters**

## Filtering Fields in the Response Object

Responses from the API can limit fields returned to only the fields required. For that, a **‘fields’** query parameter with a comma separated list of field names can be passed.

For example:

GET /api/v1/products?**fields=product\_id,name**

## Pagination

Pagination can be managed using the **‘per\_page’** and **‘page’** query parameters.

GET /api/v1/products?**per\_page=10&page=2**

## Sorting

The API offers sorting, using the **‘sort’** query parameter. The value of ‘sort’ should be a comma separated list of fields to sort by. A descending sort can be specified by prepending a **‘-’** to a field name.

For example :- To get recently added products, sorted in descending order of added\_date:

GET /api/v1/products?**sort=-created\_at**

## Products

Following are the attributes of the products in the online store:

| **Field** | **Type** | **Notes** |
| --- | --- | --- |
| Product\_Id | String | Automatically Generated ID of the Product. An auto incremented integer stored as a string. |
| Name | String | Name of the product. |
| Description | String | Description of the Product. |
| Quantity | Integer | Available quantity of the product in Store. 0 if the item is not available. |
| Availability | String | One of these: in\_stock, not\_available |
| Price | Decimal | Price of the Product. |
| Categories | String | Comma Separated List of Category ID’s to which the Product Belongs. |
| Variants | Resource | Denotes the various variants a product may have. |
| Attributes | Resource | Denotes the attributes which are used to distinguish variants from one another. |
| CustomAttributes | Resource | Denotes the custom attributes of a product. |
| Created\_at | Timestamp | Timestamp of the time when the product was created. |
| Last\_Updated\_At | Timestamp | Timestamp of the time when the product was last updated. |

### Listing products(Searching enabled using query parameters)

GET /api/v1/products

| **Query Parameter** | **Notes** |
| --- | --- |
| Product\_Id | Comma separated list of Product Id’s for corresponding Products which need to be displayed. |
| Name | Name of the Product. |
| Category\_id | Comma separated list of one or more Category ID’s. |
| Availability | One out of “in\_stock”,”not\_available” |

Sample Request to display all Products

GET /api/v1/products

200 OK

Content-Type: application/json

[

{

"product\_id": "29",

"name": "Leather Jacket",

"description": "Leather Jacket Sample Description",

"categories": "1,8,16",

"availability": "in\_stock",

"quantity": "5",

"price": "504.5",

"customattributes": {

"url": "https://www.onlinestore.com/api/v1/products/29/customattributes"

},

"Variants": {

"url": " https://www.onlinestore.com/api/v1/products/29/Variants"

},

"attributes": {

"url": "https://www.onlinestore.com/api/v1/products/29/attribues"

},

"last\_updated\_at": "2013-07-25 12:19:33",

"created\_at": "2013-07-17 18:58:44"

}

]

Sample Request for Products which have Category ID=8

GET /api/v1/products?**category\_id=8&fields=product\_id,name,categories,quantity,Variants**

200 OK

Content-Type: application/json

[

{

"product\_id": "29",

"name": "Leather Jacket",

"categories": "1,8,16",

"quantity": "5",

"Variants": {

"url": " https://www.onlinestore.com/api/v1/products/29/Variants"

}

}

]

### Getting details of a single product

GET /api/v1/product/product\_id

A Sample Request:

GET /api/v1/products/29?fields=product\_id,name,categories

200 OK

Content-Type: application/json

[

{

"product\_id": "29",

"name": "Leather Jacket",

"categories": "1,8,16"

}

]

### Updating a product

### No attribute is compulsory for updating the product. The attributes included in the JSON request object will get updated. The rest shall remain the same

PUT /api/v1/products/29

Sample Request:

PUT /api/v1/products

Content-Type: application/json

[

{

"name": "Leather Jacket",

"description": "Leather Jacket Sample Description Updated",

"categories": "1,8,16,24"

}

]

200 OK

Content-Type: application/json

[

{

"product\_id": "29",

"name": "Leather Jacket",

"description": "Leather Jacket Sample Description Updated",

"categories": "1,8,16,24",

"availability": "in\_stock",

"quantity": "5",

"price": "504.5",

"customattributes": {

"url": "https://www.onlinestore.com/api/v1/products/29/customattributes"

},

"Variants": {

"url": " https://www.onlinestore.com/api/v1/products/29/Variants"

},

"attributes": {

"url": "https://www.onlinestore.com/api/v1/products/29/attribues"

},

"last\_updated\_at": "2013-07-25 12:19:33",

"created\_at": "2013-07-17 18:58:44"

}

]

### Deleting a product

DELETE /api/v1/products/29

Content-Type: application/json

{

"deleted": true

}

### Creating a new product

POST /api/v1/products

Sample Request:

POST /api/v1/products

Content-Type: application/json

{

{

"name": "Leather Jacket 2",

"description": "Leather Jacket 2 Sample Description",

"categories": "1,8,16",

"availability": "in\_stock",

"quantity": "5",

"price": "504.5"

}

}

201 Created

Location: https://www.onlinestore.com/api/v1/products/30

Content-Type: application/json

{

{

"product\_id": "30",

"name": "Leather Jacket",

"description": "Leather Jacket Sample Description",

"categories": "1,8,16",

"availability": "in\_stock",

"quantity": "5",

"price": "504.5",

"customattributes": {

"url": "https://www.onlinestore.com/api/v1/products/30/customattributes"

},

"Variants": {

"url": " https://www.onlinestore.com/api/v1/products/30/Variants"

},

"attributes": {

"url": "https://www.onlinestore.com/api/v1/products/30/attribues"

},

"last\_updated\_at": "2013-07-17 18:58:44",

"created\_at": "2013-07-17 18:58:44",

}

}

## Categories

Following are the attributes of categories in the online store:

| **Field** | **Type** | **Notes** |
| --- | --- | --- |
| Category\_Id | String | Automatically Generated ID of the Category. An auto incremented integer stored as a string. |
| Name | String | Name of the category. |
| Description | String | Description of the category. |
| Parent\_Id | String | Used to store the category ID of the parent category |

### Listing categories

GET /api/v1/categories

| **Query Parameter** | **Notes** |
| --- | --- |
| Category\_Id | Comma separated list of Category Id’s for corresponding categories which need to be displayed. |
| Parent\_id | To Display Categories which have a specific Parent\_id |

Sample Request to display all Categories

GET /api/v1/categories

200 OK

Content-Type: application/json

[

{

"category\_id": "1",

"name": "Computer Accessories",

"description": "Computer Accessories Sample Description",

"parent\_id": "0"

},

{

"category\_id": "2",

"name": "Laptops",

"description": "laptops Sample Description",

"parent\_id": "0"

}

. . .

]

Sample Request for Categories which have Parent Category ID=2

GET /api/v1/categories?**parent\_id=2&fields=category\_id,name**

200 OK

Content-Type: application/json

[

{

"category\_id": "8",

"name": "Lenovo Laptops”

},

{

"category\_id": "9",

"name": "Dell Laptops”

}

]

### Getting details of a single category

GET /api/v1/categories/category\_id

A Sample Request:

GET /api/v1/categories/9?fields=name,description

200 OK

Content-Type: application/json

[

{

"name": "Dell Laptops",

"description": "Dell Laptops Description",

}

]

### Updating a category

PUT /api/v1/categories/9

| **Field** | **Notes** |
| --- | --- |
| Name | Name of the category. |
| Description | Description of the category. |
| Parent\_Id | Used to store the category ID of the parent category |

A Sample Request:

PUT /api/v1/categories/9

Content-Type: application/json

[

{

"parent\_id": "2",

"name": "Dell Laptops",

"description": "Dell Laptops Description Updated"

}

]

Response:

200 OK

Content-Type: application/json

[

{

"id": "9",

"parent\_id": "2",

"name": "Dell Laptops",

"description": "Dell Laptops Description Updated"

}

]

### Deleting a category

DELETE /api/v1/categories/29

Content-Type: application/json

{

"deleted": true

}

### Creating a new category

POST /api/v1/categories

Sample Request:

POST /api/v1/categories

Content-Type: application/json

{

"name": "Shoes",

"description": "Shoe Category Description",

"parent\_id": "1",

}

201 Created

Location: https://www.onlinestore.com/api/v1/categories/10

Content-Type: application/json

{

"id": "10",

"name": "Shoes",

"description": "Shoe Category Description",

"parent\_id": "1"

}

### 

## Variants

Following are the attributes of variants in the online store:

| **Field** | **Type** | **Notes** |
| --- | --- | --- |
| Variant\_Id | String | Automatically Generated ID of the Variant. An auto incremented integer stored as a string. |
| Variant\_SKU | String | SKU of the variant. |
| Price | String | Price of the Variant. |
| Quantity | Integer | Quantity available in Store of the Variant. |
| Product\_Id | String | Used to denote the parent product ID of the Variant. |
| Attributes | Object Array | Array of type object{“attribute\_id”:int, ”value\_id”:int} |

### Listing Variants of a product

GET /api/v1/products/id/variants

| **Query Parameter** | **Notes** |
| --- | --- |
| Variants\_Id | Unique ID of the Variant |
| Variant\_SKU | SKU of the variant. |

Sample Request to display all Variants of a Product with ID=2

GET /api/v1/products/2/variants

200 OK

Content-Type: application/json

[

{

"variant\_id": "9",

"product\_id": "2",

"variant\_SKU": "AHDM-2013",

"quantity": "5",

"price": "230",

"attributes": [

{

"attribute\_id": 15,

"value\_id": 17

},

{

"attribute\_id": 16,

"value\_id": 28

}

]

},

{

"variant\_id": "10",

"product\_id": "2",

"variant\_SKU": "AHDM-2014",

"quantity": "1",

"price": "250",

"attributes": [

{

"attribute\_id": 14,

"value\_id": 13

},

{

"attribute\_id": 19,

"value\_id": 31

}

]

}

]

### Getting details of a single VARIANT

GET /api/v1/products/2/variants/id

A Sample Request:

GET /api/v1/products/2/variants/9?fields=Variant\_SKU,quantity

200 OK

Content-Type: application/json

[

{

"VARIANT\_SKU": "AHDM-2013",

"Quantity": "5"

}

]

### Updating a VARIANT

PUT /api/v1/products/2/Variant/9

| **Field** | **Notes** |
| --- | --- |
| VARIANT\_SKU | SKU of the VARIANT. |
| Quantity | Quantity in stock of the VARIANT. |
| Price | Price of the VARIANT |
| Product\_Id | Used to denote the parent product ID of the VARIANT. |
| Attributes | Array of type object{“attribute\_id”:int, ”value\_id”:int} |

A Sample Request:

PUT /api/v1/products/2/Variants/9

Content-Type: application/json

[

{

"product\_id": "2",

"Variant\_SKU": "AHDM-2013",

"quantity": "1",

"price": "230",

"attributes": [

{

"attribute\_id": 15,

"value\_id": 17

},

{

"attribute\_id": 16,

"value\_id": 28

}

]

}

]

Response:

200 OK

Content-Type: application/json

[

{

"Variant\_id": "9",

"product\_id": "2",

"variant\_SKU": "AHDM-2013",

"quantity": "1",

"price": "230",

"attributes": [

{

"attribute\_id": 15,

"value\_id": 17

},

{

"attribute\_id": 16,

"value\_id": 28

}

]

}

]

### Deleting a VARIANT

DELETE /api/v1/products/2/Variants/9

Content-Type: application/json

{

"deleted": true

}

### Deleting all Variants of a Product

DELETE /api/v1/products/2/variants/

Content-Type: application/json

{

"deleted": true

}

### Creating a new Variant of a product

POST /api/v1/products/2/variants/

Sample Request:

POST /api/v1/products/2/variants

Content-Type: application/json

[

{

"product\_id": "2",

"Variant\_SKU": "AHDM-2015",

"quantity": "1",

"price": "280",

"attributes": [

{

"attribute\_id": 11,

"value\_id": 12

},

{

"attribute\_id": 13,

"value\_id": 14

}

]

}

]

Response:

200 OK

Content-Type: application/json

[

{

"Variant\_id": "11",

"product\_id": "2",

"variant\_SKU": "AHDM-2015",

"quantity": "1",

"price": "280",

"attributes": [

{

"attribute\_id": 11,

"value\_id": 12

},

{

"attribute\_id": 13,

"value\_id": 24

}

]

}

]

## Attributes

Following are the attributes of an attribute in the online store:

| **Field** | **Type** | **Notes** |
| --- | --- | --- |
| Attribute\_Id | String | Automatically Generated ID of the attribute. An auto incremented integer stored as a string. |
| Attribute\_Name | String | Name of the attribute. |

### Listing All Attributes in the Store

GET /api/v1/attributes

### Listing Attributes of a product

GET /api/v1/products/id/attributes

| **Query Parameter** | **Notes** |
| --- | --- |
| Attribute\_Id | Unique ID of the Attribute |
| Attribute\_Name | Name of the Attribute |

Sample Request to display all attributes of a Product with ID=2

GET /api/v1/product/2/attributes

200 OK

Content-Type: application/json

[

{

"attribute\_id": "9",

"attribute\_name": "Size"

},

{

"attribute\_id": "10",

"attribute\_name": "Color"

}

]

### Getting details of a single attribute

GET /api/v1/attributes/id

### Getting details of a single attribute of a product

GET /api/v1/products/2/attributes/id

A Sample Request:

GET /api/v1/products/2/attributes/9?fields=attribute\_name

200 OK

Content-Type: application/json

[

{

"attribute\_name": "Size"

}

]

### Updating an attribute

PUT /api/v1/attributes/2

| **Field** | **Notes** |
| --- | --- |
| Attribute\_Name | Name of the attribute. |

A Sample Request:

PUT /api/v1/attributes/9

Content-Type: application/json

[

{

"attribute\_name": "Colour"

}

]

Response:

200 OK

Content-Type: application/json

[

{

"attribute\_id": "10",

"attribute\_name": "Colour",

}

]

### Deleting an attribute

DELETE /api/v1/attributes/2

Content-Type: application/json

{

"deleted": true

}

### Creating a new attribute

POST /api/v1/attributes/

Sample Request:

POST /api/v1/attributes/

Content-Type: application/json

[

{

"attribute\_name": "Length"

}

]

Response:

200 OK

Content-Type: application/json

[

{

"attribute\_id": "11",

"attribute\_name": "Length"

}

]

## Values

Following are the attributes of values in the online store:

| **Field** | **Type** | **Notes** |
| --- | --- | --- |
| Value\_ID | String | Automatically Generated ID of the value. An auto incremented integer stored as a string. |
| Attribute\_Id | String | ID of the attribute to which the value belongs. |
| Label | String | Label of the value(The text used to show the value on the website) |
| Value | String | Actual value of the value resource instance |

### Listing All Values of an Attribute

GET /api/v1/attributes/2/values

| **Query Parameter** | **Notes** |
| --- | --- |
| Value\_Id | Unique ID of the Value instance |
| Label | Label of the value(The text used to show the value on the website) |

Sample Request to display all values of an attribute with ID=2

GET /api/v1/attributes/2/values

200 OK

Content-Type: application/json

[

{

"attribute\_id": "2",

"value\_id": "9",

"label": "White",

"value": "#FFFFFF"

},

{

"attribute\_id": "2",

"value\_id": "10",

"label": "Black",

"value": "#000000"

}

]

### Getting details of a single attribute

GET /api/v1/attributes/values/id

A Sample Request:

GET /api/v1/attributes/values/9?fields=label,value

200 OK

Content-Type: application/json

[

{

"label": "Black",

"value": "#000000"

}

]

### Updating a specific Value Resource

PUT /api/v1/attributes/values/2

A Sample Request:

PUT /api/v1/attributes/9

Content-Type: application/json

[

{

"label": "Red",

"value": "#FF0000"

}

]

Response:

200 OK

Content-Type: application/json

[

{

"attribute\_id": "2",

"value\_id": "9",

"label": "Red",

"value": "#FF0000"

}

]

### Deleting all values of an attribute instance

DELETE /api/v1/attributes/2/values/

Content-Type: application/json

{

"deleted": true

}

### Deleting a value

DELETE /api/v1/attributes/values/2

Content-Type: application/json

{

"deleted": true

}

### Creating a new value resource instance

POST /api/v1/attributes/2/values

Sample Request:

POST /api/v1/attributes/2/values

Content-Type: application/json

[

{

"attribute\_id": "2",

"label": "Blue",

"value": "#FF00FF"

}

]

Response:

200 OK

Content-Type: application/json

[

{

{

"attribute\_id": "2",

"value\_id": "11",

"label": "Blue",

"value": "#FF00FF"

}

}

]

## Custom Attributes

Following are the attributes of custom attributes in the online store:

| **Field** | **Type** | **Notes** |
| --- | --- | --- |
| Attribute\_ID | String | Automatically Generated ID of a custom attribute. An auto incremented integer stored as a string. |
| Product\_ID | String | ID of the product to which the custom attribute belongs. |
| Name | String | Label of the text shown for the custom attribute. |
| Value | String | Actual value of the custom attribute |

### Listing All Custom Attributes of a Product

GET /api/v1/products/2/customattributes

| **Query Parameter** | **Notes** |
| --- | --- |
| Attribute\_Id | Unique ID of the Custom Attribute |
| Name | Name of the Custom Attribute |

Sample Request to display all custom attributes of a product with ID=2

GET /api/v1/products/2/customattributes

200 OK

Content-Type: application/json

[

{

"attribute\_id": "9",

"parent\_id": "2",

"name": "Sample Name",

"value": "Sample Value"

}

]

### Getting details of a single custom attribute

GET /api/v1/products/2/customattributes/9

A Sample Request:

GET /api/v1/products/2/customattributes/9?fields=name,value

200 OK

Content-Type: application/json

[

{

"name": "Sample Name",

"value": "Sample Value"

}

]

### Updating a specific custom attribute

PUT /api/v1/products/2/customattributes/9

| **Field** | **Notes** |
| --- | --- |
| Product\_ID | ID of the product to which the custom attribute belongs. |
| Name | Label of the text shown for the custom attribute. |
| Value | Actual value of the custom attribute |

A Sample Request:

PUT /api/v1/products/2/customattributes/9

Content-Type: application/json

[

{

"parent\_id": "2",

"name": "Sample Name Updated",

"value": "Sample Value Updated"

}

]

Response:

200 OK

Content-Type: application/json

[

{

"attribute\_id": "9",

"parent\_id": "2",

"name": "Sample Name Updated",

"value": "Sample Value Updated"

}

]

### Deleting all custom attributes of a product

DELETE /api/v1/products/2/customattributes/

Content-Type: application/json

{

"deleted": true

}

### Deleting a specific custom attribute

DELETE /api/v1/products/2/customattributes/9

Content-Type: application/json

{

"deleted": true

}

### Creating a new custom attribute for a product

POST /api/v1/products/2/customattributes/

Sample Request:

POST /api/v1/products/2/customattributes/

Content-Type: application/json

[

{

"parent\_id": "2",

"name": "Sample Name",

"value": "Sample Value"

}

]

Response:

200 OK

Content-Type: application/json

[

{

"attribute\_id": "10",

"parent\_id": "2",

"name": "Sample Name",

"value": "Sample Value"

}

]