## Online Store API Design

## Base URL

The base URL of the API is [**https://localhost:8080/onlinestore**](https://localhost:8080/onlinestore)

## 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.

Username : nitish  
Password : nitish

## Authentication

Authentication shall happen using HTTP Basic Authentication over SSL. The user will have to provide his username and password as authorization credentials.  
For every API request the headers will contain Basic Authorization data with username and password encoded in base64.

**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 backend System.

There are four 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”}

## 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
* 417 Expectation Failed – Occurs when the expected operation does not goes successful.

## 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. |
| 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 | String | URL to fetch various variants a product may have. |
| Attributes | String | URL to fetch the attributes which are used to distinguish variants from one another. |
| 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 & Path Params)

GET /onlinestore/products

| **Query Parameter** | **Notes** |
| --- | --- |
| Product\_Id | Provided in path param, 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 /onlinestore/products

200 OK

Content-Type: application/json

[

{

"productId": 1,

"name": "Dell Inspiron Laptop",

"description": "Sample Dell Inspiron Laptop Description",

"quantity": 20,

"availability": "in\_stock",

"price": 31000,

"categories": "1,6",

"attributes": "0",

"createdAt": "2015-07-25 12:19:33.0",

"lastUpdatedAt": "2015-07-29 18:19:33.0",

"attributes\_url": "localhost:8080/onlinestore/products/1/attributes",

"variants\_url": "localhost:8080/onlinestore/products/1/variants",

"PRODUCT\_URL": "localhost:8080/onlinestore/products/"

}

]

Sample Request for Products which have name = Mossad

GET onlinestore/products?name=Mossad

200 OK

Content-Type: application/json

[

{

"productId": 3,

"name": "Mossad",

"description": "Sample Mossad book Description",

"quantity": 120,

"availability": "in\_stock",

"price": 450,

"categories": "2",

"attributes": "3",

"createdAt": "2015-08-15 12:19:33.0",

"lastUpdatedAt": "2015-08-29 18:19:33.0",

"attributes\_url": "localhost:8080/onlinestore/products/3/attributes",

"variants\_url": "localhost:8080/onlinestore/products/3/variants",

"PRODUCT\_URL": "localhost:8080/onlinestore/products/"

}

]

### Getting details of a single product

GET /onlinestore/product/product\_id

A Sample Request:

GET /onlinestore/products/2

200 OK

Content-Type: application/json

[

{

"productId": 2,

"name": "HP Laptop",

"description": "Sample HP Laptop Description",

"quantity": 0,

"availability": "not\_available",

"price": 32000,

"categories": "1,6",

"attributes": "0",

"createdAt": "2015-04-25 12:19:33.0",

"lastUpdatedAt": "2015-07-29 18:19:33.0",

"attributes\_url": "localhost:8080/onlinestore/products/2/attributes",

"variants\_url": "localhost:8080/onlinestore/products/2/variants",

"PRODUCT\_URL": "localhost:8080/onlinestore/products/"

}

]

### Updating a product

PUT /onlinestore/products

Sample Request: (The api returns the updated resource)

PUT /onlinestore/products

Content-Type: application/json

[

{

"productId": 1,

"name": "Dell Inspiron Laptop",

"description": "Sample Dell Inspiron Laptop Description",

"quantity": 20,

"availability": "in\_stock",

"price": 27000,

"categories": "1,6",

"attributes": "0",

"createdAt": "2015-07-25 12:19:33.0",

"lastUpdatedAt": "2015-07-29 18:19:33.0",

"attributes\_url": "localhost:8080/onlinestore/products/1/attributes",

"variants\_url": "localhost:8080/onlinestore/products/1/variants",

"PRODUCT\_URL": "localhost:8080/onlinestore/products/"

}

]

200 OK

Content-Type: application/json

[

{

"productId": 1,

"name": "Dell Inspiron Laptop",

"description": "Sample Dell Inspiron Laptop Description",

"quantity": 20,

"availability": "in\_stock",

"price": 27000,

"categories": "1,6",

"attributes": "0",

"createdAt": "2015-07-25 12:19:33.0",

"lastUpdatedAt": "2015-07-29 18:19:33.0",

"attributes\_url": "localhost:8080/onlinestore/products/1/attributes",

"variants\_url": "localhost:8080/onlinestore/products/1/variants",

"PRODUCT\_URL": "localhost:8080/onlinestore/products/"

}

]

### Deleting a product( Api returns the deleted resource)

DELETE /onlinestore/products/16

Content-Type: application/json

{

"productId": 16,

"name": "Induction cooker 3",

"description": "Sample Induction Cooker 3 Description",

"quantity": 45,

"availability": "in\_stock",

"price": 2400,

"categories": "3,9",

"attributes": "0",

"createdAt": "2015-07-25 12:19:33.0",

"lastUpdatedAt": "2015-07-29 18:19:33.0",

"attributes\_url": "localhost:8080/onlinestore/products/16/attributes",

"variants\_url": "localhost:8080/onlinestore/products/16/variants",

"PRODUCT\_URL": "localhost:8080/onlinestore/products/"

}

### Creating a new product

POST /onlinestore/products

Sample Request:

POST /onlinestore/products

Content-Type: application/json

{

"name": "Induction cooker 2",

"description": "Sample Induction Cooker 2 Description",

"quantity": 45,

"availability": "in\_stock",

"price": 2400,

"categories": "3,9",

"attributes": "0",

"createdAt": "2015-07-25 12:19:33.0",

"lastUpdatedAt": "2015-07-29 18:19:33.0",

"attributes\_url": "localhost:8080/onlinestore/products/10/attributes",

"variants\_url": "localhost:8080/onlinestore/products/10/variants",

"PRODUCT\_URL": "localhost:8080/onlinestore/products/"

}

201 Created

Location: localhost:8080/onlinestore/products/15

Content-Type: application/json

{

"productId": 15,

"name": "Induction cooker 2",

"description": "Sample Induction Cooker 2 Description",

"quantity": 45,

"availability": "in\_stock",

"price": 2400,

"categories": "3,9",

"attributes": "0",

"createdAt": "2015-07-25 12:19:33.0",

"lastUpdatedAt": "2015-07-29 18:19:33.0",

"attributes\_url": "localhost:8080/onlinestore/products/15/attributes",

"variants\_url": "localhost:8080/onlinestore/products/15/variants",

"PRODUCT\_URL": "localhost:8080/onlinestore/products/"

}

## 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. |
| 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 /onlinestore/categories

| **Query Parameter** | **Notes** |
| --- | --- |
| Category\_Id | Category Id for category which need to be displayed.( Passed as Path Params) |
| Parent\_id | To Display Categories which have a specific Parent\_id |

Sample Request to display all Categories

GET /onlinestore/categories

200 OK

Content-Type: application/json

[

{

"categoryId": 1,

"name": "Electronics",

"description": "Sample Electronics Description",

"parentId": -1

},

{

"categoryId": 2,

"name": "Books",

"description": "Sample Books Description",

"parentId": -1

},

{

"categoryId": 3,

"name": "Home and Kitchen",

"description": "Sample Home and Kitchen Description",

"parentId": -1

},

{

"categoryId": 4,

"name": "Music",

"description": "Sample Music Description",

"parentId": -1

},

. . .

]

Sample Request for Categories which have Parent Category ID=2

GET /onlinestore/categories?**parentId=1**

200 OK

Content-Type: application/json

[

{

"categoryId": 5,

"name": "Mobiles",

"description": "Mobiles Description",

"parentId": 1

},

{

"categoryId": 6,

"name": "Computers",

"description": "Sample Computers Description",

"parentId": 1

}]

### Getting details of a single category

GET /onlinestore/categories/categoryId

A Sample Request:

GET /online/categories/1

200 OK

Content-Type: application/json

[

{

"categoryId": 1,

"name": "Electronics",

"description": "Sample Electronics Description",

"parentId": -1

}

]

### Updating a category (The api returns updated category)

PUT /onlinestore/categories/9

| **Field** | **Notes** |
| --- | --- |
| CategoryId | Id of the updating category |
| 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 /onlinestore/categories/9

Content-Type: application/json

[

{

"categoryId": 11,

"name": "Clothings & Apparels",

"description": "Sample Apparels Description",

"parentId": -1

}

]

Response:

200 OK

Content-Type: application/json

{

"categoryId": 11,

"name": "Clothings & Apparels",

"description": "Sample Apparels Description",

"parentId": -1

}

### Deleting a category(API returns deleted resource)

DELETE /onlinestore/categories/12

Content-Type: application/json

{

"categoryId": 12,

"name": "Men Clothings",

"description": "Sample Men Clothings Description",

"parentId": 11

}

### Creating a new category(API returns newly created category resource)

POST /onlinestore/categories

Sample Request:

POST /onlinestore/categories

Content-Type: application/json

{

"name": "Women Clothings",

"description": "Sample Women Clothings Description",

"parentId": 11

}

201 Created

Location: https://localhost:8080/onlinestore/categories/13

Content-Type: application/json

{

"categoryId": 13,

"name": "Women Clothings",

"description": "Sample Women Clothings Description",

"parentId": 11

}

### 

## 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. |
| Product\_ Id | String | Product ID for which this variant belong to. |
| Variant\_SKU | String | SKU of the variant. |
| Price | String | Price of the Variant. |
| Quantity | Integer | Quantity available in Store of the Variant. |
| Attribute\_id | Integer | Id of associated Attribute of this variant |
| Attribute\_value | String | Value of associated Attribute of this variant |

### Listing Variants of a product

GET /onlinestore/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 /onlinestore/products/9/variants

200 OK

Content-Type: application/json

[

{

"variantId": 5,

"productId": 9,

"variantSKU": "APRL-2015/9/S1",

"quantity": 50,

"price": 1299,

"attributeId": 1,

"attributeValue": "30"

},

{

"variantId": 6,

"productId": 9,

"variantSKU": "APRL-2015/9/S2",

"quantity": 50,

"price": 1299,

"attributeId": 1,

"attributeValue": "32"

},

{

"variantId": 7,

"productId": 9,

"variantSKU": "APRL-2015/9/S3",

"quantity": 50,

"price": 1299,

"attributeId": 2,

"attributeValue": "BLUE"

}

. . .

]

### Getting details of a single VARIANT

GET /onlinestore/products/9/variants?variantId=<variantId>

A Sample Request:

GET /onlinestore/products/9/variants?variantId=6

200 OK

Content-Type: application/json

[

{

"variantId": 6,

"productId": 9,

"variantSKU": "APRL-2015/9/S2",

"quantity": 50,

"price": 1299,

"attributeId": 1,

"attributeValue": "32"

}

]

### Updating a VARIANT(API returns the updated Resource)

PUT /onlinestore/products/<productID>/Variant/<variantID>

| **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. |
| Attribute\_Id | ID of attribute for corresponding attribute in string format. |
| Attribute\_value | Value of attribute for corresponding attribute in string format. |

A Sample Request: (All fields are mandatory to provide during update request)

PUT /onlinestore/products/9/Variants/5

Content-Type: application/json

{

"variantId": 5,

"productId": 9,

"variantSKU": "APRL-2015/9/S1",

"quantity": 50,

"price": 1299,

"attributeId": 1,

"attributeValue": "40"

}

Response:

200 OK

Content-Type: application/json

{

"variantId": 5,

"productId": 9,

"variantSKU": "APRL-2015/9/S1",

"quantity": 50,

"price": 1299,

"attributeId": 1,

"attributeValue": "40"

}

### Deleting a VARIANT(API returns the deleted resource)

DELETE /onlinestore/products/9/Variants/10

Content-Type: application/json

{

"variantId": 5,

"productId": 9,

"variantSKU": "APRL-2015/9/S1",

"quantity": 50,

"price": 1299,

"attributeId": 1,

"attributeValue": "40"

}

### Creating a new Variant of a product(API returns newly created resource)

POST /onlinestore/products/9/variants/

Sample Request:

POST /onlinestore/products/9/variants

Content-Type: application/json

{

"productId": 9,

"variantSKU": "APRL-2015/9/S3",

"quantity": 50,

"price": 1299,

"attributeId": 1,

"attributeValue": "38"

}

Response:

201 CREATED

Content-Type: application/json

{

"variantId": 11,

"productId": 9,

"variantSKU": "APRL-2015/9/S3",

"quantity": 50,

"price": 1299,

"attributeId": 1,

"attributeValue": "38"

}

## 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 /onlinestore/attributes

### Listing Attributes of a product

GET /online/products/<productId>/attributes

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

GET /onlinestore/products/9/attributes

200 OK

Content-Type: application/json

[

{

"attributeId": 1,

"attributeName": "size"

},

{

"attributeId": 2,

"attributeName": "color"

}

]

### Updating an attribute(API returns the updated resource)

PUT /onlinestore/attributes/2

| **Field** | **Notes** |
| --- | --- |
| Attribute\_ID | Id of the attribute to be updated |
| Attribute\_Name | Name of the attribute. |

A Sample Request:

PUT /onlinestore/attributes/9

Content-Type: application/json

{

"attributeId": 3,

"attributeName": "book\_type1"

}

Response:

200 OK

Content-Type: application/json

{

"attributeId": 3,

"attributeName": "book\_type1"

}

### Deleting an attribute(API Returns deleted Resource)

DELETE /onlinestore/attributes/3

Content-Type: application/json

{

"attributeId": 3,

"attributeName": "book\_type1"

}

### Creating a new attribute (API Returns newly created resource)

POST /onlinestore/attributes/

Sample Request:

POST /onlinestore/attributes/

Content-Type: application/json

{

"attributeName": "book\_type"

}

Response:

201 CREATED

Content-Type: application/json

Location: localhost:8080/onlinestore/attributes/4

{

"attributeId": 4,

"attributeName": "book\_type"

}