

API Structure:

- api/{version}/{microservice_name}/{module_name}/{operation/action}
 - ex: api/v2/service_setup/service_types/get_service_types
 - ex: api/v2order_orchestrator/webstore_order/create_order
 - ex: api/v2/products_management/products/list

Requests/Response Guideline:

• We use only two types of API requests

- GET
- POST

• "account_id" is a mandatory query param for any type of request.

• "Authorization" is a mandatory header attribute in any type of request.

• GET:

- Only Use **Query params** a No path params or body.
 - **example:**
 - **Use:** {domain}/api/v2/products_management/products/list?account_id=117&perPage=20&pageNo=1
 - **Don't Use:** {domain}/api/v2/products_management/products/list/11720/1

• POST:

- Only Use Query params and No path params
- The post method should be of **JSON** format.
- Should Include:
 - data : data consists of post which is related to entity data you want to send.
 - meta_data: To send any metadata (Future use)
 - event_data: Metadata related to request.
 - example:

```
{
  "data":{
    "customer_id":"4",
    "account_id":"117",
    "mobile":"3786543678654",
    "email":"sachiraju@gmail.com",
    "name":"Machiraju",
    "notes":"test",
    "customer_type":"consumer"
  },
  "meta_data":{
    "user_id":"1"
  },
}
```

```

    "event_data":{
    }
}

```

• Responses:

- Response should be in JSON format
- Consists of:
 - **status** : true|false (return true if desired output is achieved else false)
 - **message**(Only for POST response) : Should contain a message in string format, which is displayed in frontend (as notification, alert, popup etc.,, for every response)
 - **error_obj** : For every false response, error_obj dict should be returned with error_code and description.(With detailed explanation of error coide)
 - example for status "true":

```

{
  "status": true,
  "data": {
    "customer_type": "consumer",
    "_id": "5e2f008a6b936623f31436af",
    "account_id": 117,
    "mobile": "3786543678654",
    "__v": 0,
    "customer_id": "4",
    "email": "sachiraju@gmail.com",
    "name": "MAchiraju",
    "notes": "asdfg",
    "updated_by": "1",
    "ud": "2020-02-03T08:39:56.357Z"
  },
  "message": "Customer is sucessfully updated"
}

```

- example for status "false":

```

{
  "status": false,
  "message": "Email already exists",
  "error_obj": {
    "error_code": "IDP_0001",
    "description": "Duplicate Entery trying to insert the value in acolumns
which is defined as unique"
  }
}

```

```
}
```

• **NOTE:**

- **IDP_0001** is defined in the backend in error_codes file. These codes are used for tracing and easy understanding of what went wrong in the centralized logs.

• **Api's with pagination should return the response in the following formart**

- **status:** true|false
- **data :** Array of objects
- **pagination:** Dict with following values:
 - **total_pages :** Max no. of pages
 - **per_page :** No. of objects per page
 - **current_page :** Page number of current page
 - **next_page :** Page number of next page
 - **total_rows :** Total no of objects present for that account in that table or collection
 - example:

```
{
  "pagination": {
    "total_pages": 10,
    "per_page": 20,
    "current_page": 10,
    "next_page": 2,
    "previous_page": 1,
    "total_rows":100
  }
}
```

- Full response example:
- api_request: **/api/v2/products_management/products/list?account_id=117&perPage=20&pageNo=1**

```
{
  "status": true,
  "data": [
    {
      "_id": "5e32dbf5ff17cc69d7e12aba",
      "account_id": "117",
      "sku_id": "venkat",
      "__v": 0,
      "category_id": "10014",
      "front_view": "http://ecx.images-
amazon.com/images/I/41IkLScBMHL._SL75_.jpg",
```

```

"package_height": "1.98",
"package_length": "4.72",
"package_weight": "1.76",
"package_width": "3.94",
"sub_category_id": 20193,
"title": "Safe-0-Kid- Pack of 8 Single Key Multifunctional Lock,
Dual Color Cabinet/Drawer Lock- Green",
"center_inventory": {
  "channels": [
    "9"
  ],
  "quantity": "999"
},
"marketplace_info": {
  "9": {
    "marketplace_id": "B07H9ss959QB",
    "parent_marketplace_id": "B07PFBHS9S",
    "last_sync_date": "2020-01-30T13:40:29.527Z"
  }
},
"pricing": {
  "9": {
    "listing_price": "999"
  }
},
"ud": "2020-01-30T13:36:53.117Z",
"is_posstore_enabled": true,
"pos_store_info": {
  "b1ae5dd5-20a5-48f8-a51b-6ef2f23a83b0": {
    "is_active": true,
    "listing_price": 0,
    "inventory": 100,
    "channel_id": "b1ae5dd5-20a5-48f8-a51b-6ef2f23a83b0"
  }
}
},
{
  "_id": "5e33c245ff17cc69d7e1b00e",
  "account_id": "117",
  "sku_id": "SOK9LOC37P006",

```

```

    "__v": 0,
    "category_id": "10014",
    "front_view": "http://ecx.images-
amazon.com/images/I/41IkLScBMHL._SL75_.jpg",
    "package_height": "1.97",
    "package_length": "4.72",
    "package_weight": "1.76",
    "package_width": "3.94",
    "sub_category_id": 20193,
    "title": "Safe-0-Kid- Pack of 8 Single Key Multifunctional Lock,
Dual Color Cabinet/Drawer Lock- Green",
    "decenter_inventory": {
      "1": {
        "quantity": "111222"
      }
    },
    "marketplace_info": {
      "1": {
        "marketplace_id": "B07H9ss959QB",
        "parent_marketplace_id": "B07PFBHS9S"
      }
    },
    "pricing": {
      "1": {
        "listing_price": "3011"
      }
    },
    "ud": "2020-01-31T05:59:33.785Z",
    "is_posstore_enabled": true,
    "pos_store_info": {
      "b1ae5dd5-20a5-48f8-a51b-6ef2f23a83b0": {
        "is_active": true,
        "listing_price": 0,
        "inventory": 100,
        "channel_id": "b1ae5dd5-20a5-48f8-a51b-6ef2f23a83b0"
      }
    }
  },
],

```

```
"pagination": {  
  "total_pages": 1,  
  "per_page": 20,  
  "current_page": 1,  
  "next_page": 0,  
  "previous_page": 0,  
  "total_rows": 2  
}
```

• **Note:**

- Use proper Http status codes when sending response:
 - 200: For success response
 - 200: For data not found
 - 406: For validation errors
 - 500: Internal services errors
 - 4XX: For client side errors
 - 5XX: For server errors

- For reference:

<https://www.restapitutorial.com/httpstatuscodes.html>

Request Validation:

- Every request coming to microservice must be validated to check client errors.
- Any client errors should arise in routes these client error should not arise in application logic
- For payload or query_params validations in backend use `json-schema-validator` which follows openapi 3.0 specifications.
 - Ref <https://www.npmjs.com/package/express-json-validator-middleware>