

SIGNUP MODULE

Standard Status Codes

- 404 - not found
- 200 - success okay
- 400 - Client Error
- 500 - Server Error
- 1001 - Session Expiry (With Standard Message)
- 1003 - Empty Data.

{baseurl} = <https://appdomain/webservices/v1>

login - loginapi (2 api's)

1. Signup Api

Url: <https://{baseurl}/authentication/signup>

Method Type: POST

Header Params:

1. Content type (application/json)
2. Accept encoding (gzip) - *(Backend team will check if it works for node)*

Request Body

1. Name = string
2. Password = passwordval
3. Phone = number (17 numbers max - international)

Post API :

Trigger sms api.- Vendor Specific

Response Body:

1. { msg: <errormessage>,
 statuscode: <codenum> }

Custom Status Codes:

1001:

2. Update Profile API

Url: <https://{baseurl}/authentication/profileapi>

Method Type: POST

Header Params:

1. Content type (multipart/formdata)
2. Accept encoding (gzip) - *(Backend team will check if it works for node)*

Request Body (key value pair seperated by boundary)

1. Name = string
2. Email = email id.
3. Picture = uploaded picture.
4. Other additional parameters.

Response Body:

2. { msg: <errormessage>,
statuscode: <codenum> }

Custom Status Codes:

1001:

3. Change Password

Url: <https://{baseurl}/authentication/changepassword>

Method Type: POST

Header Params:

1. Content type (application/json)
2. Accept encoding (gzip) - *(Backend team will check if it works for node)*

Request Parameters

1. Old Password= string
2. NewPassword= string
3. AccessToken = <code> // Required for security.

Response Body:

3. { msg: <errormessage>,
statuscode: <codenum> }

Custom Status Codes: 1001:

4. Forgot Password

Url: <https://{baseurl}/authentication/forgotpassword>

Method Type: POST

Header Params:

1. Content type (application/json)
2. Accept encoding (gzip) - (*Backend team will check if it works for node*)

Request Parameters

1. Phone number - 17 digit

Response Body:

```
{ token: <OTP> // Generated by Post Api call of sms api.  
  msg: <errormessage>,  
  statuscode: <codenum> }
```

Post API

Generate OTP and trigger sms api to send otp to the user's phone number.

Custom Status Codes:

1001:

5.Login API

Url: <https://{baseurl}/authentication/login/>

Header Params:

1. Username
2. Password
3. Response_type - Code
4. Clientid

Method Type: GET

Response Body:

1. {msg: <errormessage>,
 statuscode: <codenum> }

Custom Status Codes:

1001:

6. Transaction Reports

Url: <https://{baseurl}/transaction/reports>

Header Params:

1. Username
2. Password
3. Response_type - Code
4. Clientid

Query Params

1. Start Date
2. End Date
3. Transaction ID (optional)

Method Type: POST

Response Body:

2. {msg: <errormessage>,
statuscode: <codenum>,
Response: [{
 'id': '',
 'Transaction details': [],
 'Customer details': [],
 'Shipment details': [],
 'Product details': []
 }]
 }

Custom Status Codes:

1001:

6. Cart Module

Url: <https://{baseurl}/cartmodule/post>

Header Params:

1. User id

Query Params

1. Product id.
2. quantity.

Method Type: POST

Response Body:

```
3. {msg: <errorMessage>,
    statuscode: <codenum>,
    Response: [{
        'cartId': "",
        'Product details': [],
    }
    ]
}
```

Custom Status Codes:

1001:

Additional Modules on cart can be written to add and delete products.

Url: <https://{baseurl}/cartmodule/add> (Addition of products)

Url: <https://{baseurl}/cartmodule/delete> (Deletion of Products)

7. Payment Module

Url: <https://{baseurl}/payment>

Header Params:

1. User id

Query Params

1. Cartid.

Method Type: POST

Response Body:

```
4. {msg: <errormessage>,
    statuscode: <codenum>,
    Response: [{
        'cartId': '',
        'Product details': [],
        'amount': '<amountvalue>',
        'Paymentmethods': [{
            'Paymentmethod1': '<urltopaymentgateway>',
            //if we are handling or we can direct it to billdesk or ccavenue.
        }]
    }]
```

```
    }]
```

```
  }
```

Custom Status Codes:

1001:

8. Order History Listing

Url: <https://{baseurl}/orderdetails>

Header Params:

1. User id

Method Type: POST

Response Body:

```
{
    msg: <errormessage>,
    statuscode: <codenum>,
    Response: [{
        'orderid': '<id>',
        'Customerid': "",
        'Username': "",
    }]
}
```

Custom Status Codes:

1001:

9. Order Details

Url: [https://{baseurl}/orderdetails/key=\\$value](https://{baseurl}/orderdetails/key=$value)

Header Params:

1. User id

Query Params:

1. Order id: <value>

Method Type: GET

Response Body:

```
{
    msg: <errormessage>,
    statuscode: <codenum>,
    Response: {
        'orderid': '<id>',
```

```
        'Customerid': "",
        'Username': "",
    }
}
```

Custom Status Codes:

1001:

9. Track Shipment Details

Url: [https://{baseurl}/trackdetails/key=\\$value](https://{baseurl}/trackdetails/key=$value)

Header Params:

1. User id

Query Params:

2. Order id: <value>

Method Type: GET

Response Body:

```
{
    msg: <errormessage>,
    statuscode: <codenum>,
    Response: {
        'shipmentid': '<id>',
        'deliverydate': '<date>'
    }
}
```

Custom Status Codes:

1001:

10. Category Listing

Url: <https://{baseurl}/categorydetails>

Header Params:

1. User id

Method Type: GET

Response Body:

```
{
```

```

        msg: <errormessage>,
        statuscode: <codenum>,
        Response: {
            idcategory: '<id>',
            categoryname: '<name>',
            description: '<value>'
        }
    }
}

```

Custom Status Codes:

1001:

11. Product Listing (from category)

Url: [https://{baseurl}/categorydetails/category=\\$value&page=value](https://{baseurl}/categorydetails/category=$value&page=value)

Header Params:

1. User id

Query Params:

1. Value = '<all/categoryid>'

2. Page = pagenumber.

Method Type: GET

Response Body:

```

{
    msg: <errormessage>,
    statuscode: <codenum>,
    Response:[ {
        'productid': '<id>',
        'sku': '<name>',
        'price': '<value>',
        'image': '<imgurl>',
        'name': '<name>'
    }
]
}

```

Custom Status Codes:

1001:

12. Product details.

Url: [https://{baseurl}/productdetails/\\$key=\\$value](https://{baseurl}/productdetails/$key=$value)

Header Params:

1. User id

Query Params:

1. Product id: '<id>'

Method Type: GET

Response Body:

```
{
    msg: <errormessage>,
    statuscode: <codenum>,
    Response: {
        'productid': '<id>',
        'sku': '<name>',
        'price': '<value>',
        'image': '<imgurl>',
        'name': '<name>'
    }
}
```

Custom Status Codes:

1001:

12. Store Details

Url: <https://{baseurl}/storedetails/?key=value>

Header Params:

1. User id

Query Params:

3. Value = '<all/storeid>'
4. Page = pagenumber.

Method Type: GET

Response Body:

```
{
    msg: <errormessage>,
    statuscode: <codenum>,
    Response:[ {
        storeid: '<id>',
        'storename': '<name>',
        'storeaddress': '<value>',
        'latitude': '<lat>',
    }
]
```

```
        'longitude': '<long>'
    }}
}
```

Custom Status Codes:

1001:

NOTE: THESE API'S CAN BE BUILT USING THE FOLLOWING STACK

1. Nodejs + Mongoddb + laravel (cms)
Or
2. MYSQL + PHP (codeigniter or laravel)