User Services:

1. adduser:

input Json:

{

1223 (int validation key)

}

out put:

{

“fdsadf” (string client key)

}

errors:

401: unauthorized

500: internal server error

2. registeruser: (Input validations are not checked at server side like does phone number has 10 digits etc expected client to check)

input:

{

“name” = “phani”

**“**email” = “XXX@gmail.com”

**“**phno” = ”4534663456”

}

output:

{

“success”

}

errors:

500: internal server error

3. login:

input Json:

{

**“**email” = “XXX@gmsil.com”

}

out put:

{

“success”

}

errors:

401: unauthorized (with proper message like user doesn’t exist etc)

4. getshoplist: Gives shoplist which are around 2 kms for a given location

input:

{

“latitude”: 12.5464

“longitude”: 17.4456745

}

output:

{

[

“name” : “shopname”

“id” : 32435

“location”:{

“latitude”: 12.5464

“longitude”: 17.4456745

},

“name” : “shopname”

“id” : 32435

“location”:{

“latitude”: 12.5464

“longitude”: 17.4456745

}

]

}

errors:

401: unauthorized

500: internal server error

5. connect: Services connect to particular shop

input:

{

12333 (id of the shop which you want to connect)

}

output:

{

“success”

}

errors:

204: No shop found with specified Id.

401: unauthorized

500: internal server error

6. getproduct: Services to get product details

input:

{

“type”: “productid” (can be productId / varianceId)

“id” : 122334 (id of the product or variance which you want )

}

output: will return product details. If input type is varainaceId the variance list will contain only one object

{

“id”: 1223

“name”:”kitkat”

“brandId”:23

“varianceList”:[

{

“id”: 1223

“name”:”kitkat”

“price”:23

“quantity”: 12 (quantity available)

},

{

“id”: 1223

“name”:”kitkat”

“price”:23

“quantity”: 12 (quantity available)

}

]

}

errors:

204: No product found with specified Id.

401: unauthorized

500: internal server error

6. packproducts: Services connect to particular shop

input:

{

“type”: “productid” (can be productId / varianceId)

“id” : 122334 (id of the product or variance which you want )

}

output: will return product details. If input type is varainaceId the variance list will contain only one object

{

“id”: 1223

“name”:”kitkat”

“brandId”:23

“varianceList”:[

{

“id”: 1223

“name”:”kitkat”

“price”:23

“quantity”: 12 (quantity available)

},

{

“id”: 1223

“name”:”kitkat”

“price”:23

“quantity”: 12 (quantity available)

}

]

}

errors:

401: unauthorized

412: Pre condition failed

204: if no product is found with the id

500: internal server error

6. packproducts: Services to pack products

input:

{

“state”: “INSERT” (can be INSERT / UPDATE/DELETE)

“products” : {[

{

“varainceid”: 1223

“quantity”:”kitkat”

“price”:23

},

{

“varainceid”: 1223

“quantity”:”kitkat”

“price”:23

}

]}

}

output:

{

“success”

}

errors:

401: unauthorized

412: Pre condition failed

204: no order is found to update or delete

500: internal server error

7. submitorder: Services to finalise order

input:

{

“satus”: “INSERT” (can be FORPAYMENT / FORHOMEDELIVERYPAYMENT /FORHOMEDELIVERY /FORDELIVERY)

“address” : “”

“latitude” :12.23424 (delivery latitude)

“longitude” :34.435436 (delivery longitude)

“amount” : 365.78 (order amount)

“refnumber” : “23r2

}

For home delivery address, latitude and longitude are must. For online payments reference number is must

output:

{

“success”

}

errors:

401: unauthorized

412: Pre condition failed

204: no order is found to update or delete

500: internal server error