



# INTEGROMAT - SHOPIFY INTEGRATION API

V 1.5

24/02/2021

URL : <https://sms.costbucket.com/api/v2/posapi/>

Method / Action Name: **makeinvoicewebintegration**

Parameters:

JSON as in following format

```
{
  "param": [{
    "orderlines": [{
      "stockid": "CRD",
      "price": 3.45,
      "Qty": 2,
      "description": "Corn Dog"
      "discountAmount": "0"
    }, {
      "stockid": "6PP",
      "price": 7.99,
      "Qty": 1,
      "description": "6 inch Pepperoni Pizza"
      "discountAmount": "0.80"
    }],
    "custName": "CASH"
  }],
  "command": "makeinvoicewebintegration",
  "api": "live_mDhTrVewN5vMILwnc9qn60AOc",
  "username": "restaurant",
  "taxTotal": 0,
  "payment": [{
    "type": "CARD",
    "refno": "12345",
    "maskedCardNum": "1xxxxxxxxxxx0000",
    "amount": 10
  },
  {
    "type": "CASH",
    "refno": "12345",
    "cashTendered": 2,
    "amount": 1.44
  }],
  "filename": "T120201127120243"
}
```

### Description:

This method receives JSON from Shopify and creates an invoice into the system.

custName - Customer code

Price - mentioned in orderlines is unit price for 1 quantity

Filename - with "T1" - YYYYMMDDHHMMSS from Order confirmation date

### Returns:

```
{
  "data": {
    "status": "success",
    "invoiceno": 957,
    "receiptString": " INVOICE\n ----- \n December 15, 2016,...",
    "filename": "pos_12152016094415.txt"
  }
}
```

### Error Scenario:

#### 1. Empty Param:

```
{"data":{"status":"error","code":"C0002","message":"Customer parameter is empty."}}
```

```
{"data":{"status":"error","code":"ParamEmpty","message":"Payment parameter is empty."}}
```

```
{"data":{"status":"error","code":"ParamEmpty","message":"Orderline and Payment parameter is empty."}}
```

```
{"data":{"status":"error","code":"OrderlinesEmpty","message":"Stock list parameter is empty."}}
```

#### 2. Customer Exist Scenario:

```
{"data":{"status":"error","code":"C0002","message":"This customer is no longer exist."}}
```

#### 3. User Exist / Active Scenario:

```
{"data":{"status":"error","code":"C0002","message":"This user is no longer exist."}}
```

```
{"data":{"status":"error","code":"C0002","message":"This user is no longer active."}}
```

4. Order Exist Scenario:

```
{"data":{"status":"error","instatus":"success","code":"C0002","message":"Warning: Reference number associated with the order already exist."}}
```

Method / Action Name: **addCustomer**

Parameters:

```
{  
  "api": "live_mDhTrVewN5vMILwnc9qn60AOc",  
  "command": "addcustomer",  
  "customerName": "test002",  
  "customerEmail": "test002@gmail.com",  
  "customerPhone": "09234454321",  
  "customerStreet1": "Street 1",  
  "customerStreet2": "Street2",  
  "customerCity": "New York City",  
  "customerState": "State",  
  "customerZipcode": "12345",  
  "customerCountry": "USA",  
  "user": "restaurant"  
}
```

Description:

This method adds a customer into the system and gives back the customer name and code.

Return:

*A sample JSON return file is below:*

```
{  
  "status": "success",  
  "data": [{  
    "customerCode": "CODE",  
    "customerName": "test002"  
  }],  
  "message": "success"  
}
```

Method / Action Name: **updatecustomer**

Parameters:

```
{  
  "api": "live_mDhTrVewN5vMILwnc9qn60AOc",  
  "command": "updatecustomer",  
  "custCode": "XQTLYDCHSM",  
  "customerName": "test033",  
  "customerEmail": "test133@gmail.com",  
  "customerPhone": "09234454321",  
  "customerStreet1": "Street 1",  
  "customerStreet2": "Street2",  
  "customerCity": "New York City",  
  "customerState": "State",  
  "customerZipcode": "12345",  
  "customerCountry": "USA",  
  "username": "restaurant"  
}
```

Description:

This method updates a customer into the system and gives back the customer name and code.

Return:

*A sample JSON return file is below:*

```
{  
  "data": {  
    "status": "Success",  
    "message": "Updated Success!"  
  }  
}
```

## LIST STOCK CATEGORY LIST:

Param:

```
{
    "api": "live_mDhTrVewN5vMILwnc9qn60AOc",
    "command": "stockcategorylist",
    "username": "restaurant"
}
```

Description:

API to list out stock category in to the app.

Return:

*A sample JSON return file is below:*

```
{
    "data": {
        "status": "Success",
        "stockcategory": [
            {
                "categoryid": "BR",
                "categorydescription": "Burger",
                "stocktype": "F",
                "backcolor": "ea9336",
                "textcolor": "ffffff"
            },
            {
                "categoryid": "DD",
                "categorydescription": "Puddings",
                "stocktype": "F",
                "backcolor": "ff0000",
                "textcolor": "ffffff"
            }
        ]
    }
}
```

## ADD STOCK CATEGORY:

Param:

```
{
  "api": "live_mDhTrVewN5vMILwnc9qn60AOc",
  "command": "addstockcategoryfromWebsite",
  "username": "restaurant",
  "Param": [{
    "categorydesc": "test 4",
    "stocktype": "F",
    "backcolor": "eaeaea",
    "textcolor": "000000"
  }]
}
```

### Description:

API to add stock categories into the app.

Note:

1. Stock Type : F is static for now.

### Return:

*A sample JSON return file is below:*

```
{
  "data": {
    "status": "Success",
    "CategoryCode": "BEG",
    "message": "Insert Successfully!"
  }
}
```



## UPDATE STOCK CATEGORY:

### Param:

```
{
    "api": "live_mDhTrVewN5vMILwnc9qn60AOc",
    "command": "updatestockcategory",
    "username": "restaurant",
    "param": [{
        "categorycode": "test",
        "categorydesc": "test1",
        "stocktype": "F",
        "backcolor": "test1",
        "textcolor": "test1"
    }]
}
```

### Description:

API to update stock categories into the app.

### Return:

*A sample JSON return file is below:*

```
{
  "data": {
    "status": "Success",
    "message": "Insert Success!"
  }
}
```

## ADD PRODUCT:

### Param:

```
{
  "api": "live_mDhTrVewN5vMILwnc9qn60AOc",
  "command": "addstockitem",
  "username": "restaurant",
  "param": {
    "barcode": "6006031682771",
    "batchControl": 0,
    "current": 0,
    "itemCode": "121CP",
    "itemType": "B",
    "longDescription": "12 inch Cheese Pizza",
    "pricelist": "DE",
    "salesPrice": 18.99,
    "serialised": 0,
    "shortDescription": "12 inch Cheese Pizza",
    "stockCategory": "KD",
    "taxCategory": 1,
    "unitcost": 10.00,
    "units": "each",
    "salesFilter": []
  }
}
```

### Description:

API to add Product in to the app

### **Note:**

1. **"pricelist": "DE"** is a hardcode value now and it will be later changed via API
2. **"batchControl": 0**, is a hardcode value now and it will be later changed via API
3. **"Current": 0**, is a hardcode value now and it will be later changed via API
4. **"Serialised": 0**, is a hardcode value now and it will be later changed via API
5. **"taxCategory": 1** (1-Taxable supply, 2- Luxury Items, 4-Exempt, 5 - Freight)

### Return:

A sample JSON return file is below:

```
{
  "data": {
    "status": "Success",
    "message": "Insert Success!"
  }
}
```

## UPDATE PRODUCT:

### Param:

```
{
  "api": "live_mDhTrVewN5vMILwnc9qn60AOc",
  "command": "updatestockitem",
  "username": "restaurant",
  "param": {
    "barcode": "6006031682772",
    "batchControl": 0,
    "current": 0,
    "itemCode": "124CP",
    "itemType": "B",
    "longDescription": "124 inch Cheese Pizza",
    "pricelist": "DE",
    "salesPrice": 12,
    "serialised": 0,
    "shortDescription": "124 inch Cheese Pizza",
    "stockCategory": "KD",
    "taxCategory": 1,
    "unitcost": 12.00,
    "units": "each",
    "salesFilter": []
  }
}
```

### Description:

API to update Product in to the app

### **Note:**

6. "pricelist": "DE" is a hardcoded value now and it will be later changed via API
7. "batchControl": 0, is a hardcoded value now and it will be later changed via API
8. "Current": 0, is a hardcoded value now and it will be later changed via API
9. "Serialised": 0, is a hardcoded value now and it will be later changed via API
10. "taxCategory": 1 (1-Taxable supply, 2- Luxury Items, 4-Exempt, 5 - Freight)

### Return:

A sample JSON return file is below:

```
{
  "data": {
    "status": "Success",
    "message": "Updated Success!"
  }
}
```

## **CHECK IF STOCK CATEGORY EXISTS:**

### **Param:**

```
{
  "api": "live_mDhTrVewN5vMILwnc9qn60AOc",
  "command": "checkIfStockCategoryExist",
  "username": "restaurant",
  "param": {
    "category": "snacks"
  }
}
```

### **Description:**

API to check if a stock category exists or not in the app.

### **Return:**

*A sample JSON return file is below:*

### **Success:**

```
{
  "data": {
    "status": "Success",
    "stockcategory": [
      {
        "categoryid": "AP",
        "categorydescription": "Snacks",
        "stocktype": "F",
        "backcolor": "ffff00",
        "textcolor": "000000"
      }
    ],
    "message": "Stock category exists!"
  }
}
```

### **Error:**

```
{
  "data": {
    "status": "Success",
    "stockcategory": [],
    "message": "Stock category does not exist!"
  }
}
```

## Check If Customer Exist:

### Param:

```
{
  "api": "live_mDhTrVewN5vMILwnc9qn60AOc",
  "command": "checkIfCustomerExist",
  "username": "restaurant",
  "param": {
    "email": "guest@costbucket.com"
  }
}
```

### Description:

API to check if customer existing or not in the app

### Return:

*A sample JSON return file is below:*

```
{"data":{"status":"error","message":"Customer not exists"}}
{
  "data": {
    "status": "success",
    "debtor": [
      {
        "customerCode": "CASH",
        "customerName": "GUEST",
        "customerPhone": "0000000000",
        "customerEmail": "guest@costbucket.com"
      }
    ],
    "message": "Customer already exists"
  }
}
```

## Check If Product Exist:

### Param:

```
{
  "api": "live_mDhTrVewN5vMILwnc9qn60AOc",
  "command": "checkIfProductExist",
  "username": "restaurant",
  "param": {
    "stockid": "121CP"
  }
}
```

### Description:

API to check if product existing or not in the app

### Return:

*A sample JSON return file is below:*

```
{"data":{"status":"error","message":"Stock is not exists"}}
{
  "data": {
    "status": "success",
    "stocklist": [
      {
        "stockid": "121CP",
        "categoryid": "KD",
        "description": "12 inch Cheese Pizza",
        "standardcost": "10.00"
      }
    ],
    "message": "Stock already exists"
  }
}
```

## Check If Order Reference Number Exist:

### Param:

```
{
  "api": "live_mDhTrVewN5vMILwnc9qn60AOc",
  "command": "checkIfOrderReferenceExist",
  "username": "restaurant",
  "param": {
    "filename": "T120210123022931"
  }
}
```

### Description:

API to check if reference number existing or not with any order in the app

### Return:

*A sample JSON return file is below:*

```
{
  "data": {
    "status": "success",
    "message": "Reference number not associated with any order!"
  }
}

{
  "data": {
    "status": "error",
    "message": "Warning: Reference number associated with the order already exists."
  }
}
```

## **CHECK IF API KEY STATUS:**

### **Param:**

```
{
    "api": "live_mDhTrVewN5vMlLwnc9qn60AOc",
    "command": "checkApiKeyStatus",
    "username": "restaurant"
}
```

### **Description:**

API to check if an API Key exists or not in the app.

### **Return:**

*A sample JSON return file is below:*

#### **Success:**

```
{
    "data": {
        "status": "Success",
        "message": "API key exists!"
    }
}
```

#### **Error:**

```
{
    "status": "error",
    "error": "Authentication Error",
    "message": "The key is not correct. Please check it"
}
```