

Web Services API Doc 1.0

API URL: -

Sandbox: <https://dev.salesvu.com/townvu/api/index.php>

Production: <https://www.salesvu.com/townvu/api/index.php>

Overview:

All communications take place over secure HTTPS links on the Internet. Requests are sent to the servers by "POSTing" JSON data that has been formatted per the schema defined in this specification. The request should be sent as a value of a parameter called **request**. The response is returned as JSON data in the body of the HTTP reply.

API also supports JSONP. To pass function name use parameter called **callback**.

You can use cURL library to POST data on our server. Here is a cURL base code snippet in PHP. You can convert it to whatever language you are using at your end.

```
<?php
$ch = curl_init();
curl_setopt($ch, CURLOPT_URL, "https://dev.salesvu.com/townvu/api/index.php");
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
curl_setopt($ch, CURLOPT_POST, true);
curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
$data =
'request={"api_key":"YOUR_API_KEY","action":"get_order","store_id":"722","online_customer_id":"1","order_id":"1292"}';
curl_setopt($ch, CURLOPT_POSTFIELDS, $data);
$output = curl_exec($ch);
curl_close($ch);
echo $output;
?>
```

Right now there are following actions available in web service that can be consumed by Client:

- 1) change_approval_type
- 2) business_types
- 3) business
- 4) about



- 5) categories
- 6) product
- 7) modifiers
- 8) create_order
- 9) add_product
- 10) delete_order
- 11) delete_product
- 12) update_product_quantity
- 13) update_product_modifiers
- 14) get_order
- 15) complete_order
- 16) create_customer
- 17) assign_customer
- 18) get_card_info
- 19) process_payment
- 20) get_payment_url
- 21) add_order_note
- 22) init_payment
- 23) get_payment_devices
- 24) get_store_order
- 25) complete_store_order
- 26) get_mercury_account_info (Special permission needed for this API)
- 27) create_products
- 28) sub_categories

Actions related to Coupons

- 1) get_coupons

- 2) get_coupon_code
- 3) get_user_coupons
- 4) get_store_coupons
- 5) get_store_user_coupons
- 6) send_email

Brief description of above mentioned actions

- 1) **change_approval_type** – This action is to change the approval type for 3rd party apps. TownVu will send notification email with approve link in case of “manual” approval type. 3rd party app developer need to click on this link to approve business for their platform. In case of “automatic” TownVu will automatically approve business for 3rd party platform. Default approval type is “automatic”.

Parameters

- I. API Key/Id (**mandatory**)
- II. Action (**mandatory**)
- III. Approval Type (**mandatory**) (**MANUAL OR AUTOMATIC**)

Input JSON:

```
request = {"api_key" : "YOUR_API_KEY", "action" : "change_approval_type",  
"approval_type" : ["MANUAL" OR "AUTOMATIC"]}
```

Output JSON:

```
{"status": "ok", "msg": "Approval Type Changed."}
```

OR

```
{"status": "ok", "msg": "Invalid Approval Type."}
```

- 2) **business_types** – This will return categories/types of businesses available on TownVu.

Parameters



- I. API Key/Id (**mandatory**)
- II. Action (**mandatory**)

Input JSON:

```
request = {"api_key" : "YOUR_API_KEY", "action" : "business_types"}
```

Output JSON:

```
{ "status":"ok",  
  "data": [  
    {"id":"14","name":"Active Life"},  
    {"id":"17","name":"Appliances & Repair"}  
  ]  
}
```

3) business – This will return list of businesses belong to given business type.

Parameters

- I. API Key/Id (**mandatory**)
- II. Action (**mandatory**)
- III. Business Type (**mandatory**)

Input JSON:

```
request = {"api_key" : "YOUR_API_KEY", "action" : "business", "business_type" : "Business  
Type Id"}
```

Output JSON:

```
{ "status":"ok",  
  "data": [  
    {  
      "store_id":"647",  
      "name":"Test Store",  
      "online_ordering":0,  
      "location":{  
        "address":"New Delhi",  
        "country":"United States",  
        "state":"Alabama",  
        "city":"New Delhi",
```

```
        "zip": "110092",  
        "longitude": "77.2249600",  
        "latitude": "28.6353080"  
    }  
  ]  
}
```

- 4) **about** – This will return the information about the store id like currency, location, payment options, delivery mechanisms etc.

Parameters

- I. API Key/Id (**mandatory**)
- II. Action (**mandatory**)
- III. Store Id (**mandatory**) (Type: Numeric)

Input JSON:

```
request = {"api_key" : "YOUR_API_KEY", "action" : "about", "store_id" : "Store/Business Id"}
```

Output JSON:

```
{  
  "status": "ok",  
  "data": {  
    "currency_symbol": "$",  
    "currency_position": "L",  
    "delivery_mechanism": [  
      {  
        "dm_id": "1",  
        "name": "Delivery",  
        "fee": "0.00"  
      },  
      {  
        "dm_id": "2",  
        "name": "Mail",  
        "fee": "5.00"  
      },  
      {  
        "dm_id": "3",  
        "name": "Pickup",  
        "fee": "0.00"  
      }  
    ],  
    "payment_options": [  
      {  
        "type": "Credit Card",  
        "name": "Credit Card",  
        "fee": "0.00"  
      },  
      {  
        "type": "Net Banking",  
        "name": "Net Banking",  
        "fee": "0.00"  
      },  
      {  
        "type": "Cash",  
        "name": "Cash",  
        "fee": "0.00"  
      }  
    ]  
  }  
}
```

```
{
  "oopo_id": "2",
  "name": "Full payment in advance at the time of online order",
  "advance_amount": "0.00",
  "advance_type": "0",
  "refund_policy": "All products sold by ACME are brand new and are manufactured
to meet the highest industry standards. All prices and specifications are subject to change without
notice. Products are shipped F.O.B. (Freight On Board) and become the sole property of the
purchaser upon delivery to the specified shipping agent. For shipping damage the rahul ..... For
shipping damage the rahul All products sold by ACME are brand new and are manufactured to meet
the highest industry standards. All prices and specifications are subject to change without notice.
Products are shipped F.O.B. (Freight On Board) and become the sole property of the purchaser upon
delivery to the specified shipping agent. For shipping damage the rahul ..... For shipping damage the
rahul All products sold by ACME are brand new and are manufactured to meet the highest industry
standards. All prices and specifications are subject to change without notice. Products are shipped
F.O.B. (Freight On Board) and become the sole property of the purchaser upon delivery to the
specified shipping agent. For shipping damage the rahul ..... For shipping damage the rahul All
products sold by ACME are brand new and are manufactured to meet the highest industry standards.
All prices and specifications are subject to change without notice. Products are shipped F.O.B.
(Freight On Board) and become the sole property of the purchaser upon delivery to the specified
shipping agent. For shipping damage the rahul ..... For shipping damage the rahul All products sold by
ACME are brand new and are manufactured to meet the highest industry standards. All prices and
specifications are subject to change without notice. Products are shipped F.O.B. (Freight On Board)
and become the sole property of the purchaser upon delivery to the specified shipping agent. For
shipping damage the rahul ..... For shipping damage the rahul"
}
],
"published": "1",
"business_name": "Inventory Test 2",
"business_desc": "This is a Test Store.",
"opening_days": [
  "Sunday 8:30 AM to 8:45 PM",
  "Monday 8:00 AM to 8:00 PM",
  "Tuesday 8:00 AM to 8:00 PM",
  "Wednesday 8:00 AM to 8:00 PM",
  "Thursday 8:00 AM to 8:00 PM",
  "Friday 8:00 AM to 8:00 PM",
  "Saturday 8:00 AM to 8:00 PM"
],
"website": "",
"email": "anil@akchauhan.com",
"location": {
  "address": "Test Address",
  "city": "Austin",
  "state": "Texas",
  "country": "United States",
  "zip": "110092"
},
"phone": "981-074-9668",
"store_logo": "http://dev.salesvu.com/townvustore_logo/thumbs/[72x72]2006265808-
1328217011.jpg",
```

```
"facilities": {
  "num_of_seats": null,
  "average_price": null,
  "meals_served": [
    {
      "value": "Breakfast"
    },
    {
      "value": "Lunch"
    },
    {
      "value": "Dinner"
    }
  ]
},
"additional_info": {
  "take_out": null,
  "reservations": null,
  "outdoor_seating": null,
  "buffet": null,
  "wheelchair_accessible": null,
  "attire": null,
  "parking": null,
  "smoking": null,
  "alcohol": null,
  "home_delivery": null,
  "pets_allowed": null,
  "cuisine": [
    {
      "value": "American"
    },
    {
      "value": "Chinese"
    },
    {
      "value": "Indian"
    },
    {
      "value": "Vegan"
    }
  ]
},
"facebook_url": "",
"twitter_url": "",
"yelp_url": "",
"business_type": [
  {
    "tid": "13",
    "vid": "1",
    "name": "Beauty and Spas",
    "description": ""
  }
]
```

```
        "weight": "0"
      },
      {
        "tid": "20",
        "vid": "1",
        "name": "Couriers & Delivery Services",
        "description": "",
        "weight": "0"
      }
    ]
  }
}
```

5) categories – This will return all the product/services categories belong to given store ID.

Parameters

- I. API Key/Id (**mandatory**)
- II. Action (**mandatory**)
- III. Store Id (**mandatory**) (Type: Numeric)

Input JSON:

```
request = {
  "api_key": "YOUR_API_KEY",
  "action": "categories",
  "store_id": "Store/Business Id"
}
```

Output JSON:

```
{ "status": "ok",
  "data": [
    {
      "category_id": "4",
      "category_name": "Fast Food"
    },
    {
      "category_id": "6",
```



```
        "category_name":"Fruits"
      }
    ]
  }
```

6) product – This will return list of all products/services belong to given category ID.

Parameters

- I. API Key/Id (mandatory)
- II. Action (mandatory)
- III. Store Id (mandatory) (Type: Numeric)
- IV. Category Id (mandatory) (Type: Numeric)

Input JSON:

```
request = {
  "api_key": "YOUR_API_KEY",
  "action": "product",
  "store_id": "Store/Business Id",
  "category": "Category Id"
}
```

Output JSON:

```
{ "status":"ok",
  "data": [
    {
      "product_id":"2",
      "product_name":"choco",
      "description":"choco",
      "unit_id":"15",
      "selling_price":"50.00",
      "image":"https://dev.salesvu.com/product_images/582_2_1334672557.png",
      "total_modifiers":"0"
    }
  ]
}
```

7) **modifiers** – This will return list of all modifiers assigned to given product ID.

Parameters

- I. API Key/Id (**mandatory**)
- II. Action (**mandatory**)
- III. Store Id (**mandatory**) (Type: Numeric)
- IV. Product Id (**mandatory**) (Type: Numeric)

Input JSON:

```
request = {  
  
  "api_key": "YOUR_API_KEY",  
  
  "action": "modifiers",  
  
  "store_id": "Store/Business Id",  
  
  "product": "Product Id"  
  
}
```

Output JSON:

```
{  
  "status": "ok",  
  "data": [  
    {  
      "modifier_id": 1,  
      "modifier_name": "Cake Mod.",  
      "modifier_items": [{  
        "modifier_item_id": "1",  
        "product_id": "3",  
        "product_name": "Sugar",  
        "unit_id": "14",  
        "quantity": "100",  
        "extra_cost": "1.00"  
      }],  
    },  
    {  
      "modifier_id": 4,  
      "modifier_name": "cake mod3",  
      "modifier_items": [{  
        "modifier_item_id": "3",  
        "product_id": "3",  
        "product_name": "Sugar",  
        "unit_id": "14",  
      }],  
    }  
  ]  
}
```

```
        "quantity": "100",  
        "extra_cost": "0.00"  
    }  
  ]  
}
```

- 8) **create_order** – This will create your order and return you an order id, if you will pass products along with this request then this api will add product in this order as well and return you full order.

Parameters

- I. API Key/Id (**mandatory**)
- II. Action (**mandatory**)
- III. Store Id (**mandatory**) (Type: Numeric)
- IV. Online Customer Id (**mandatory**) (Type: Numeric)
 - a. Take "1" id for Guest customer.
- V. Order Details (**optional**)

Input JSON:

```
request = {  
  "api_key": "YOUR_API_KEY",  
  "action": "create_order",  
  "store_id": "Store/Business Id",  
  "online_customer_id": "1",  
  "order_details": [  
    {  
      "product_id": 12,  
      "selling_price": "10.00",  
      "unit_id": 15,  
      "quantity": 2,  
    }  
  ]  
}
```

```
"notes": [  
  {  
    "comment": "new test comment"  
  }  
]  
}  
]
```

Output JSON:

```
{  
  "status": "ok",  
  "data": {  
    "online_customer_id": 1,  
    "store_id": "623",  
    "order_id": 1288,  
    "dm_id": null,  
    "csa_id": null,  
    "note": "",  
    "partial_amount": "0.00",  
    "appointment_info": null,  
    "dm_product": 0,  
    "order_details": [  
      {  
        "order_detail_id": "1800",  
        "product_id": "10",  
        "unit_id": "15",  
        "quantity": "3",  
        "selling_price": "80.00",  
        "product_name": "Lunch",  
        "image": "https://dev.salesvu.com/product_images/623_10_1328216455.jpg",  
        "percentage": "10.00",  
        "notes": [  
          {  
            "customer_modifier_id": "380",  
            "comment": "Hello World"  
          }  
        ],  
        "od_selling_price": 80,  
        "extracost_total": 0,  
        "base_price": "80.00",  
        "base_price_total": 240,  
        "amount": 240,  
        "after_discount_amount": 240,  
      }  
    ]  
  }  
}
```

```
        "discount": 0,  
        "od_tax": 8,  
        "tax": 24,  
        "od_total_amount": 88,  
        "total_amount": 264  
    }  
],  
    "subtotal_amount": 240,  
    "total_tax": 24,  
    "total_discount": 0,  
    "total_amount": 264,  
    "total_items": 3  
}  
}
```

9) **add_product** – This will add your product in an order provide by you.

Parameters

- I. API Key/Id (**mandatory**)
- II. Action (**mandatory**)
- III. Store Id (**mandatory**) (Type: Numeric)
- IV. Order Id (**mandatory**) (Type: Numeric)
- V. Online Customer Id (**mandatory**) (Type: Numeric)
 - a. Take "1" id for Guest customer.
- VI. Order Details (**mandatory**)
 - a. Product Id
 - b. Selling Price
 - c. Unit Id
 - d. Quantity
 - e. Notes
 - i. Comment

Input JSON:

```
request = {  
    "api_key": "YOUR_API_KEY",
```



```
"action": "add_product",
"store_id": "Store/Business Id",
"online_customer_id": "1",
"order_id": "1288",
"order_details": [
  {
    "product_id": 12,
    "selling_price": "10.00",
    "unit_id": 15,
    "quantity": 2,
    "notes": [
      {
        "comment": "new test comment"
      }
    ]
  }
]
```

Output JSON:

```
{
  "status": "ok",
  "data": {
    "online_customer_id": 1,
    "store_id": "623",
    "order_id": "1288",
    "dm_id": null,
    "csa_id": null,
    "note": "",
    "partial_amount": "0.00",
    "appointment_info": null,
    "dm_product": 0,
    "order_details": [
      {
        "order_detail_id": "1800",
```

```
"product_id": "10",
"unit_id": "15",
"quantity": "3",
"selling_price": "80.00",
"product_name": "Lunch",
"image": "https://dev.salesvu.com/product_images/623_10_1328216455.jpg",
"percentage": "10.00",
"notes": [
  {
    "customer_modifier_id": "380",
    "comment": "Hello World"
  }
],
"od_selling_price": 80,
"extracost_total": 0,
"base_price": "80.00",
"base_price_total": 240,
"amount": 240,
"after_discount_amount": 240,
"discount": 0,
"od_tax": 8,
"tax": 24,
"od_total_amount": 88,
"total_amount": 264
},
{
  "order_detail_id": "1801",
  "product_id": "10",
  "unit_id": "15",
  "quantity": "3",
  "selling_price": "80.00",
  "product_name": "Lunch",
  "image": "https://dev.salesvu.com/product_images/623_10_1328216455.jpg",
  "percentage": "10.00",
  "notes": [
    {
      "customer_modifier_id": "381",
      "comment": "Hello World"
    }
  ],
  "od_selling_price": 80,
  "extracost_total": 0,
  "base_price": "80.00",
  "base_price_total": 240,
  "amount": 240,
  "after_discount_amount": 240,
  "discount": 0,
  "od_tax": 8,
  "tax": 24,
  "od_total_amount": 88,
  "total_amount": 264
}
```

```
    }  
  ],  
  "subtotal_amount": 480,  
  "total_tax": 48,  
  "total_discount": 0,  
  "total_amount": 528,  
  "total_items": 6  
}  
}
```

10) delete_order – This will delete your order.

Parameters

- I. API Key/Id (**mandatory**)
- II. Action (**mandatory**)
- III. Store Id (**mandatory**) (Type: Numeric)
- IV. Order Id (**mandatory**) (Type: Numeric)
- V. Online Customer Id (**mandatory**) (Type: Numeric)
 - a. Take "1" id for Guest customer.

Input JSON:

```
request = {  
  "api_key": "YOUR_API_KEY",  
  "action": "delete_order",  
  "store_id": "Store/Business Id",  
  "online_customer_id": "1",  
  "order_id": "1288"  
}
```

Output JSON:

```
{  
  "status": "ok",  
}
```



```
"msg": "Order Deleted"  
}
```

11) delete_product - This will delete your product from order and return your current order after deletion.

Parameters

- I. API Key/Id (**mandatory**)
- II. Action (**mandatory**)
- III. Store Id (**mandatory**) (Type: Numeric)
- IV. Order Id (**mandatory**) (Type: Numeric)
- V. Online Customer Id (**mandatory**) (Type: Numeric)
 - a. Take "1" id for Guest customer.
- VI. Order Details (**mandatory**)
 - a. Order Detail Id

Input JSON:

```
request = {  
  "api_key": "YOUR_API_KEY",  
  "action": "delete_product",  
  "store_id": "Store/Business Id",  
  "online_customer_id": "1",  
  "order_id": "1288",  
  "order_details": [  
    {  
      "order_detail_id": 12  
    }  
  ]  
}
```

```
}
```

Output JSON:

```
{  
  "status": "ok",  
  "data": {  
    "online_customer_id": 1,  
    "store_id": "623",  
    "order_id": "1288",  
    "dm_id": null,  
    "csa_id": null,  
    "note": "",  
    "partial_amount": "0.00",  
    "appointment_info": null,  
    "dm_product": 0,  
    "order_details": [  
      {  
        "order_detail_id": "1801",  
        "product_id": "10",  
        "unit_id": "15",  
        "quantity": "3",  
        "selling_price": "80.00",  
        "product_name": "Lunch",  
        "image":  
        "https://dev.salesvu.com/product_images/623_10_1328216455.jpg",  
        "percentage": "10.00",  
        "notes": [  
          {  
            "customer_modifier_id": "381",  
            "comment": "Hello World"  
          }  
        ]  
      }  
    ]  
  }  
}
```

```
    }  
  ],  
  "od_selling_price": 80,  
  "extracost_total": 0,  
  "base_price": "80.00",  
  "base_price_total": 240,  
  "amount": 240,  
  "after_discount_amount": 240,  
  "discount": 0,  
  "od_tax": 8,  
  "tax": 24,  
  "od_total_amount": 88,  
  "total_amount": 264  
}  
],  
"subtotal_amount": 240,  
"total_tax": 24,  
"total_discount": 0,  
"total_amount": 264,  
"total_items": 3  
}  
}
```

12) update_product_quantity - This will update your product quantity of product in order and return your current order after updation.

Parameters

- I. API Key/Id (**mandatory**)
- II. Action (**mandatory**)



- III. Store Id (**mandatory**) (Type: Numeric)
- IV. Order Id (**mandatory**) (Type: Numeric)
- V. Online Customer Id (**mandatory**) (Type: Numeric)
 - a. Take "1" id for Guest customer.
- VI. Order Details (**mandatory**)
 - a. Order Detail Id
 - b. Quantity

Input JSON:

```
request = {  
  "api_key": "YOUR_API_KEY",  
  "action": "update_product_quantity",  
  "store_id": "Store/Business Id",  
  "online_customer_id": "1",  
  "order_id": "1288",  
  "order_details": [  
    {  
      "order_detail_id": 1801,  
      "quantity": 2  
    }  
  ]  
}
```

Output JSON:

```
{  
  "status": "ok",  
  "data": {  
    "online_customer_id": 1,  
  }  
}
```

```
"store_id": "623",
"order_id": "1288",
"dm_id": null,
"csa_id": null,
"note": "",
"partial_amount": "0.00",
"appointment_info": null,
"dm_product": 0,
"order_details": [
  {
    "order_detail_id": "1801",
    "product_id": "10",
    "unit_id": "15",
    "quantity": "3",
    "selling_price": "80.00",
    "product_name": "Lunch",
    "image": "https://dev.salesvu.com/product_images/623_10_1328216455.jpg",
    "percentage": "10.00",
    "notes": [
      {
        "customer_modifier_id": "381",
        "comment": "Hello World"
      }
    ],
    "od_selling_price": 80,
    "extracost_total": 0,
    "base_price": "80.00",
```

```
        "base_price_total": 240,  
        "amount": 240,  
        "after_discount_amount": 240,  
        "discount": 0,  
        "od_tax": 8,  
        "tax": 24,  
        "od_total_amount": 88,  
        "total_amount": 264  
    }  
],  
    "subtotal_amount": 240,  
    "total_tax": 24,  
    "total_discount": 0,  
    "total_amount": 264,  
    "total_items": 3  
}  
}
```

13) update_product_modifiers - This will update your product quantity of product in order and return updated order.

Parameters

- I. API Key/Id (**mandatory**)
- II. Action (**mandatory**)
- III. Store Id (**mandatory**) (Type: Numeric)
- IV. Order Id (**mandatory**) (Type: Numeric)
- V. Online Customer Id (**mandatory**) (Type: Numeric)
 - a. Take "1" id for Guest customer.

VI. Order Details (mandatory)

- a. Order Detail Id
- b. Modifiers
 - i. Modifier Id
 1. Modifier Item Id
 2. Modifier Item Unit Id
 3. Quantity

Input JSON:

```
request = {  
  "api_key": "YOUR_API_KEY",  
  "action": "update_product_modifiers",  
  "store_id": "623",  
  "online_customer_id": 1,  
  "order_id": 1288,  
  "order_details": [  
    {  
      "order_detail_id": 1771,  
      "product_id": 10,  
      "modifiers": [  
        {  
          "modifier_id": 3,  
          "modifier_items": [  
            {  
              "modifier_item_id": 3,  
              "modifier_item_unit_id": 4,  
              "quantity": 100
```

```
    }
  ]
}
]
}
]
}
```

Output JSON:

```
{
  "status":"ok",
  "data":{
    "online_customer_id":1,
    "store_id":"623",
    "order_id":"1288",
    "dm_id":null,
    "csa_id":null,
    "note":"",
    "partial_amount":"467.50",
    "appointment_info":null,
    "dm_product":0,
    "order_details":[{
      "order_detail_id":"1771",
      "product_id":"10",
      "unit_id":"15",
      "quantity":"5",
      "selling_price":"80.00",
      "product_name":"Lunch",

      "image":"https://dev.salesvu.com/product_images/623_10_1328216455.jpg",
      "percentage":"10.00",
      "modifiers":[{
        "modifier_id":"3",
        "modifier_name":"Milk Group",
        "modifier_items":[{
          "modifier_item_id":"3",
          "modifier_item_unit_id":"4",
          "quantity":"100",
          "extra_cost":"5.00",
          "product_name":"Milk",
          "total_extracost":25
        }
      ]
    }],
    "od_selling_price":85,
    "extracost_total":25,
```



```
        "base_price": "80.00",
        "base_price_total": 400,
        "amount": 425,
        "after_discount_amount": 425,
        "discount": 0,
        "od_tax": 8.5,
        "tax": 42.5,
        "od_total_amount": 93.5,
        "total_amount": 467.5
    }},
    "subtotal_amount": 425,
    "total_tax": 42.5,
    "total_discount": 0,
    "total_amount": 467.5,
    "total_items": 5
}}
```

14) get_order - This will return your current order.

Parameters

- I. API Key/Id (**mandatory**)
- II. Action (**mandatory**)
- III. Store Id (**mandatory**) (Type: Numeric)
- IV. Order Id (**mandatory**) (Type: Numeric)
- V. Online Customer Id (**mandatory**) (Type: Numeric)
 - a. Take "1" id for Guest customer.

Input JSON:

```
request = {
    "api_key": "YOUR_API_KEY",
    "action": "get_order",
    "store_id": "Store/Business Id",
    "online_customer_id": "1",
    "order_id": "1288"
}
```

Output JSON:

```
{
  "status": "ok",
  "data": {
    "online_customer_id": 1,
    "store_id": "623",
    "order_id": "1288",
    "dm_id": null,
    "csa_id": null,
    "note": "",
    "partial_amount": "0.00",
    "appointment_info": null,
    "dm_product": 0,
    "order_details": [
      {
        "order_detail_id": "1792",
        "product_id": "10",
        "unit_id": "15",
        "quantity": "2",
        "selling_price": "80.00",
        "product_name": "Lunch",
        "image": "https://dev.salesvu.com/product_images/623_10_1328216455.jpg",
        "percentage": "10.00",
        "notes": [
          {
            "customer_modifier_id": "372",
            "comment": "Hello World"
          }
        ],
        "od_selling_price": 80,
        "extracost_total": 0,
        "base_price": "80.00",
        "base_price_total": 160,
        "amount": 160,
        "after_discount_amount": 160,
        "discount": 0,
        "od_tax": 8,
        "tax": 16,
        "od_total_amount": 88,
        "total_amount": 176
      }
    ],
    "subtotal_amount": 160,
    "total_tax": 16,
    "total_discount": 0,
    "total_amount": 176,
    "total_items": 2
  }
}
```

15) complete_order – This will place your order into the particular store and return you a store order id. But you have to call this service only in case of “Full Payment on Pickup/Delivery”, which was sent to you in “about” service otherwise

Parameters

- I. API Key/Id (**mandatory**)
- II. Action (**mandatory**)
- III. Store Id (**mandatory**) (Type: Numeric)
- IV. Order Id (**mandatory**) (Type: Numeric)
- V. Online Customer Id (**mandatory**) (Type: Numeric)
 - a. Take “1” id for Guest customer.

Input JSON:

```
request = {  
    "api_key": "YOUR_API_KEY",  
    "action": "complete_order",  
    "store_id": "Store/Business Id",  
    "online_customer_id": "1",  
    "order_id": "1288"  
}
```

Output JSON:

```
{  
    "status": "ok",  
    "data": {  
        "online_customer_id": "1",  
        "store_id": "623",  
        "order_id": "1288",  
        "store_order_id": "98",  
    }  
}
```

Errors



If there is any error then output JSON will have *status = error* and *error_msg* key for error detail.

Output JSON:

```
{"status":"error","error_msg":"Invalid Request."}
```

16) create_customer – This API help to create customer.

Parameters

- I. API Key/Id (**mandatory**)
- II. Action (**mandatory**)
- III. Store Id (**mandatory**) (Type: Numeric)
- IV. FaceBook User Id
- V. First Name (**mandatory**)
- VI. Last Name
- VII. Email (**mandatory**)

Input JSON:

```
request = {  
  "api_key": "YOUR_API_KEY",  
  "action": "create_customer",  
  "store_id": "Store/Business Id",  
  "fb_user_id": "1234",  
  "first_name": "abc",  
  "last_name": "test",  
  "email": "test@test.com",  
  "phone": "123456789"  
}
```

Output JSON:

```
{
```

```
"status": "ok",  
"data": {  
    "online_customer_id": "564",  
    "first_name": "Jitendra Kumar",  
    "last_name": "",  
    "email": "jktittalom@gmail.com",  
    "phone": "123456789"  
}  
}
```

17) assign_customer – This API used to assign any customer to the order other than guest.

Parameters

- I. API Key/Id (**mandatory**)
- II. Action (**mandatory**)
- III. Store Id (**mandatory**) (Type: Numeric)
- IV. Order Id (**mandatory**) (Type: Numeric)
- V. Online Customer Id (**mandatory**) (Type: Numeric)
- VI. Old Customer Id
- VII. New Customer Id

Input JSON:

```
request = {  
    "api_key": "YOUR_API_KEY",  
    "action": "assign_customer",  
    "store_id": "623",  
    "order_id": "1288",  
    "online_customer_id": "1",  
    "old_customer": 1,
```

```
"new_customer": 565  
}
```

Output JSON:

```
{  
  "status": "ok",  
  "data": {  
    "online_customer_id": 565,  
    "first_name": "Anil",  
    "last_name": "Chauhan",  
    "email": "meetanilchauhan@gmail.com",  
    "phone": "+91 9810749668"  
  }  
}
```

18) get_card_info – This API used to assign any customer to the order other than guest.

Parameters

- I. API Key/Id (**mandatory**)
- II. Action (**mandatory**)
- III. Store Id (**mandatory**) (Type: Numeric)
- IV. Order Id (**mandatory**) (Type: Numeric)
- V. Online Customer Id (**mandatory**) (Type: Numeric)

Input JSON:

```
request = {  
  "api_key": "YOUR_API_KEY",  
  "action": "get_card_info",  
  "store_id": "623",  
  "online_customer_id": 565,  
  "order_id": 1288  
}
```

Output JSON:

```
{
  "status": "ok",
  "data": {
    "online_customer_id": 565,
    "first_name": "Anil",
    "last_name": "Chauhan",
    "email": "meetanilchauhan@gmail.com",
    "phone": "+91 9810749668",
    "cards_info": [
      {
        "id": "33",
        "card_no": "xxxxxxx5373",
        "card_type": "DCVR"
      }
    ]
  }
}
```

19) process_payment – This API used to process payment for an order by existing credit card.

Parameters

- I. API Key/Id (**mandatory**)
- II. Action (**mandatory**)
- III. Store Id (**mandatory**) (Type: Numeric)
- IV. Order Id (**mandatory**) (Type: Numeric)
- V. Online Customer Id (**mandatory**) (Type: Numeric)
- VI. Card Id (**mandatory**) (Type: Numeric)

Input JSON:

```
request = {
  "api_key": "YOUR_API_KEY",
  "action": "process_payment",
  "store_id": "Store/Business Id",
  "online_customer_id": "1234",
  "order_id": "1288",
  "card_id": 2
}
```

```
}
```

Output JSON:

```
{  
  "status": "ok",  
  "data": {  
    "online_customer_id": "1",  
    "store_id": "623",  
    "order_id": "1288",  
    "store_order_id": "15",  
    "amount_paid": 250  
  }  
}
```

20) get_payment_url – This API will return payment URL. User browser needs to be redirected to this URL to enter credit card information.

Note: In case of successful payment API will call “complete_order”. So, there is no need to call this API after successful payment.

Parameters

- I. API Key/Id (**mandatory**)
- II. Action (**mandatory**)
- III. Store Id (**mandatory**) (Type: Numeric)
- IV. Order Id (**mandatory**) (Type: Numeric)
- V. Online Customer Id (**mandatory**) (Type: Numeric)
- VI. Return URL (**mandatory**)
- VII. Platform (**mandatory**)

Input JSON:

```
request = {  
  "api_key": "YOUR_API_KEY",  
  "action": "get_payment_url",  
  "store_id": "623",  
  "online_customer_id": 1,
```



```

"order_id": 1269,

"return_url": "https://dev.salesvu.com/townvu/api/dummy_url.php",

"platform": "mobile"

}

```

Output JSON:

```

{

"status": "ok",

"data": {

    "payment_url": "https://dev.salesvu.com/townvu/api/payment.php?
info=YTo2OntzOjQ6InNfaWQiO3M6MzoiNjIzOjQ6Im9faWQiO2k6MTI2OTtzOjU6InJfdXJsJt
zOjQ4OiJodHRwczovL2Rldi5zYWxl3Z1LmNvbS90b3dudnUvYXBpL2R1bW15X3Vybc5waHAiO
3M6Njoib19jX2lkIjtpOjE7czo0OiJwIjtzOjY6Im1vYmIsZSI7czo0OiJwX2lkIjtzOjM2OiI0MDRjNzUx
NC1kNmNiLTRIMjgtOTc2OS01N2Q4Yjc3NzQ0MzEiO30="

    }

}

```

Form Post Parameters Passed Back to return_url

Once the user enters the credit card data and presses Submit Payment, the payment is processed and control is returned back to the web site developer via the return_url. The fields passed to the developer are listed below.

Field Name	Description
status	Status of payment. ok – Successful payment error – Some error occurred
online_customer_id	Customer ID associated to order.
store_id	Store ID this order belongs to.
order_id	Order ID of order for which payment was made.
store_order_id	Order ID generated at store/business end.
amount_paid	Amount paid through Credit Card.
error_msg	Error message in case of status is “error”.

Credit Card Numbers for Testing in Sandbox Environment

Card Type	Test Card Number	Exp. Date	CVV	Remarks
MC	5499990123456781	0513	123	Amounts over \$10.99 will decline.
Visa	4003000123456781	0513	123	Amounts over \$10.99 will decline.
Amex	373953244361001	0513	1234	Supports dollar amounts over \$10.99
Discover	6011000997235373	0513	362	Supports dollar amounts over \$10.99

21) add_order_note – This will add note/comment in order.

Parameters

- I. API Key/Id (**mandatory**)
- II. Action (**mandatory**)
- III. Store Id (**mandatory**) (Type: Numeric)
- IV. Order Id (**mandatory**) (Type: Numeric)
- V. Online Customer Id (**mandatory**) (Type: Numeric)
 - a. Take "1" id for Guest customer.
- VI. Note (**mandatory**)

Input JSON:

```
request = {  
  "api_key": "YOUR_API_KEY",  
  "action": "add_order_note",  
  "store_id": "Store/Business Id",  
  "online_customer_id": "1",  
  "order_id": "1288",  
  "note": "Test Note"  
}
```

Output JSON:

```
{
  "status": "ok",
  "data": {
    "online_customer_id": 1,
    "store_id": "623",
    "order_id": "1288",
    "dm_id": null,
    "csa_id": null,
    "note": "",
    "partial_amount": "0.00",
    "appointment_info": null,
    "dm_product": 0,
    "order_details": [
      {
        "order_detail_id": "1800",
        "product_id": "10",
        "unit_id": "15",
        "quantity": "3",
        "selling_price": "80.00",
        "product_name": "Lunch",
        "image": "https://dev.salesvu.com/product_images/623_10_1328216455.jpg",
        "percentage": "10.00",
        "notes": [
          {
            "customer_modifier_id": "380",
            "comment": "Hello World"
          }
        ]
      },
      {
        "od_selling_price": 80,
        "extracost_total": 0,
        "base_price": "80.00",
        "base_price_total": 240,
        "amount": 240,
        "after_discount_amount": 240,
        "discount": 0,
        "od_tax": 8,
        "tax": 24,
        "od_total_amount": 88,
        "total_amount": 264
      }
    ],
    {
      "order_detail_id": "1801",
      "product_id": "10",
      "unit_id": "15",
      "quantity": "3",
      "selling_price": "80.00",
      "product_name": "Lunch",
      "image": "https://dev.salesvu.com/product_images/623_10_1328216455.jpg",
      "percentage": "10.00",
      "notes": [
        {
```

```
        "customer_modifier_id": "381",
        "comment": "Hello World"
    }
],
"od_selling_price": 80,
"extracost_total": 0,
"base_price": "80.00",
"base_price_total": 240,
"amount": 240,
"after_discount_amount": 240,
"discount": 0,
"od_tax": 8,
"tax": 24,
"od_total_amount": 88,
"total_amount": 264
}
],
"subtotal_amount": 480,
"total_tax": 48,
"total_discount": 0,
"total_amount": 528,
"total_items": 6,
"note": "Test Note"
}
}
```

22) **init_payment** – This API call require three steps:

- a) Call “init_payment” API by passing payment information like (amount, description, etc.)
- b) Collect customer payment information by sending them to “payment_url” returned by “init_payment”.
- c) Customer will be redirected to “return_url” page after payment. Show payment confirmation page to customer.

Note: -Same fields will be POSTed on “return_url” as listed for “get_payment_url” API.

Parameters

- I. API Key/Id (**mandatory**)
- II. Action (**mandatory**)
- III. Store Id (**mandatory**) (Type: Numeric)
- IV. Amount (**mandatory**) (Type: Numeric)



- V. Description (**mandatory**)
- VI. Return URL (**mandatory**)
- VII. Platform (**mandatory**)

Input JSON:

request =

```
{  
  
  "api_key": "YOUR_API_KEY",  
  
  "action": "init_payment",  
  
  "store_id": "623",  
  
  "amount": 9,  
  
  "desc": "Test Comment",  
  
  "return_url": "https://dev.salesvu.com/townvu/api/dummy_url.php",  
  
  "platform": "website"  
}
```

Output JSON:

```
{  
  "status": "ok",  
  "data": {  
    "payment_url": "https://dev.salesvu.com/townvu/api/payment.php?  
info=YTo2OntzOjQ6InNfaWQiO3M6MzoiNjZlJztOjQ6Im9faWQiO2k6MTI5OTtzOjU6InJfdXJsljt  
zOjQ4OiJodHRwczovL2Rldi5zYWxlci3Z1LmNvbS90b3dudnUvYXBpL2R1bW15X3VybC5waHAiO  
3M6NjoiY19jX2lkIjtpOjE7czoXOiJwIjtzOjY6Im9kaWY1LWJkMDktN2QyZjY3ZTJjOGRkljt9",  
    "order_id": "1299"  
  }  
}
```

23) get_payment_devices – This API call will return all active payment devices available in store/business.

Parameters

- I. API Key/Id (**mandatory**)
- II. Action (**mandatory**)

III. Store Id (**mandatory**) (Type: Numeric)

Input JSON:

request =

```
{
  "api_key": "YOUR_API_KEY",
  "action": "get_payment_devices",
  "store_id": "623"
}
```

Output JSON:

```
{
  "status": "ok",
  "data": {
    "payment_devices": [
      "My iPhone",
      "My iPad",
      "My iPad 01"
    ]
  }
}
```

24) get_store_order – This API call will return order detail from store/business.

Parameters

- I. API Key/Id (**mandatory**)
- II. Action (**mandatory**)
- III. Store Id (**mandatory**) (Type: Numeric)
- IV. Order ID (**mandatory**) (Type: Numeric)
- V. Payment Device (**mandatory**)

Input JSON:

request =

```
{
  "api_key": "YOUR_API_KEY",
  "action": "get_store_order",
  "store_id": "623",
  "order_id": "1",
}
```

```
"payment_device": "Demo"
}
```

Output JSON:

```
{
  "status": "ok",
  "data": {
    "store_id": "206",
    "order_id": "1",
    "note": null,
    "partial_amount": 10.07,
    "order_details": [
      {
        "order_detail_id": "3",
        "product_id": "1205",
        "unit_id": "15",
        "quantity": "1",
        "base_price": "2.00",
        "selling_price": "3.50",
        "discount_amount": "0.00",
        "product_name": "12 oz Iced Coffee",
        "image": "https://s3.amazonaws.com/prod_bucket/avatar/[72x72]907653772-1344928703.jpg",
        "percentage": "8.25",
        "modifiers": [
          {
            "modifier_id": "166",
            "modifier_name": "Iced Coffee Modifier",
            "modifier_items": [
              {
                "modifier_item_id": "1458",
                "modifier_item_unit_id": "0",
                "quantity": "1",
                "extra_cost": "0.00",
                "product_name": "TO GO",
                "total_extracost": 0
              },
              {
                "modifier_item_id": "1462",
                "modifier_item_unit_id": "7",
                "quantity": "1",
                "extra_cost": "1.50",
                "product_name": "4 oz Espresso",
                "total_extracost": 1.5
              }
            ]
          }
        ],
        "extracost_total": 1.5,
        "base_price_total": 3.5,

```

```
    "discount": 0
  },
  {
    "order_detail_id": "4",
    "product_id": "1204",
    "unit_id": "15",
    "quantity": "1",
    "base_price": "2.60",
    "selling_price": "4.10",
    "discount_amount": "0.00",
    "product_name": "16 oz Iced Coffee",
    "image": "https://s3.amazonaws.com/prod_bucket/avatar/[72x72]907653772-1344928703.jpg",
    "percentage": "8.25",
    "modifiers": [
      {
        "modifier_id": "166",
        "modifier_name": "Iced Coffee Modifier",
        "modifier_items": [
          {
            "modifier_item_id": "1462",
            "modifier_item_unit_id": "7",
            "quantity": "1",
            "extra_cost": "1.50",
            "product_name": "4 oz Espresso",
            "total_extracost": 1.5
          }
        ]
      }
    ],
    "extracost_total": 1.5,
    "base_price_total": 4.1,
    "discount": 0
  },
  {
    "order_detail_id": "5",
    "product_id": "1018",
    "unit_id": "0",
    "quantity": "1",
    "base_price": "1.70",
    "selling_price": "1.70",
    "discount_amount": "0.00",
    "product_name": "2%, Glass",
    "image": "https://s3.amazonaws.com/prod_bucket/avatar/[72x72]907653772-1344928703.jpg",
    "percentage": "8.25",
    "extracost_total": 0,
    "base_price_total": 1.7,
    "discount": 0
  }
],
```



```
"subtotal_amount": "9.30",  
"total_tax": "0.77",  
"total_discount": "0.00",  
"total_amount": "10.07"  
}  
}
```

25) complete_store_order – This API call will complete store/business order. You also need to pass payment information returned by Mercury payment systems in this API call. You should also use Invoice No as Order ID when you initialize payment request in Mercury payment system.

Parameters

- I. API Key/Id (**mandatory**)
- II. Action (**mandatory**)
- III. Store Id (**mandatory**) (Type: Numeric)
- IV. Order ID (**mandatory**) (Type: Numeric)
- V. Payment Device (**mandatory**)
- VI. Amount (**mandatory**)
- VII. Process Data (**mandatory**)
- VIII. AcqRefData (**mandatory**)
- IX. Token (**mandatory**)
- X. Auth Code (**mandatory**)
- XI. Status (**mandatory**)
- XII. Tran Type (**mandatory**)
- XIII. Card Type (**mandatory**)
- XIV. Masked Amount (**mandatory**)
- XV. Ref No (**mandatory**)

Input JSON:

request =

```
{
```

```
"api_key": "YOUR_API_KEY",
"action": "complete_store_order",
"store_id": "623",
"order_id": "1",
"payment_device": "Demo",
"amount": 10.07,
"process_data": "|00|210140620000",
"acq_ref_data": "aEb282072021160725cLXFXd5e00IG2",
"token": "RC56T9wH1HAJ+LzcYW8+b65MUIIRd7GprYeiPM8zPngTERIQAREQBlje",
"auth_code": "511526",
"status": "Approved",
"tran_type": "Credit",
"card_type": "VISA",
"masked_account": "491986XXXXXX6807",
"ref_no": "6"
}
```

Output JSON:

```
{
  "status": "ok",
  "data": {
    "msg": "Information updated successfully"
  }
}
```

26) get_mercury_account_info – This API call will return merchant’s mercury account ID and web service password. You need to ask for special permission to call this API.

Parameters

- I. API Key/Id (**mandatory**)
- II. Action (**mandatory**)
- III. Store Id (**mandatory**) (Type: Numeric)

Input JSON:

request =

```
{
  "api_key": "YOUR_API_KEY",
  "action": "get_mercury_account_info",
  "store_id": "623"
}
```

Output JSON:

```
{
  "status": "ok",
  "data": {
    "merchant_id": "395347308=E2ETKN",
    "webservice_password": "123E2ETKN"
  }
}
```

27) create_products – This API call will create products (with categories, modifiers, and sale tax brackets) with in merchant’s store. Each item (product, category, modifier group, modifier group item, and sales tax slab) should have unique mapping ID. You will get “success” and “errors” keys in response. The “success” will contains all products created successfully in our system and “errors” keys contain products having any issues.

Parameters

- I. API Key/Id (**mandatory**)
- II. Action (**mandatory**)
- III. Store Id (**mandatory**) (Type: Numeric)
- IV. Products (**mandatory**) (Type: JSON)

Input JSON:

request =

```
{
  "api_key": "YOUR_API_KEY",
  "action": "create_products",
  "store_id": "623",
  "products": [
    {
      "product_mapping_id": "AB-040",
      "product_id": "",
      "product_name": "TO GO",
      "status": "1",
      "description": "",
      "category_mapping_id": "AB-041",
      "category": "Paper Goods",
      "product_type": "Component"
    },
    {
      "product_mapping_id": "AB-042",
```

```
"product_id": "",
"product_name": "Egg White",
"status": "1",
"description": "",
"category_mapping_id": "AB-043",
"category": "Meat",
"product_type": "Product",
"price": 2,
"tax_bracket_mapping_id": "AB-032",
"tax_bracket": "Texas Tax Rate (8.25%)",
"tax_percentage": 8.25,
"color": "Purple"
},
{
  "product_mapping_id": "AB-001",
  "product_id": "",
  "product_name": "Italian Eggs",
  "status": "1",
  "description": "",
  "category_mapping_id": "AB-002",
  "category": "Breakfast Food",
  "product_type": "Product",
  "price": 4,
  "tax_bracket_mapping_id": "AB-032",
  "tax_bracket": "Texas Tax Rate (8.25%)",
  "tax_percentage": 8.25,
  "color": "Purple",
  "upc_code": "UPC1010101",
  "sku": "SKU837465",
  "modifiers": [
    {
      "modifier_mapping_id": "AB-003",
      "modifier_name": "Breakfast Meat",
      "modifier_items": [
        {
          "modifier_item_mapping_id": "AB-004",
          "product_name": "Egg White",
          "quantity": 1,
          "extra_cost": 2,
          "position": 1
        },
        {
          "modifier_item_mapping_id": "AB-006",
          "product_name": "Honey Ham",
          "quantity": 1,
```

```
        "extra_cost": 2.5,
        "position": 2
      },
      {
        "modifier_item_mapping_id": "AB-007",
        "product_name": "Mozzarella Cheese",
        "quantity": 1,
        "extra_cost": 2.5,
        "position": 3
      },
      {
        "modifier_item_mapping_id": "AB-008",
        "product_name": "Prosciutto",
        "quantity": 1,
        "extra_cost": 2.5,
        "position": 4
      }
    ],
    "position": 1
  },
  {
    "modifier_mapping_id": "AB-015",
    "modifier_name": "To Go",
    "modifier_items": [
      {
        "modifier_item_mapping_id": "AB-016",
        "product_name": "To Go",
        "quantity": 1,
        "extra_cost": 0,
        "position": 1
      }
    ],
    "position": 2
  }
]
}
```

Output JSON:

```
{
  "status": "ok",
  "data": {
    "success": {
      "AB-040": {
```

```
"product_mapping_id": "AB-040",
"product_id": 69,
"product_name": "TO GO",
"status": "1",
"description": "",
"category_mapping_id": "AB-041",
"category": "Paper Goods",
"product_type": "Component",
"category_id": "29"
},
"AB-042": {
  "product_mapping_id": "AB-042",
  "product_id": 70,
  "product_name": "Egg White",
  "status": "1",
  "description": "",
  "category_mapping_id": "AB-043",
  "category": "Meat",
  "product_type": "Product",
  "price": 2,
  "tax_bracket_mapping_id": "AB-032",
  "tax_bracket": "Texas Tax Rate (8.25%)",
  "tax_percentage": 8.25,
  "color": "Purple",
  "category_id": "30",
  "tax_bracket_id": "6"
},
"AB-001": {
  "product_mapping_id": "AB-001",
  "product_id": 71,
  "product_name": "Italian Eggs",
  "status": "1",
  "description": "",
  "category_mapping_id": "AB-002",
  "category": "Breakfast Food",
  "product_type": "Product",
  "price": 4,
  "tax_bracket_mapping_id": "AB-032",
  "tax_bracket": "Texas Tax Rate (8.25%)",
  "tax_percentage": 8.25,
  "color": "Purple",
  "upc_code": "UPC0986534",
  "sku": "SKU837465",
  "modifiers": [
    {
      "modifier_mapping_id": "AB-003",
      "modifier_name": "Breakfast Meat",
      "modifier_items": [
        {
          "modifier_item_mapping_id": "AB-004",
          "product_name": "Egg White",
```

```
    "quantity": 1,
    "extra_cost": 2,
    "position": 1,
    "modifier_item_id": 45
  },
  {
    "modifier_item_mapping_id": "AB-006",
    "product_name": "Honey Ham",
    "quantity": 1,
    "extra_cost": 2.5,
    "position": 2,
    "modifier_item_id": 46
  },
  {
    "modifier_item_mapping_id": "AB-007",
    "product_name": "Mozzarella Cheese",
    "quantity": 1,
    "extra_cost": 2.5,
    "position": 3,
    "modifier_item_id": 47
  },
  {
    "modifier_item_mapping_id": "AB-008",
    "product_name": "Prosciutto",
    "quantity": 1,
    "extra_cost": 2.5,
    "position": 4,
    "modifier_item_id": 48
  }
],
"position": 1,
"modifier_id": 26
},
{
  "modifier_mapping_id": "AB-015",
  "modifier_name": "To Go",
  "modifier_items": [
    {
      "modifier_item_mapping_id": "AB-016",
      "product_name": "To Go",
      "quantity": 1,
      "extra_cost": 0,
      "position": 1,
      "modifier_item_id": 49
    }
  ]
},
"position": 2,
"modifier_id": 27
}
],
"category_id": "35",
```

```
        "tax_bracket_id": "6"
      }
    },
    "errors": []
  }
}
```

28) sub_categories – This will return all the sub categories belong to given category ID.

Parameters

- I. API Key/Id (**mandatory**)
- II. Action (**mandatory**)
- III. Store Id (**mandatory**) (Type: Numeric)
- IV. Category Id (**mandatory**) (Type: Numeric)

Input JSON:

```
request = {
  "api_key": "YOUR_API_KEY",
  "action": "sub_categories",
  "store_id": "Store/Business Id",
  "category_id": "Category Id"
}
```

Output JSON:

```
{ "status": "ok",
  "data": [
    {
      "category_id": "4",
      "category_name": "Fast Food"
    },
    {
      "category_id": "6",
      "category_name": "Fruits"
    }
  ]
}
```


Brief description of actions related to Coupons

1) **get_coupons** – This will give you list of coupons as per passed parameters.

Parameters

- I. API Key/Id (**mandatory**)
- II. Action (**mandatory**)
- III. Facebook User Id (**mandatory**) (Type: Numeric)
- IV. Start From (**mandatory**) (Type: Numeric)
- V. No. Of Records (**mandatory**) (Type: Numeric)
- VI. Latitude (**mandatory**) (Type: Numeric)
- VII. Longitude (**mandatory**) (Type: Numeric)
- VIII. Radius (in Miles) (**mandatory**) (Type: Numeric)

Input JSON:

```
request = {  
  "api_key": "YOUR_API_KEY",  
  "action": "get_coupons",  
  "facebook_user_id": "723041413",  
  "start_from": 0,  
  "no_of_record": 2,  
  "latitude": 97.7428,  
  "longitude": 30.2699,  
  "radius": 100  
}
```

Output JSON:

```
{
  "status": "ok",
  "data": {
    "response": [
      {
        "leftDealsText": "5 left",
        "store_id": "6863",
        "store_name": "Dolce Baking Company",
        "store_address": "6974 Mission Road, Prairie Village, Kansas, United States",
        "store_address_without_country": "6974 Mission Road, Prairie Village, Kansas",
        "coupon_id": "11",
        "townvustore_id": "1268",
        "townvustore_avatar":
        "https://s3.amazonaws.com/prod_bucket/avatar/[72x72]1820247281-1345642519.jpg",
        "coupon_name": "Free slice of pumpkin bread pudding",
        "saving": 100,
        "distance": "3239.04 Miles",
        "email": "crystal@dolcebakingco.com",
        "phone": null,
        "coupon_image":
        "https://s3.amazonaws.com/prod_bucket/avatar/[72x72]1820247281-1345642519.jpg"
      },
      {
        "leftDealsText": "4 left",
        "store_id": "11517",
        "store_name": "Le creazioni di Melina",
        "store_address": "Via Monviso1/D, Arese (MI), Italia, Italia",
        "store_address_without_country": "Via Monviso1/D, Arese (MI), Italia",
        "coupon_id": "1",
        "townvustore_id": "1311",
        "townvustore_avatar":
        "https://s3.amazonaws.com/prod_bucket/avatar/[72x72]2066174671-1351436648.jpg",
        "coupon_name": "10% off Noci",
        "saving": 10,
        "distance": "3573.34 Miles",
        "email": "buonincarm@me.com",
        "phone": null,
        "coupon_image":
        "https://s3.amazonaws.com/prod_bucket/avatar/[72x72]2066174671-1351436648.jpg"
      }
    ],
    "total_record": "14"
  }
}
```

2) **get_coupon_code** – This will return coupon code for selected coupon.

Parameters

I. API Key/Id (mandatory)

- II. Action (**mandatory**)
- III. Facebook User Id (**mandatory**) (Type: Numeric)
- IV. Facebook User First Name (**mandatory**)
- V. Facebook User Last Name (**mandatory**)
- VI. Store Id (**mandatory**) (Type: Numeric)
- VII. Coupon Id (**mandatory**) (Type: Numeric)

Input JSON:

```
request = {  
  "api_key": "YOUR_API_KEY",  
  "action": "get_coupon_code",  
  "facebook_user_id": "723041413",  
  "facebook_user_firstname": "Anil",  
  "facebook_user_lastname": "Kumar",  
  "store_id": 11517,  
  "coupon_id": 1  
}
```

Output JSON:

```
{  
  "status": "ok",  
  "data": {  
    "response": {  
      "coupon_code_id": 2,  
      "coupon_code": "370543",  
      "coupon_code_expiry_date": "2012-11-04",  
      "coupon_code_create_date": "2012-11-01"  
    }  
  }  
}
```

3) **get_user_coupons** – This will give you list of coupons used by passed Facebook user ID.

Parameters

- I. API Key/Id (**mandatory**)

- II. Action (**mandatory**)
- III. Facebook User Id (**mandatory**) (Type: Numeric)
- IV. Start From (**mandatory**) (Type: Numeric)
- V. No. Of Records (**mandatory**) (Type: Numeric)
- VI. Latitude (**mandatory**) (Type: Numeric)
- VII. Longitude (**mandatory**) (Type: Numeric)
- VIII. Radius (in Miles) (**mandatory**) (Type: Numeric)

Input JSON:

```
request = {  
  "api_key": "YOUR_API_KEY",  
  "action": "get_user_coupons",  
  "facebook_user_id": "723041413",  
  "start_from": 0,  
  "no_of_record": 2,  
  "longitude": 97.7428,  
  "latitude": 30.2699,  
  "radius": 100  
}
```

Output JSON:

```
{  
  "status": "ok",  
  "data": {  
    "response": [  
      {  
        "store_id": "11517",  
        "store_name": "Le creazioni di Melina",  
        "store_address": "Via Monviso1/D, Arese (MI), Italia, Italia",  
        "store_address_without_country": "Via Monviso1/D, Arese (MI), Italia",  
        "coupon_id": "1",  
        "townvustore_avatar":  
        "https://s3.amazonaws.com/prod_bucket/avatar/[72x72]2066174671-1351436648.jpg",  
        "coupon_name": "10% offNoci",  
        "saving": 10,  
        "distance": "4710.2 Miles",  
      }  
    ]  
  }  
}
```

```
        "email": "buonincarm@me.com",
        "phone": null,
        "coupon_image":
"https://s3.amazonaws.com/prod_bucket/avatar/[72x72]2066174671-1351436648.jpg",
        "coupon_code_id": "2",
        "coupon_code": "370543",
        "use_status": "1",
        "today": "2012-11-01 14:23:36",
        "townvustore_id": "1311",
        "coupon_code_expiry_date": "2012-11-04"
    }
],
"total_record": 1
}
}
```

4) get_store_coupons – This will give you list of coupons for passed store ID.

Parameters

- I. API Key/Id (mandatory)**
- II. Action (mandatory)**
- III. Facebook User Id (mandatory)** (Type: Numeric)
- IV. Start From (mandatory)** (Type: Numeric)
- V. No. Of Records (mandatory)** (Type: Numeric)
- VI. Latitude (mandatory)** (Type: Numeric)
- VII. Longitude (mandatory)** (Type: Numeric)
- VIII. Store Id (mandatory)** (Type: Numeric)

Input JSON:

```
request = {
    "api_key": "YOUR_API_KEY",
    "action": "get_store_coupons",
    "facebook_user_id": "723041413",
    "start_from": 0,
    "no_of_record": 2,
    "longitude": 97.7428,
    "latitude": 30.2699,
```

```
"store_id": 6863  
}
```

Output JSON:

```
{  
  "status": "ok",  
  "data": {  
    "response": [  
      {  
        "leftDealsText": "5 left",  
        "store_id": "6863",  
        "store_name": "Dolce Baking Company",  
        "store_address": "6974 Mission Road, Prairie Village, Kansas, United States",  
        "store_address_without_country": "6974 Mission Road, Prairie Village, Kansas",  
        "coupon_id": "11",  
        "townvustore_id": "1268",  
        "townvustore_avatar":  
        "https://s3.amazonaws.com/prod_bucket/avatar/[72x72]1820247281-1345642519.jpg",  
        "coupon_name": "Free slice of pumpkin bread pudding",  
        "saving": 100,  
        "distance": "7584.44 Miles",  
        "email": "crystal@dolcebakingco.com",  
        "phone": null,  
        "coupon_image":  
        "https://s3.amazonaws.com/prod_bucket/avatar/[72x72]1820247281-1345642519.jpg"  
      }  
    ],  
    "total_record": "1"  
  }  
}
```

5) **get_store_user_coupons** – This will give you list of coupons used by passed Facebook user ID in given store.

Parameters

- I. API Key/Id (**mandatory**)
- II. Action (**mandatory**)
- III. Facebook User Id (**mandatory**) (Type: Numeric)
- IV. Start From (**mandatory**) (Type: Numeric)
- V. No. Of Records (**mandatory**) (Type: Numeric)
- VI. Latitude (**mandatory**) (Type: Numeric)
- VII. Longitude (**mandatory**) (Type: Numeric)

VIII. Radius (in Miles) (**mandatory**) (Type: Numeric)

IX. Store Id (**mandatory**) (Type: Numeric)

Input JSON:

```
request = {  
  
  "api_key": "YOUR_API_KEY",  
  
  "action": "get_store_user_coupons",  
  
  "facebook_user_id": "723041413",  
  
  "start_from": 0,  
  
  "no_of_record": 2,  
  
  "longitude": 97.7428,  
  
  "latitude": 30.2699,  
  
  "radius": 100,  
  
  "store_id": 1311  
  
}
```

Output JSON:

```
{  
  "status": "ok",  
  "data": {  
    "response": [  
      {  
        "store_id": "11517",  
        "store_name": "Le creazioni di Melina",  
        "store_address": "Via Monviso1/D, Arese (MI), Italia, Italia",  
        "store_address_without_country": "Via Monviso1/D, Arese (MI), Italia",  
        "coupon_id": "1",  
        "townvustore_avatar":  
"https://s3.amazonaws.com/prod_bucket/avatar/[72x72]2066174671-1351436648.jpg",  
        "coupon_name": "10% offNoci",  
        "saving": 10,  
        "distance": "4710.2 Miles",  
        "email": "buonincarm@me.com",  
        "phone": null,  
        "coupon_image":  
"https://s3.amazonaws.com/prod_bucket/avatar/[72x72]2066174671-1351436648.jpg",  
        "coupon_code_id": "2",  
        "coupon_code": "370543",  
        "use_status": "1",  
        "today": "2012-11-01 14:49:51",  
      }  
    ]  
  }  
}
```

```
        "townvustore_id": "1311",
        "coupon_code_expiry_date": "2012-11-04"
    }
],
"total_record": 1
}
}
```

6) send_email – This will send coupon detail to given email address.

Parameters

- I. API Key/Id (mandatory)**
- II. Action (mandatory)**
- III. Facebook User Id (mandatory)** (Type: Numeric)
- IV. Email Address (mandatory)**
- V. Store Id (mandatory)** (Type: Numeric)
- VI. Coupon Code Id (mandatory)** (Type: Numeric)

Input JSON:

```
request = {
    "api_key": "YOUR_API_KEY",
    "action": "send_email",
    "facebook_user_id": "723041413",
    "email_address": "anil.chauhan@salesvu.com",
    "store_id": 11517,
    "coupon_code_id": 2
}
```

Output JSON:

```
{
  "status": "ok",
  "data": {
    "response": "Sent"
  }
}
```