

StoreHub REST APIs

General	2
API Specifications	2
Authentication	2
Rate Limiting	2
APIs	2
Products	2
Get Product List	2
Get Product by Id	6
Customers	6
Create Customer	6
Get Customer by RefId	9
Update Customer	9
Get All Customers	10
Search Customers	10
Inventory	11
Get Inventory	11
Transaction	12
Get Transactions	12
Add Transaction	15
Cancel Transaction	17
Employees	18
Get All Employees	18
Stores	19
Get All Stores	19
Timesheets	20
Search Timesheets	20
Revision History	21

General

API Specifications

Protocol	HTTPS
Host	api.storehubhq.com
Accepted Request Format	JSON
Response Format	JSON
Date Format	Any request parameters or response field of date are in ISO 8601 , e.g. 1970-01-01T00:00:00+00:00 1970-01-01T00:00:00Z

Authentication

All APIs are authenticated via basic HTTP authentication. The Username is the store name you have registered with StoreHub (ie. subdomain of the back-office URL) and password is the API token. The API token is auto-generated for each registered account. Please contact your StoreHub sales representative to receive your account's API token.

Rate Limiting

Rate limiting may be applied if your application is detected to have exceeded over 3 calls per second. Once rate limiting is applied, your application will receive an error in response. If your account has been rate limited, you will need to contact the StoreHub customer success team to remove the rate limiting. Please note that we reserve the right to permanently block accounts who repeatedly abuse the StoreHub API.

APIs

Products

1. Get Product List

URL	/products
HTTP Verb	GET

Parameters

This API doesn't accept any additional query parameters

Response

Returns an array of Product objects. If there's no products, empty array is returned.

The Transfer-Encoding is **chunked**

```
{
  id: "587890b412e90163297e8e7d",
  name: "Product 1",
  tags: [
    "test",
    "only"
  ],
  category: "Product",
  sku: "P001",
  priceType: "Fixed",
  unitPrice: 94.33962264150944,
  cost: 53.33,
  barcode: "19238912312,1231242314",
  trackStockLevel: true,
  isParentProduct: false
},
{
  id: "587890bf12e90163297e8e94",
  name: "Product 2",
  tags: [
    "item",
    "test"
  ],
  category: "Item",
  sku: "P002",
  priceType: "Fixed",
  unitPrice: 94.33962264150944,
  cost: 20,
  barcode: "204833921023881",
  trackStockLevel: true,
  isParentProduct: false,
}
```

Product Schema

id	String	This is a unique id that identifies the product
name	String	Product's name
sku	String	Product's SKU
barcode	String	Product's barcode number

category	String	Product's category
subCategory	String	Product's subCategory
tags	Array of String	Tags of the product
priceType	Enum	<p>Possible values are 'Fixed' or 'Variable'.</p> <p>If the product's priceType = 'fixed', its unitPrice is determined by the unitPrice set on the product.</p> <p>If the product's priceType = 'variable', its unitPrice is specified during check-out. And unitPrice set on the product is ignored.</p>
unitPrice	Number	Unit price of the product
cost	Number	The amount of money or cost required to manufacture the product
trackStockLevel	Boolean	If the product stock level is tracked by StoreHub, this value is set to true
isParentProduct	Boolean	<p>A product is considered as a parent product if and only if it has variants and trackStockLevel equals to true.</p> <p>Please refer here for more information about Parent Products</p>
variantGroups	Array of VariantGroup	<p>Variant groups of a product, this value is only returned on parent products.</p> <p>Please refer here for more information</p>
variantValues	Array of VariantValue	<p>Variants value of a child product, this value is only returned for child products</p> <p>Please refer here for more information</p>
parentProductId	String	<p>Parent product id, this value is only returned on child products.</p> <p>Please refer here for more information</p>

Parent and Child Products

If a product has variants and tracking stock level is enabled, it's considered to be a parent product.

An example of a Parent Product is a product named 't-shirt' that is configured to track stock level and has two variants groups:

- Colour. Options are red, green, yellow
- Size. Options are S,M,L,

A t-shirt that is red and small sized is a child product whose variantValues are colour = red and size = S. The parent product t-shirt with two variant groups of 3 options, will have 3 child products. ($3 \times 3 = 9$)

If a product has variants but tracking stock value is not enabled, it's not considered a parent product and doesn't have child products.

VariantGroup Schema

id	String	id of the variant group
name	String	Variant group name, e.g. color, size
options	Array of VariantOption	Available options of this variant group

VariantOption Schema

optionValue	String	Value of the option, e.g 'red'
isDefault	Boolean	If this option is the default value of the variant, the value is set to true.
priceDifference	Number	<p>The unitPrice set on the product is the base for the price difference. And the unitPrice of a variant is determined by product unitPrice and priceDifference set on the variant group.</p> <p>For example, a coffee's unitPrice is \$1 and has two variant groups, one is 'temperature' with options hot and cold, one is 'size' with normal and large. If the priceDifference of cold option is \$1 and priceDifference of large option is \$2, then the unit price of a cold large coffee is \$1 + \$1 + \$2 = \$4.</p>

VariantValue Schema

variantGroupId	String	id of the variantGroup
----------------	--------	------------------------

value	String	Value of the corresponding variant
-------	--------	------------------------------------

2. Get Product by Id

URL	/products/<id>
HTTP Verb	GET

Response

Returns a product if a product with specified id can be found. Otherwise, returns 404 as status code.

Please refer to Product schema [here](#)

```
{
  id: "587890b412e90163297e8e7d",
  name: "Product 1",
  tags: [
    "test",
    "only"
  ],
  category: "Product",
  sku: "P001",
  priceType: "Fixed",
  unitPrice: 94.33962264150944,
  cost: 53.33,
  barcode: "19238912312,1231242314",
  trackStockLevel: true,
  isParentProduct: false
}
```

Customers

1. Create Customer

URL	/customers
HTTP Verb	POST
Content Type	application/json

Request Body

refId	UUID	Required	uniquely identify a customer, need to be passed in by client
firstName	String	Required	customer's first name
lastName	String	Required	customer's last name
email	String	Optional	customer's email
phone	String	Optional	customer's phone number
address1	String	Optional	street line 1 of customer's address
address2	String	Optional	street line 2 of customer's address
city	String	Optional	city of customer's address
state	String	Optional	state of customer's address
postalCode	String	Optional	postal code of customer's address
birthday	String	Optional	customer's birthday, format should be YYYY-MM-DD
memberId	String	Optional	customer's member Id
tags	Array of String	Optional	tags of the customer. Multiple tags can be added separated by a comma.
createdTime	String	Optional	timestamp when the customer is added. if createdTime is missing in request body, current time is used when the request is processed

Response

If an existing customer with the same refId is found, the request is ignored and the existing customer object is returned.

There's no validation on uniqueness of name/email/phone. As long as the refId is unique, a customer with the same name/email/phone can be added multiple times. If no errors are found in the POST, a new customer is created and the created customer object is returned.

```
{
  refId: "815c8c86-1b1d-41b7-b979-7e9d03f3f9fe",
  firstName: "Tom",
  lastName: "Anderson",
  email: "tom@stone.com",
  phone: "0123456789",
```

```

address1: "52, Jalan USJ 2/5J",
address2: "USJ 2",
city: "Subang Jaya",
state: "Selangor",
postalCode: "47600",
createdTime: "2017-02-03T04:24:20.625Z",
modifiedTime: "2017-03-24T09:18:54.354Z",
tags: [
  "vip",
  "member"
],
birthday: "1989-01-01",
memberId: "ABC123"
}

```

Customer Schema

refId	UUID	uniquely identify a customer
firstName	String	customer's first name
lastName	String	customer's last name
email	String	customer's email
phone	String	customer's phone number
address1	String	street line 1 of customer's address
address2	String	street line 2 of customer's address
city	String	city of customer's address
state	String	state of customer's address
postalCode	String	postal code of customer's address
memberId	String	customer's member Id
birthday	String	customer's birthday, format is YYYY-MM-DD
tags	Array of String	tags of the customer, empty array returns if there's no tags
loyalty	Number	the store credit amount the customer has. This field is NOT supported in Create Customer API and is ONLY returned in Get Customer By RefId API if store credit is enabled for this account.

storeCreditsBalance	Number	the store credit amount the customer has this field is NOT supported in Create Customer API and ONLY for get customer API, if customer don't have store credit, response not show this field
cashbackBalance	Number	the store credit amount the customer has this field is NOT supported in Create Customer API and ONLY for get customer API, if customer don't have store credit, response not show this field
createdTime	String	timestamp when the customer is added
modifiedTime	String	timestamp when the customer is modified last time

2. Get Customer by RefId

URL	/customers/<refId>
HTTP Verb	GET

Response

If no customer can be found with specified refId, status code 404 is returned.

Otherwise the customer object of specified refId is returned, please note that the response schema is same as the [Create Customer POST](#).

3. Update Customer

URL	/customers/<refId>
HTTP Verb	PUT

Request Body

firstName	String	Required	customer's first name
lastName	String	Required	customer's last name
email	String	Optional	customer's email

phone	String	Optional	customer's phone number
address1	String	Optional	street line 1 of customer's address
address2	String	Optional	street line 2 of customer's address
city	String	Optional	city of customer's address
state	String	Optional	state of customer's address
postalCode	String	Optional	postal code of customer's address
tags	Array of String	Optional	tags of the customer. Multiple tags can be added separated by a comma.
modifiedTime	String	Optional	timestamp when the customer is modified. If this is missing in the request, current time is used when the request is processed

Response

If no customer can be found with specified refId, status code 404 is returned.

Otherwise, the customer data is replaced with the data passed in the request body. **Partial update is not supported.** That means if some property is missing in the update body, the existing value of the property is removed.

If the update is performed successfully without errors, customer object is returned. Please refer [here](#) for customer schema.

4. Get All Customers

URL	/customers
HTTP Verb	GET

Response

Returns an array of customers. If no customer can be found, empty array is returned.

The Transfer-Encoding is **chunked**.

Please reference [here](#) for Customer schema

5. Search Customers

URL	/customers
HTTP Verb	GET

Request Query Parameters

Customers can be searched by **first name/last name/email/phone**. Query is specified by query parameters. Accepted query parameters are **firstName**, **lastName**, **email**, **phone**.

When searching using **firstName** or **lastName**, all results that begin with the specified parameter value will be returned. When searching using **email** or **phone**, results containing the specific parameter value will be returned. If multiple query parameters are specified, AND logic will be applied to the search.

e.g <https://api.storehubhq.com/customers?email=abc@storehub.com> returns all customers with email contains "abc@storehub.com"

Response

Returns an array of customer objects that matches query.

If search results is more than 100, only first 100 results are returned sorted by last name and then first name.

Please reference [here](#) for customer schema.

Inventory

1. Get Inventory

URL	/inventory/<storeId>
HTTP Verb	GET

Response

If no store is found with specified **storeId**, status code 404 is returned. Otherwise it returns an array of stocks. Only products that configured to track stock level has stock information. If you don't set **Warning Stock Level** or **Ideal Stock Level**, will only return **Quantity**.

```
{  
  productId: "58affc2b8d2ffae03a385909",  
  quantityOnHand: 1,  
}
```

```

        warningStock: 10
    },
    {
        productId: "58affc2b8d2ffae03a38590c",
        quantityOnHand: 2,
        idealStock: 12
    },
    {
        productId: "58affc2b8d2ffae03a385906",
        quantityOnHand: 2,
        warningStock: 10,
        idealStock: 12
    }
}

```

Stock Schema

productId	String	unique Id of the product
quantityOnHand	Number	quantity of product available in stock
warningStock	Number	quantity of product warning stock level
idealStock	Number	quantity of product ideal stock level

Transaction

1. Get Transactions

URL	/transactions
HTTP Verb	GET

Request Query Parameters

from	String	Optional, format is YYYY-MM-DD, e.g. 1970-01-01 default to Jan 1st, 1970 if not specified	transactions with created time is later or equal than this date are returned
------	--------	--	--

to	String	Optional, format is YYYY-MM-DD, e.g. 1970-01-01 default to current day if not specified	transactions with created time is earlier or equal than this date are returned
storeId	String	Optional default to all stores if not specified	transactions of the store with specified storeId are returned
includeOnline	Boolean	Optional	online orders would be included in the returned transactions
onlineOnly	Boolean	Optional	only online orders are returned

Response

Returns an array of transactions. If no transactions can be found, empty array is returned.

Only up to **5000** transactions are returned within one API call.

Note: return date is ISO date.

```
[
  {
    "refId": "BE7606B7-EE3D-4D2C-93A1-B3594FAFFAC7",
    "invoiceNumber": "0001808071236168",
    "storeId": "5b62d783081ccd016185cf08",
    "registerId": "5b62d783081ccd016185cf09",
    "employeeId": "5b62d783081ccd016185cf0b",
    "transactionType": "Sale",
    "transactionTime": "2018-08-07T04:36:16.473Z",
    "total": 473,
    "subTotal": 473,
    "tax": 0,
    "discount": 0,
    "roundedAmount": 0,
    "serviceCharge": 0,
    "promotions": [],
    "items": [
      {
        "productId": "5b641ae7278bb801617052b2",
        "quantity": 1,
        "total": 473,
        "subTotal": 473,
        "taxCode": "gst",
        "discount": 0,
        "unitPrice": 473,
```

```

        "itemType": "Item",
        "promotions": []
    }
],
"payments": [
    {
        "paymentMethod": "Cash",
        "amount": 473
    }
],
"isCancelled": false,
"channel": "OFFLINE_PAYMENTS"
},
{
    "refId": "864ece52-902d-4dfc-ad35-10fc1eff9137",
    "invoiceNumber": "813366324236434",
    "storeId": "5b62d783081ccd016185cf08",
    "registerId": "",
    "transactionType": "Sale",
    "transactionTime": "2018-08-10T06:40:02.377Z",
    "customerRefId": "5b613aae5e1fe60074d57f24",
    "total": 258.8,
    "subTotal": 253.8,
    "tax": 0,
    "discount": 0,
    "roundedAmount": 0,
    "serviceCharge": 0,
    "promotions": [],
    "items": [
        {
            "productId": "5b69083a891f4e0161b3af68",
            "quantity": 1,
            "total": 253.8,
            "subTotal": 253.8,
            "taxCode": "gst",
            "discount": 0,
            "unitPrice": 253.8,
            "itemType": "Item",
            "notes": "comment",
            "promotions": [],
            "selectedOptions": [
                {
                    "groupId": "5b69083a891f4e0161b3af64",
                    "optionId": "5b69083a891f4e0161b3af67",
                    "quantity": 1
                }
            ]
        }
    ]
},
"payments": [
    {
        "amount": 258.8
    }
]

```

```

    }
  ],
  "isCancelled": false,
  "channel": "ONLINE_PAYMENTS",
  "deliveryInformation": [
    {
      "address": {
        "updatedAt": "2018-08-10T06:40:02.390Z",
        "createdAt": "2018-08-10T06:40:02.390Z",
        "name": "viji",
        "phone": "0123456789",
        "address": "C-G-31, 10 Boulevard, Lebuhraya SPRINT, Kayu Ara, ",
        "city": "Petaling Jaya",
        "postCode": "47400",
        "state": "Selangor",
        "country": "MY",
        "companyName": "StoreHub"
      },
      "shippingFee": 5,
      "courier": "",
      "trackingId": "",
      "deliveryMethodInfo": {
        "shippingZoneName": "Selangor",
        "deliveryMethodName": "Standard",
        "rate": 5,
        "minShippingTime": 1,
        "maxShippingTime": 2,
        "isFree": false
      }
    }
  ],
  "contactDetail": {
    "email": "viji.ravi+1@storehub.com",
    "phone": "0123456789",
    "name": "viji"
  },
  "pickupInformation": [],
  "shippingFee": 5,
  "pendingPaymentStartTime": null,
  "status": "created",
  "shippingType": "delivery",
  "shippingFeeDiscount": null
}]

```

Transaction Response Schema

refId	UUID	uniquely identity of the transaction
invoiceNumber	String	invoice number of the transaction

storeId	String	Id of the store which the transaction is associated with
registerId	String	Id of the register which the transaction is processed by
employeeId	String	Id of the employee who processed the transaction
transactionType	Enum	possible values are Sale/Return
transactionTime	Date	Time at which the transaction occurred (in UTC time)
customerRefId	UUID	refId of the customer which the transaction is associated with
total	Number	total amount of the transaction
subTotal	Number	subTotal amount of the transaction
discount	Number	discount applied to the transaction
tax	Number	tax applied to the transaction
roundedAmount	Number	rounded amount of the transaction
items	Array of Item Response	products purchased/returned in the transaction
payments	Array of Payment Response	payments associated of the transaction
isCancelled	Boolean	true if the transaction is cancelled
cancelledBy	String	Id of employee who cancelled the transaction
cancelledTime	Date	time when the transaction is cancelled
comment	String	comment on the transaction
returnReason	String	the reason the purchased item was returned
saleInvoiceNumber	String	if it's a returned transaction, this is the invoice number of sale transaction that the return is associated with
serviceCharge	Number	the amount of service charge applied to the transaction

promotions	Array of AppliedPromotion response	promotions that have been applied at the transaction level. These promotions are independent from the promotions applied at line item level.
tableId	String	table info of the transaction
deliveryInformation	Array of DeliveryInformation response	delivery information array
contactDetail	ContactDetail	contact detail
pickupInformation	Array of PickupInformation response	pick up information
shippingFee	Number	shipping fee
pendingPaymentStartTime	Date	pending payment start date
status	String	order status
shippingType	String	delivery or pickup
shippingFeeDiscount	Number	shipping fee discount
channel	String	ONLINE_PAYMENTS: 1, OFFLINE_PAYMENTS: 2, BEEP_ORDERS: 3

Item Response Schema

productId	String	Id of sold/returned product
itemType	Enum	possible values are Item/Discount/ServiceCharge
quantity	Number	sold/returned quantity of the product
total	Number	total of the item

subTotal	Number	subTotal of the item
discount	Number	discount applied to the item
notes	String	comment for the item
tax	Number	tax applied to the item
taxCode	String	tax code applied to this line item
unitPrice	Number	unit price of the item if the product's priceType is Variable
promotions	Array of AppliedPromotion response	promotions that have been applied to this line item
tableId	String	table info of the transaction

Payment Response Schema

paymentMethod	String	Possible values are Cash/CreditCard/Loyalty, or custom payment options defined in BackOffice
amount	Number	amount paid with this payment

AppliedPromotion Response Schema

id	String	unique id of the applied promotion
name	String	applied promotion's name as configured in StoreHub BackOffice
discount	Number	discount applied due to this promotion, discounted amount is deducted from the subtotal without tax
tax	Number	tax deducted due to this promotion, when this number is not zero. The total discounted amount due to this promotion is (discount - tax) since tax is negative here.

DeliveryInformation Response Schema

address	address response	customer address information
courier	String	courier
trackingId	String	tracking id
shippingFee	Number	shipping fee
deliveryMethodInfo	delivery method information response	customer delivery method information

address Response Schema

updatedAtTime	Date	address update date
createdTime	Date	address create date
name	String	customer name
phone	String	customer phone
address	String	customer address
city	String	customer city
postCode	String	customer postcode
state	String	customer state
country	String	customer country
companyName	String	customer company name

DeliveryMethodInfo Schema

shippingZoneName	String	shipping zone name
deliveryMethodName	String	delivery method name
rate	Number	shipping fee
minShippingTime	Number	min shipping time
maxShippingTime	Number	max shipping time

shippingTimeUnit	String	shipping time unit(day or hour)
isFree	Boolean	if free shipping

ContactDetail Schema Response

email	String	email address
name	String	name
phone	String	phone number

PickupInformation Schema Response

store	stores response	store information
-------	---------------------------------	-------------------

1. Add Transaction

URL	/transactions
HTTP Verb	POST

Request Body

refId	UUID	Required	unique identity of the transaction
invoiceNumber	String	Required*	invoice number of the transaction
storeId	String	Required	store which this transaction belongs to
employeeId	String	Optional	specifies which employee processed the transaction
transactionType	Enum	Required	allowed values are Sale/Return (case sensitive).
transactionTime	Date	Required	specifies when this transaction happened
paymentMethod	Enum	Required	allowed values are Cash/CreditCard (case sensitive).

customerRefId	UUID	Optional	refId of the customer that this transaction is associated with
total	Number	Required	total amount of the transaction
subTotal	Number	Required	subTotal amount of the transaction
tax	Number	Optional	tax amount applied to the transaction
discount	Number	Required	discount applied to the transaction. specify 0 if no discount
roundedAmount	Number	Optional	rounded amount of the transaction default to 0
comment	String	Optional	comment on the transaction
items	Array of Item	Required	purchased or returned items of the transaction
returnReason	String	Optional	the reason the purchased item was returned
saleInvoiceNumber	String	Optional	If this is a Return transaction, specifies the invoiceNumber of sale transaction that this return associated with
tableId	String	Optional	table info of the transaction

Item Schema

productId	String	Required	specifies the product Id
quantity	Number	Required	purchased/returned quantity of the product
total	Number	Required	total of this line item
subTotal	Number	Required	subTotal of this line item
tax	Number	Optional	tax applied to this line item
discount	Number	Required	discount applied to this line Specify 0 if no discount applied.

unitPrice	Number	Optional	If the product's priceType is 'Variable', specifies the unitPrice on item which is determined when check-out
-----------	--------	----------	--

Notes:

- 1) There's no validation on whether sum-up of subtotal/total/discount/tax on each item equals to subtotal/total/discount/tax of the transaction
- 2) All discounts should be specified in absolute value instead of percentage.
- 2) Return transactions needs to be performed as a different transaction, which can be associated with the related Sale transaction by specifying the matching saleInvoiceNumber.
- 3) There's no validation on whether the returned items are actually part of the sale transaction or has been returned already not even if the saleInvoiceNumber is specified.
- 4) All numbers should be tax exclusive. It's expected that **total = subtotal + tax - discount + roundedAmount + serviceCharge** at transaction level and **total = subTotal + tax - discount** at item level.

Response

If an existing transaction with same refId can be found, the request is ignored and the existing transaction is returned.

Otherwise the transaction is added. And the added transaction is returned.

Please reference **Transaction Response Schema** in Get Transactions section for response

3. Cancel Transaction

URL	/transactions/<refId>/cancel (refId is the refId of transaction to be cancelled)
HTTP Verb	POST

Request Body

cancelledTime	Date	Required	specify cancelled time.
cancelledBy	String	Optional	specify employee Id who cancelled the transaction

Response

If the transaction with specified refId cannot be found, status code 404 is returned.

Otherwise the transaction is cancelled and the response object is returned. Please reference [here](#) for transaction schema.

Notes:

- 1) Only Sale transactions can be cancelled

Employees

1. Get All Employees

URL	/employees
HTTP Verb	GET

Query Parameters

modifiedSince	Date	Optional	if specified, only returns those employees added or modified equal to or after the specified date time using format YYYY-MM-DD, e.g. "2015-10-01"
---------------	------	----------	---

Response

An array of employees are returned. Please reference the Employee schema below.

```
{
  id: "589fe4ffb92508602f477428",
  firstName: "John",
  lastName: "Doe",
  email: "john@gmail.com",
  phone: "0123456789",
  createTime: "2017-02-12T04:30:55.601Z",
  modifiedTime: "2017-02-15T08:02:17.472Z"
}
```

Employee Schema

id	String	Unique Id of the employee
firstName	String	first name of the employee

lastName	String	last name of the employee
email	String	email of the employee
phone	String	phone of the employee
createdTime	Date	the time of the employee added to StoreHub system
modifiedTime	Date	the last time the employee gets modified

Stores

1. Get All Stores

URL	/stores
HTTP Verb	GET

Response

An array of all stores are returned. Please reference the Store schema below.

```
{
  id: "580c5da45c196d8304aa4986",
  name: "ew",
  address1: "1, The Intermark",
  address2: "Jalan Tun Razak",
  city: "Kuala Lumpur",
  state: "Kuala Lumpur",
  country: "Malaysia",
  postalCode: "50400",
  phone: "0321660000",
  email: "ew@gmail.com",
  website: "ew-shop.com"
}
```

Store Schema

id	String	Unique Id of the store
name	String	name of the store
address1	String	address line 1 of the store
address2	String	address line 2 of the store
city	String	the city the store is located in

state	String	the state the store is located in
country	String	the country the store is located in
postalCode	String	postal code of the store
phone	String	store's phone number
email	String	store's email address
website	String	store's website

Timesheets

1. Search Timesheets

URL	/timesheets
HTTP Verb	GET

Query Parameters

storeId	String	Unique Id of the store [Optional]
employeeId	String	Unique Id of the employee [Optional]
from	Date	search the records employee clock in after this time [Optional]
to	Date	search the records employee clock in before this time [Optional]

Response

An array of all timesheet records meet the search conditions.

```
{
  employeeId: "xxxx",
  storeId: "xxxx",
  clockInTime: "2019-01-31T11:11:04.013Z",
  clockOutTime: "2019-01-31T11:11:04.013Z",
}
```

Timesheet Schema

employeeId	String	Unique Id of the employee
------------	--------	---------------------------

storeId	String	Unique Id of the store that employee created this timesheet
clockInTime	String	When the employee clock in
clockOutTime	String	When the employee clock out

Revision History

Date	Version	Description
-	1.0	Initial draft
28th March 2017	1.1	- Added response object examples - Refined descriptions and definitions
21st June 2017	1.2	Added store credit balance field to customer object
31st Jan 2019	1.3	Add timesheets api document
1st Nov 2019	1.4	Add tableId to Transaction API
12th Nov 2019	1.5	Update get Transaction api