Altametrics

Anyconnector Reference Document

Version 0.1

Prepared by: **Altametrics**

Feburary 22, 2018

Altametrics, LLC
3191 Red Hill Avenue
Costa Mesa, CA 92626 USA

Tel: (800) 676-1281

Last edited: 27 September 2018

Table of Content

1. Introduction

The purpose of this document is to provide detailed description of data field required to fetch and post the data from Hubworks, also contains the type of file format that Hubworks will support to display the data. In Any connector we are trying to fetch the sales, employees, menu-mix, invoices, and ordering etc. data from POS and try to post the data of employees and schedules.

2. Data Transmission Files

There are basically two types of files:

1. **Configuration File:** It is basically configuration file where different parameters are defined with their unique ids and name. This file will be transmitted when there is any modification or addition of any new parameter. This table acts as a look-up table for transaction file

2. **Transaction file:** This file will transmitted on daily basis that contains the sales data, employee data, menu-mix data etc.

3. File naming convention of both the files should be as per the format given below:

<filename>_storeID_<MMDDYYYY>

2.1 Configuration File

Configuration files are basically a file containing various look ups for the transaction data to work on to and create appropriate data object for use in Hubworks applications. Look-Up is basically a sort of object index that transaction file will refer to do correct mapping of data values. This file will contain the following look-ups:

2.1.1 Jobs

This look-up contains the details of all the positions that are configured at the POS system. Each position will be uniquely identified using its ID and notation of the same will be done based on the supplied name of the position.

Sample JSON:

```
{
        "jobs": [{
                "id": “701”,
                "name": "Chef",
                "isSalaried": true,
                "wage": 0.0
        }]
}
```

Parameter Details in Jobs

Parameter	Description	Type	Required
Id	ID of the job.	string	Yes
name	Name of the job.	string	Yes
isSalaried	If the position is salaried (non-hourly pay)	boolean	No
wage	Hourly pay rate of job.	Decimal(2)	No

2.1.2 Categories

This look-up contains an array of different categories and each category should have the fields mentioned in the parameter details table below. The details of each category that are uniquely identified using its id.

Sample JSON:

```
{
        "categories": [{
                "id": “500”,
                "name": "Burgers"
        }]
}
```

Parameter Details of Categories

Parameter	Description	Type	Required
id	ID of the item category	string	Yes
name	Name of the item category	string	Yes

2.1.3 Items

This lookup contains the details of menu items that are uniquely identified using its id and name.

Sample JSON:

```
{
        "items": [{
                "id": “5001”,
                "name": "Bacon Avocado Cheeseburger",
                "categoryID": “500”,
                "price": 11.59
        }]
}
```

Parameters Details of items:

Parameter	Description	Type	Required
id	The ID of an item	string	Yes
name	The name of the item.	string	Yes
categoryID	The name of the category	string	Yes
price	Price of an item	Decimal(2)	No

2.1.4 Revcenter

This lookup contains the details of sales by type from where the order is being placed. They are uniquely identified using its id and name.

Sample JSON:

```
{
        "revCenters": [{
                "id": “1”,
                "name": "Dining Room"
        }]
}
```

Parameters Details of revCenters:

Parameter	Description	Type	Required
id	The ID of the revCenter	string	Yes
name	The name of the revCenter	string	Yes

2.1.5 Voids

This lookup contains the details of the voids due to which the items in a check are being canceled. These voids are defined with a unique id and their name or reason.

Sample JSON:

```
{
        "voids": [{
                "id": "V011",
                "name": "Unavailable"
        }]
}
```


Parameters Details of Voids:

Parameter	Description	Type	Required
id	The ID of the void	string	Yes
name	The reason for void	string	Yes

2.1.6 Discounts

This lookup contains the details of the discount that are being used in the check when an order is placed. It is identified by their unique id and name.

Sample JSON:

```
{
	"discounts": [{
		"id": "C011",
		"name": "OTH 100% Off"
	}]
}
```


Parameters Details of Discount:

Parameter	Description	Type	Required
id	The ID of the discount	string	Yes
name	Discount name	string	Yes

2.1.7 Payment types

This lookup contains the details of all the payment type that are configured at the POS system. Each payment type will be uniquely identified using its ID and name.

Sample JSON:

```
{
        "paymentTypes": [{
                "id": “1”,
                "name": "Cash",
                "paymentGroup": "CASH"
        }]
}
```


Parameters Details of payment types:

Parameter	Description	Type	Required
id	The ID of the payment type	string	Yes
name	Name of payment type	string	Yes
payment group	Name of payment group	string	Yes

2.1.8 Paid In/Out Payment

This Lookup contains the detail of all the Paid In/Out payments that are configured in the POS system.


```
{
        "paidInOut": [{
                "id": “1”,
                "name": "Food",
                        }]
}
```

Each Paid In/Out type will be uniquely identified using its Id and Name.

Sample JSON

Parameters Details of paid In/Out Payment:

Parameter	Description	Type	Required
id	The Id of the payment (Paid In/Out) reason	string	Yes
name	The name Of the payment (Paid In/Out) reason	string	Yes

2.1.9 Employees

This lookup contains the details of all the employee that are uniquely identified by their id and first name and jobs are assigned to employee.

Sample JSON:

```
{
"employees": [{
"id": “23”,
"firstName": "Drew",
"lastName": "Barrymore",
"empNum": 99998889,
"roleID": “1”,
"empJobs": [{
"id": “701”,
"wage": 3.45,
"isPrimary": false
}]
}]}
```


Parameters Details of employees:

Parameter	Description	Type	Required
id	The ID of an employee	string	Yes
First name	First name of an employee	string	No
Last name	Last name of an employee	string	No
empNum	Employee Number of an employee	integer	Yes

roleId	Role Id of an employee	String	Yes
empJobs			
Id	ID of the job.	String	Yes
wage	Hourly pay rate of job.	Decimal(2)	No
isprimary	Default job of the Employee	boolean	No

3. Transaction file:

Transaction file is the file containing daily transactions data of a single store. Transaction data refers to four types of data checks, paidins & paidouts, deposits, and shifts (timekeeping data).

One or more types of transaction data can be sent in the same request.

3.1 Check Data

A “check” refers to a single customer bill of sale. Within every “check” there are several data types. Check Info, Check Discounts, Check Items (includes voids, discounts, and modifiers), and Check Payments.

Sample JSON:

```
{
"checks": [
{
"checkNum": "10001",
"orderNum": “1007”,
"employeeID": “104”,
"openedAt": "2018-04-10T09:09:00",
"closedAt": "2018-04-10T10:42:00",
"revCenterID": “1”,
"lastModifiedAt": "2018-04-10T10:42:00",
"guestCount": 1,
"tableNum":”2”,
"inclusiveTax": 0.0,
"exclusiveTax": 0.25,
"total": 3.14,
"busiDate": "2018-04-10",
"busiTime": "09:09:00",
"items": [{
"itemID": “355”,
"quantity": 1.0,
"price": 0.0,
"inclusiveTax": 0.0,
"amount": 0.0
},
```

```
{
"itemID": “304”,
"quantity": 1.0,
"price": 2.89,
"inclusiveTax": 0.0,
"amount": 2.89
}
],
"payment": [{
"total": 6.54,
"paymentTypeID": “1”,
"received": 10.00,
"change": 3.46,
"tip": 0.0,
"appliedAt": "2007-03-29T18:10:00"
},
{
"total": 5.54,
"paymentTypeID": 1,
"received": 10.00,
"change": 4.46,
"tip": 0.0,
"appliedAt": "2007-03-29T18:10:00"
}],
```

```
“discounts": [{
"id": "2",
"name": "Comp",
"amount": 3.15,
"itemID": “5011"
}],
"voids": [{
"id": "12",
"qty": 3.0,
"amount": 15.15,
"itemID": "5011"
}
] } ] }
```

3.1.1 Check Info

The general check information. Here are the fields for a check:

FIELD	TYPE	REQUIRED	DESCRIPTION
checkNum	string	Yes	Unique ID for a check for a given POS in a location.
orderNum	string		Order number which is sometimes associated with a check in some POS systems
businessDate	date	Yes	Date used for reporting the sales
businessTime	time	Yes	Time used for reporting the sales. This is a local time (not UTC) and is used so that we can easily report on sales from 3-4pm at multiple locations across different time zones.
openedAt	datetime	Yes	Date & time the check was opened
closedAt	datetime		Date & time the check was closed, if this is NULL, we will report the check as being “open”
lastModifiedAt	datetime	Yes	Last time the check was updated
employeeId	string		ID for Employee assigned to the check. From the configuration data.
revCenterId	string		The revenue center ID. From the configuration data.
guestCount	integer		Total number of guests. If QSR, just default to 1.
tableNum	string		Table number
inclusiveTax	decimal(2)		Total amount of inclusive tax
exclusiveTax	decimal(2)		Total amount of exclusive tax
autoGratuity	decimal(2)		Total amount of auto gratuity applied to the check
total	decimal(2)	Yes	The check total

3.1.2 Check Items

A check can have multiple items applied to it. Here are the fields for an item:

FIELD	TYPE	REQUIRED	DESCRIPTION
itemId	string	Yes	Sales item ID. From the configuration data.

quantity	integer	Yes	Total quantity sold
amount	decimal(2)	Yes	Total sales amount for this item (this amount SHOULD INCLUDE discount/comp but SHOULD NOT INCLUDE taxes)
inclusiveTax	decimal(2)		Inclusive tax amount
exclusiveTax	decimal(2)		Exclusive tax amount

3.1.3 Check Payments

A check can have multiple payments applied to it. Here are the fields for a payment:

FIELD	TYPE	REQUIRED	DESCRIPTION
paymentTypeId	string	Yes	ID for the payment type
total	decimal(2)	Yes	Total amount of the payment applied towards the check total
received	decimal(2)	Yes	Total amount received from the customer
change	decimal(2)	Yes	Total amount of change given to the customer
tip	decimal(2)		Credit card tip amount
applied_at	datetime		Date & time this payment was applied

3.1.4 Check Discount

A check can have multiple discounts applied to it. Here are the fields for a discount:

FIELD	TYPE	REQUIRED	DESCRIPTION
id	string	Yes	The Id of the item.
quantity	decimal(2)		The Quantity of item on which discount is applied.
amount	decimal(2)		The total amount applied as a Discount.
Item Id	string	Yes	The id of the item on which Discount is applied.

3.1.5 Check Void

A check can have multiple discounts applied to it. Here are the fields for a discount:

FIELD	TYPE	REQUIRED	DESCRIPTION
id	string	Yes	Id of the void
amount	decimal(2)	Yes	Total amount of the void.
quantity	decimal(2)	Yes	Quantity of the voided item
itemID	string	Yes	The ID of the voided item

3.1.6 Shift Data

A “shift” refers to a single employee’s shift or timeslip that was generated from an employee clocking in and out.

Sample JSON:

```
{
"shifts": [
{
"id": “10090”,
"employeeID": “1009”,
"jobID": “3”,
"startedAt": "2007-03-28T00:00:00",
"busiDate": "2007-03-27",
"startTime": "00:00:00",
"endTime": "00:00:00",
"endedAt": "2007-03-28T00:00:00",
"shiftUpdatedAt": "2007-03-28T00:00:00",
"totalTime": 300.0,
"totalPay": 47.5,
"payRate": 9.5,
"cashTips": 0.0,
"ccTips": 0.0,
"otPayRate": 0.0,
"paidBreakMinutes": 0.0,
"unpaidBreakMinutes": 0.0
}
]
}
```

FIELD	TYPE	REQUIRED	DESCRIPTION
id	string	Yes	Unique ID for a shift in a location. If the POS System does not provide a unique ID, you can try combining the given shift ID with the shift start time.
employeeId	string	Yes	employee ID for the shift
jobId	string	Yes	job ID for the shift
businessDate	date	Yes	date used for reporting the labor
startTime	time	Yes	time used for reporting the labor
endTime	time		business time the shift ends
startedAt	datetime	Yes	date & time the shift was started
endedAt	datetime		date & time the shift ended
shiftUpdatedAt	datetime	Yes	last time the shift was updated
totalTime	decimal(2)	Yes	total time for the shift (including overtime)
totalPay	decimal(2)	Yes	total cost for the shift (including overtime)
payRate	decimal(2)		pay rate
otPayrate	decimal(2)		pay rate for overtime
ccTips	decimal(2)		total credit card tips for the employee shift
cashTips	decimal(2)		total cash tips declared by the employee at clock out
paidBreakMinutes	decimal(2)		total number of paid break minutes
unpaidBreakMinutes	decimal(2)		total number of unpaid break minutes

3.1.7 Shift Break Detail

The shift info above has fields for break totals, but break details can also optionally be added (multiple breaks are allowed). Here are the fields for break detail:

FIELD	TYPE	REQUIRED	DESCRIPTION
startedAt	datetime	Yes	date & time for the start of the break
endedAt	datetime	Yes	date & time for the end of the break

3.1.8 Paid In/Out Payment Data

A “paid in/out payment” refers to an incoming or outgoing payment using the paid in/out types from the configuration data. Negative amounts indicate paid out

Sample JSON:

```
{
        "paidInOut": [{
                "id": “P100”,
                "amount": 100.0,
                “paidAt”: “2017-03-28T00:00:00”,
                “paymentTypeID”: “1”,
                “busiDate”: “2017-03-28”,
                “lastModifiedAt”: “2017-03-28T00:00:00”,
                “paidInOutID": “1”,
                “employeeID”: “23”,
               “custAccountID”: “13”,
               “tip”: 2.0


        }]
 }
```


3.1.8 Cash Deposit

Cash Deposit refers to the amount of cash present in the cash drawer at the end of the day.

Here are the fields for cash deposit:

Sample JSON:


```
{
"version": "1.0",
"deposits": [
{
  "id": 73309,
  "amount": 1196,
  "busiDate": "2018-04-10",
  "lastModifiedAt": "2007-03-28T00:00:00",
  "depositedAt": "2007-03-28T00:00:00",
  "employeeID": 4018
},
{
  "id": 73312,
  "amount": 166,
  "busiDate": "2018-04-10",
  "lastModifiedAt": "2007-03-28T00:00:00",
  "depositedAt": "2007-03-28T00:00:00",
  "employeeID": 4018
}
]
}
```

FIELD	TYPE	REQUIRED	DESCRIPTION
id	string	Yes	unique identifier of the cash deposit
amount	Decimal(2)	Yes	Total amount deposited in the cash deposit/drawer
busiDate	date		Date used for submitting deposit
lastModifiedAt	datetime	yes	Last time the cash drawer amount was updated
depositedAt	datetime	Yes	Amount submitted into drawer
employeeID	string	Yes	employee ID for cash deposit