

eRetail3.2 Data Docking



شركة كويت كارپوريشن ليميتد
KUWAIT CORPORATION LIMITED

1. Introduce

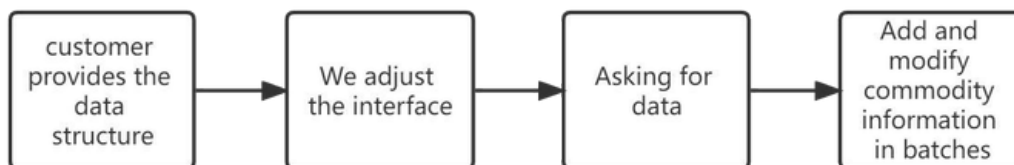
This document is for eRetail3 1 System docking method description, eRetail3 1The system supports three docking methods: DataSync / API, eRetail 3.2 and hardware devices The API is a Restful-style web API that is responsible for providing an interface for the customer system to actively push commodity data; Data Sync is a back-end service that actively captures data from the client's system (e.g. Database, WebAPI, Excel, etc.); The SDK is responsible for managing product data, hardware device information, scheduling price change tasks, etc.;

2. Docking Methods

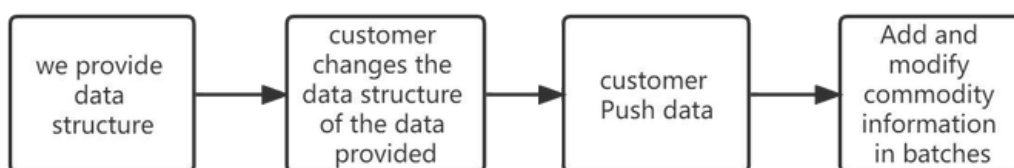
Developed by the customer to connect to eRetail3 1 webapi interface mode, in which we can actively request data as the initiator or receive the data sent by the customer as the receiver.

2.1. Docking Flow

Solution 1: Take the initiative to obtain, the data structure is provided by the customer, and we adjust the interface according to the customer data structure to complete the data docking.

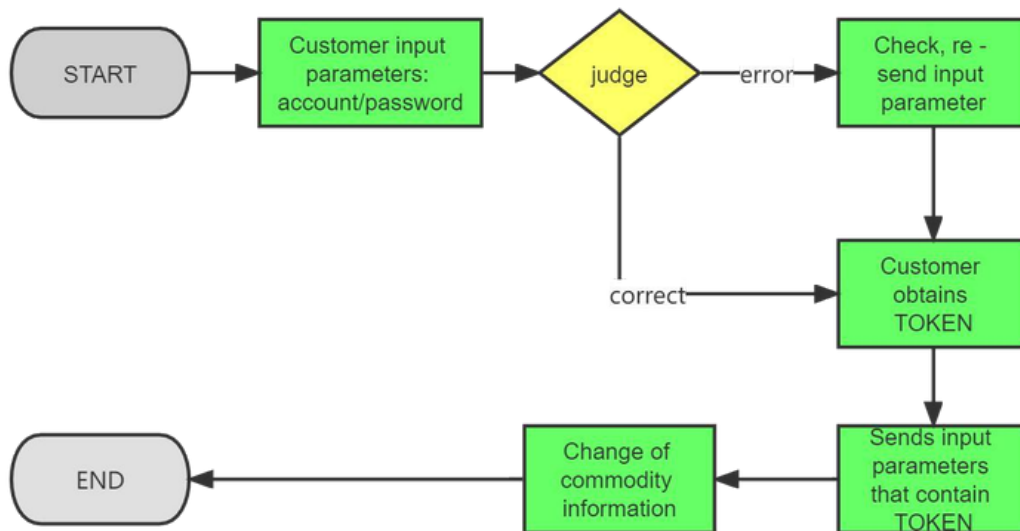


Solution 2: Passive acceptance, we provide the data structure, the customer provides data according to our data structure, and completes the data docking.



2.2. Passive docking mode:

2.2.1. Interface description



2.2.2. Call the interface's methods

(1) Handshake interface

Purpose: This interface is used for authentication. All subsequent interfaces depend on the data obtained by this interface.

HTTP POST

URL: `http:// 192.168.1.92:5000/api/login`

Content-Type: `application/json`

Request parameters:

Parameter name	type	description
userName	String	Username
password	String	password

Return format:

Parameter name	type	illustrate
code	Int	0:Success,Other>Error
message	String	Success or error messages
body	JsonNode	Message body
token	String	The token for subsequent sessions

Note: After obtaining the token, you need to add this content to the header of subsequent HTTP requests. For example: "Authorization: Bearer {token}"

Example:

Request

```
{
```

```
"userName": "admin",
```

```
"password": "Pass99"
}
```

Response

```
{
  "code": 0,
  "message": "success",
  "body": {
    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."
  }
}
```

(2) Product data interface

Purpose: This interface is used to add and update multiple product data.

HTTP POST

URL: http://

192.168.1.92:5000/api/goods/saveList Content-

Type: application/json **Request parameters:**

Parameter name	type	description
shopCode	String	Store number, you need to create the corresponding store in the system store list first
template	String	For the template name, you need to create the corresponding template in the template list first
items	ArrayString[26]	Product data attribute, temporarily fixed at 26 (extensible).

Note: For store and template creation, please contact ETEL sales or pre-sales/after-sales support engineers.

Return format:

Parameter name	type	illustrate
code	Int	0:Success, Other:Error
message	String	Success or error messages
body	JsonNode	Message body

Note: After obtaining the token, you need to add this content to the header of subsequent HTTP requests. For example: "Authorization: Bearer {token}" **Example:**

Request

```
[
  {
    "shopCode": "0001", //storenumber
    "template": " REG ", // template type: REG-regular template, SAL-promotion template,
    NOR-none stock template, MER-membership template..... "items": [
      "A050", // client shop number
      "123456", // Product unique code: can be product number or product barcode "Product 1",
      //Product Name
      "123456789012", //UPC1: product barcode 1
      "123456789013", //UPC2: product barcode 2
      "123456789014", //UPC3: product barcode 3
      "8.98", //price 1: Retail price or original price
      "8.95", //price 2: promotion price
      "8.96", //price 3: membership price
      "Shanghai", //origin
      "300ml", //specifications
      "Bottle", //unit
      "Qualified", //grade
      "2021/12/20", //promotion start date
      "2021/12/25", //promotion end date
      "http://www.baidu.com", //QR code
      "Zhang San", //employee name
      "5.1", //stock
      "Extension 1", //Extension Field 1 is not empty ""
      "Extension 2", //Extension Field 2
      "Extension 3", //Extension Field 3
      "Extension 4",
      "Extension 5",
      "Extension 6",
      "Extension 7",
      "Extension 8",
      "Extension 9",
      "Extension 10"
    ]
  }
]
```

```
}  
]
```

Response

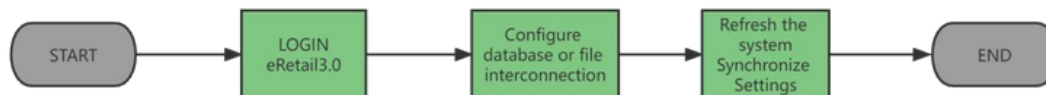
```
{  
  "code": 0,  
  "message": "success",  
  "body": "121f5151fdffds21cdf"  
}
```

Note: By default, the field properties in this example are fixed. In fact, the product data field content and template can be set freely.

2.3. Proactive acquisition methods

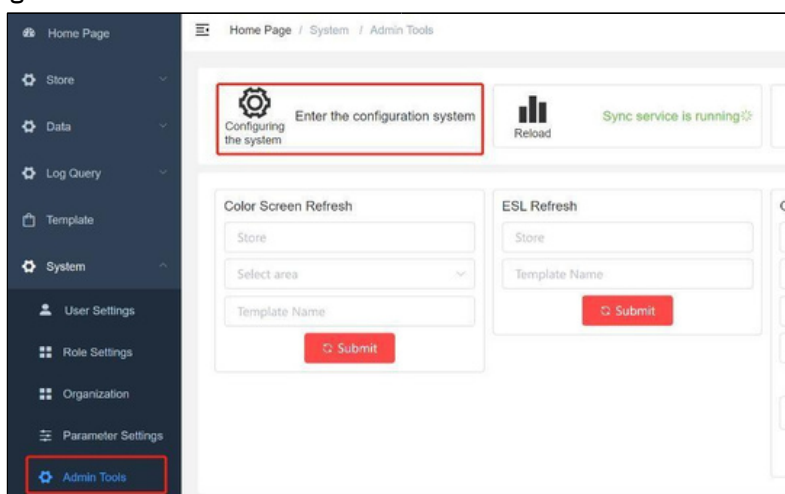
Via eRetail 3 1 D2M module, docking ERP/POS data, support the use of database, formatted text for data docking

2.3.1. Businessdockingflowchart

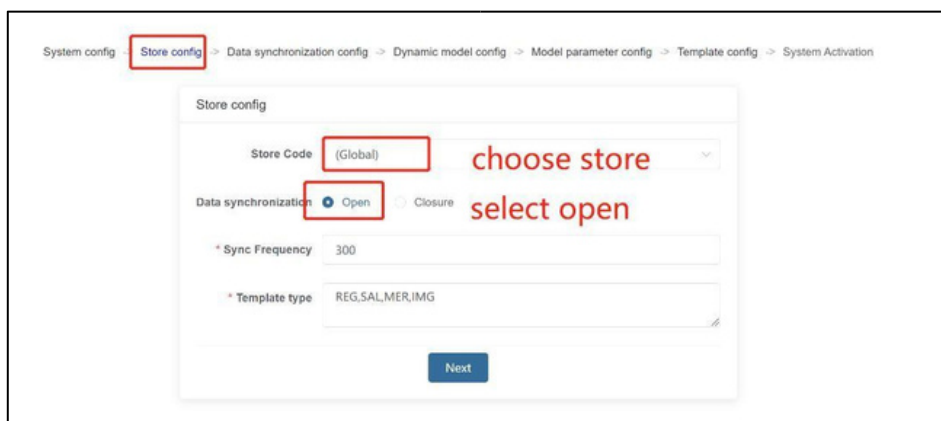


2.3.2. Databasedockingconfiguration

1. Log in to the eRetail system and open the < System >-< Admin Tools >-< System Configuration >



2. Enter the < store configuration > page, first select the store that needs to be synchronized, check the > < data synchronization, set the synchronization frequency, and click <Next >



3. Goto the <DataSynchronizationConfiguration> page, select the database type, complete the connection string, use <SQL statement > script type, and fill in the < synchronization script >, use < stored procedure > script type, and fill in the < synchronization script> and < parameters >

Data synchronization config
(Global)

Data source
☒ Database
☐ File
☐ FTP file
☐ Customize

Database Type
My SQL

Connection string
data source=114.55.52.37;database=test;user id=root;pa
Fill me in the template!

Sync Script
select * from price_100050 limit 1000

Script type
☒ SQL statement
☐ Stored Procedures

Parameters
shop=000=false

Next

The database type	Connection string	Synchronization scripts	Remarks explain
SQL server	server=XXX; uid=XX; pwd=XXX; database=XXX;	select * from table name	server=database IP uid=UserID pwd=Password database=DBNAME
My SQL	data source=XXX; database=XXX; userid=XX; password=XX; pooling=false; charset=utf8; Connect Timeout=1000000	select * from table name	data source=database IP; database=DBNAME; userid=UserID; password=Password; pooling=false; (Whether to limit the number of connections in the connection pool and control the concurrency) charset=utf8; (Character Encoding) Connect Timeout=1000000; (Maximum connection timeout time)
Oracle	Data Source=XXX; User Id=XXX; Password=XXX	select * from table name	Data Source=DBNAME (If the database is not local, you need add IP adresss before dbname) ; User Id=UserID; Password=Password;
Sybase	Data Source=xxx; It'stial Catalog=XXX; User id=XX; Password=XXX	select * from table name	Data Source=IP; Initial Catalog=DBNAME; User id=UserID; Password=Password

Ex.: Sync Script: **select t.*, t.item_no, t.item_subno from EXAM t**

Parameter: branch_no=

000 item_no = %

item_subno = %

2.3.3. Textdockingconfiguration

The data files required for text docking support two modes: local upload and FTP upload, the local upload needs to fill in the absolute path of the file location, and the FTP upload needs to provide the IP address , user name, password, file path and other information. File formats: XLSX, XLS, CSV, TXT

File encoding format: UTF-8

Data synchronization config

(Global)

Data source

☐ Database

☒ File

☐ FTP file

☐ Customize

File Path

Please enter file path

File type

xlsx

First Behavior Listing

☒ Yes

☐ No

File Encoding Format

Please enter file encoding format

Delete source file

☒ Yes

☐ No

Backup Directory

Next

Data synchronization config

(Global)

Data source

☐ Database

☒ File

☒ FTP file

☐ Customize

FTP Address

Please enter FTP address

FTP username

Please enter the FTP username

FTP Password

Please enter the FTP password

FTP path

FTP path

File Path

Please enter file path

File type

xlsx

First Behavior Listing

☒ Yes

☐ No

File Encoding Format

Please enter file encoding format

Delete source file

☒ Yes

☐ No

Backup Directory

Next

2.3.4. Dynamicmodelconfiguration

After completing the database connection configuration/text connection configuration, enter the <dynamic model configuration > page, if the data synchronization is successful, you can successfully obtain the fields of the query table. Fill in the name that needs to be displayed in the display name > the table < display name, and if the display name is empty, the field is not synchronized.

System config → Store config → Data synchronization config → Dynamic model config → Model parameter config → Template config → System Activation

Dynamic model config

(Global)

When the display name is empty, out of sync with the field

	门店编号	string	Format	Readonly More Settings
	门店名称	string	Format	Readonly More Settings
	产地	String	Format	Readonly More Settings
	客服电话	string	Format	Readonly More Settings
	物价员	string	Format	Readonly More Settings
	商品编码	string	Format	☒ Readonly More Settings
	商品名	string	Format	Readonly More Settings
	别名	string	Format	Readonly More Settings
	条码	String	Format	Readonly More Settings
	规格	string	Format	Readonly More Settings
	单位	string	Format	Readonly More Settings
	来源	String	Format	Readonly More Settings
	售价	decimal	#0.00	Readonly More Settings
	进价	decimal	#0.00	Readonly More Settings

Next

Field types: string, dateTime, decimal, int

KCL Kuwait

www.kuwaitcorp.com

3. System integration interface

3.1. Label data push refresh

Purpose: Used for customers to directly push product data and refresh to specified labels. User data

will not be saved in the system background. **HTTP POST**

URL: http:// 192.168.1.92:5000/api/esl/tag/push

Content-Type: application/json

Request parameters:

Parameter name	type	illustrate
shopCode	string	Systemstorenumber
tagID	string	TagID
ap	string	SpecifiesthatthebasestationsendsNullbydefault
item	Object	Dataentity(correspondingtothefieldbelow).
GoodsCode	string	Productuniquecode
GoodsName	string	ProductName
Template	string	Templatename
Items	Array	Anarrayofdatadetails(correspondingfieldsused to bind to the template display).

Return format:

Parameter name	type	illustrate
code	Int	0:Success,Other>Error
message	String	Successorerrormessages
body	String	Thedefaultisempty

Example:

Request

```
{
  "shopCode": "0002",
  "tagID": "4F000001320A",
  "ap": "",
  "item": {
    "GoodsCode" : "123456",
    "GoodsName" : "Dry Red at Obartley Castle",
    "Template" : "SAL",
    "Items" : [
      "0002",
      "9999",
      "Shanghai City",
      "12315",
    ]
  }
}
```

```

    "Zhang San",
    "39880",
    "Obately Castle Dry Red",
    "Obately Castle Dry Red",
    "",
    "240209",
    "750ml",
    "Bottle",
    "France",
    "828",
    "88888",
    "0",
    "10",
    "Qualified"
  ]

```

Response

```

{"code": 0,
 "message": "success",
 "body": ""
}

```

3.2. Label query

Purpose: Used for customers to directly push product data and refresh to specified labels. User data will not be saved in the system background.

HTTP POST

URL: http:// 192.168.1.92:5000/api/esl/tag/query/{id}

Content-Type: application/json

Request parameters:

Parameter name	type	illustrate
id	string	Tag ID

Ret

urn format:

Parameter name	type	illustrate
code	Int	0: Success, Other: Error
message	String	Success or error messages
body	Thebject	Label entity
id	string	Label ID
apid	string	Finally, the base station is sent
shopCode	string	Store number
tagType	string int	Label type
tagStatus		Label status (0 - Not in use, 1 - normal, 2 - low battery, 3 - Communication, 5- Suspected loss, 6 final failure).
powerVal	int int	Electricity
temperature	int int	temperature
rsi	Array	Signal strength
token	int int	Send the token
goodsCode	string	Bind the product ID
totalSend	string	Total number of sends
errorCount		Total number of send failures
lastSendTime		Last sent time
lastRecvTime		Last return time

Example:**Request**

```
http://192.168.1.92:5000/api/esl/Tag/query/3600000148E3
```

Response

```
{
  "code": 0,
  "message": "success",
  "body": {
    "apid": "16",
    "shopCode": "0002",
    "tagType": "36",
    "tagStatus": 1,
    "powerVal": 30,
    "temperature": 24,
    "rssi": -89,
    "token": 215,
    "goodsCode": ["10824"],
    "totalSend": 322,
    "errorCount": 106,
    "lastSendTime": "2022-03-10 09:02:34",
    "lastRecvTime": "2022-03-10 09:02:56",
    "id": "3600000148E3",
  }
}
```

4. Customized interface docking

If you cannot use the above conventional docking methods or have special needs, eRetail3.supports customization and customized software interfaces on demand.

The demand side provides docking methods and data structures, which are customized and developed by our company after evaluating the requirements.



شركة كويت كارپوريشن ليميتد
KUWAIT CORPORATION LIMITED



Ali Sarfraz

Manager Operations & Sales



+965 50272775



ali@kuwaitcorp.com



<https://kuwaitcorp.com/>
<https://kcl.odocs.net/>



Certified Specialist

Software Architect, IT Services, Project Management, Cyber Security
ESL Labels, Cloud Hosting, E-Commerce, LMS System

We Deals In



ESL labels, Info Tags
Flexible Screens



Dynamic Displays
Ad Screens, Kiosks
Smart Boards

Enterprise Softwares



ERP, CRM, HRM, HMS
ESL, LMS



Integrations
K-Net, ERP, POS
Apple Pay, Google Pay

Our Global Associates



Noon Group International



www.aviox.net, UK (R): 16437921



Star Tower, Floor 17, Abdullah Al-Mubarak Street, Kuwait City.