Document Version	1.1
API version	1.0
e Retail Version	3.2.0
Release	2025-2026

D 35 eRetail 3.2 System Integration Technical Documentation

Limited Use Confidential

Table of contents

Limited Use Confidential

Limited Use Confidential

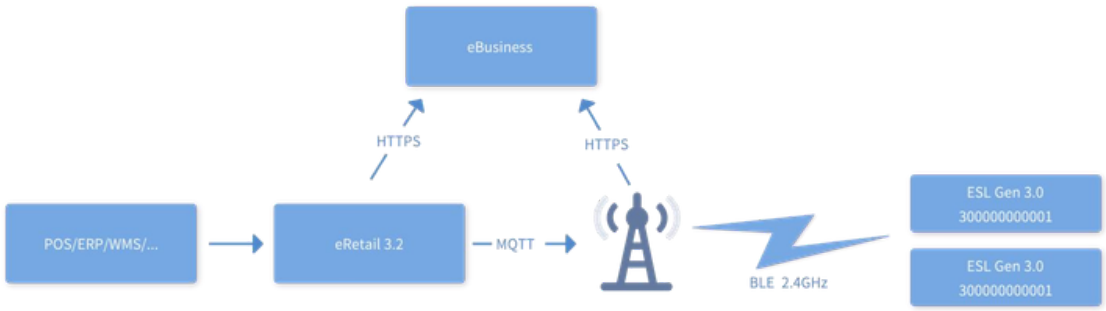
1 Brief

This document applies to projects based on .NET 6.0 or later.
This document applies to projects based on ESL Gen3 .
This document applies to eRetail- based... Projects using version 3.2 or higher.
This document applies to projects based on AP05.

1.1 System Overview

This document describes the integration methods for the eRetail 3.2 system. The eRetail 3.2 system supports three integration methods: API interface, file synchronization, and database synchronization. This document primarily focuses on the usage of the API interface. The API is a RESTful Web API , responsible for providing the interface for client systems to proactively push product data.

the eRetail 3.2 electronic shelf label solution, as divided above, is shown in the following view:



1.2 Comparison of System Integration Solutions

Generally, system integration includes three aspects: data management, template management, and device management. The table below compares the characteristics of various integration solutions to help project managers further evaluate system integration options:

dockingsolution	dockingdirection	Customerneedstopayattention	Customersdonotneed to pay attention
WebAPI: Pushproductdata	CustomerSystem->eRetail	ProductdatacontentTemplatecontent	Equipment Management
WebAPI: Push price tag->eRetail	CustomerSystem->eRetail	ProductdatacontentEquipment Templatecontent	Management

Limited Use Confidential

image			
DataSync: Provide data	productSystem	eRetail- >ClientProductdata contentEquipment	Datasource:File/DB Management /FTP

Therefore, it can be seen that:
1. This solution is suitable when the client has human resources for development, but only possesses product data;
2. This solution is suitable when the client has human resources for development and the client's system has the ability to display and manage product price tags;
3. This solution is suitable when the client lacks human resources for development but possesses product data;

2 Electronic price tag interface

API addresses are mentioned in the text , it is assumed that the deployment address of eRetail 3.2 is 192.168.1.92 and the port is 4000.

Note: HTTPS is recommended over HTTP.

For details, please refer to **Enforce HTTPS in ASP.NET Core | Microsoft Docs** .

2.1 Test Interface

Purpose: To detect whether the eRetail 3.2 API is accessible.

H TTP GET
URL : http://192.168.1.92:5000/api/hello

Return: " OK"
Note: This interface is for connectivity testing, in an informal scenario.

2.2 Handshake Interface

Purpose: This interface is used for authentication. All subsequent accesses to other interfaces depend on the data obtained from this interface.

HTTP POST
URL: http://192.168.1.92:5000/api/ login
Content -Type: application/ json

Request parameters :

Parameter type name		describe
userName	String	username
password	String	password

Return format:

Parameter type name		illustrate
code	Int	0: Success 4004 : Username or password is wrong. 6001 : Role does not exist. 4003 : User does not exist.
message	String	Successorerrormessage
body	String	Tokenforsubsequentsessions

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

Example:

Request

```
{
userName : " admin "
"password": "Pass99"
}
```

Response

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

2.3 Price tag input interface

Purpose: This interface is used for batch entry of price tags. It is a system integration-specific interface used to integrate eRetail 3.2 into the customer's system. If not necessary, the eRetail 3.2 backend management system or app can be used directly.

HTTP POST

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

Content -Type: application/ json

Request parameters :

Parameter type name		describe
shopCode String		System store number
s hopCode String Cst		Customer store number, which is empty by default. Note: If both the system store number and the customer store number exist, the customer store number will be matched first.
tagIdList	StringArray	Price tag ID list

Return format:

Parameter type name		illustrate
code	Int	0: Success Parameter validation 400 : Parameter error 3 001 : Invalid Tag ID
message	String	Successorerrormessage
body	String	Thedefaultvalueisempty.

Example: A shopCode or shopCodeCst must be passed.

Request

```
{
" shopCode ": "0001",
" tagIdList ": [
"440000010478",
      "440000010479"
]
}
```

Response

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

2.4 Price tag outbound interface

Purpose: This interface is used for price tag issuance.

Limited Use Confidential

HTTP POST
: http://192.168.1.92:5000/api/esl/tag/delete
Content -Type: application/ json

Request parameters :

Parameter name	type	describe
	StringArray	PricetagIDlist

Return format:

Parameter name	type	illustrate
code	Int	0:Success Parameter validation 400 : Parameter error 3 001 : Invalid Tag ID
message	String	Successorerrormessage
body	String	Thedefaultvalueisempty.

Example:
Request

```
[ " 440000010478 " , " 440000010479 " ]
```

Response

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

2.5 Price tag binding interface

Purpose: This interface is used to bind price tag IDs to product numbers. This is a system integration-specific interface used to integrate eRetail 3.2 into the customer's system. If not necessary, the eRetail 3.2 backend management system or app can be used directly.

HTTP POST
URL : http : // 192.168.1.92:5000/api/esl/tag/Binding
Content -Type: application/ json

Limited Use Confidential

Request parameters :

Parameter name	type	describe
shopCode	String	System store number
shopCodeCst	String	Customer store number, which is empty by default. Note: If both the system store number and the customer store number exist, the customer store number will be matched first.
binds	Array	ArrayString	Binding array
tagID	ArrayString	Price tag ID
goodsCode	ArrayString	Product Number

Return format:

Parameter name	type	illustrate
code	Int	0: Success Parameter validation 400 : Parameter error 1 001 : Tag is null (The price tag is empty) 3 001 : Invalid Tag ID 3 004 : Tag does not exist . 7 001 : Goods does not exist .
message	String	Successorerrormessage
body	String	Thedefaultvalueisempty.

Example:

Request

```
{
    " shopCode ": "0001",
    " s hopCode Cst ": "",
    "binds": [{
        " tagID ": "40000000EA40",
        " goodsCode ": "123456"
    }]
}
```

Response

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


Limited Use Confidential

2.6 Price tag unbinding interface

Purpose: This interface is used to unbind the price tag ID from the product number. This is a system integration-specific interface used to integrate eRetail 3.2 into the customer's system. If not necessary, the eRetail 3.2 backend management system or app can be used directly.

HTTP POST
URL : http://192.168.1.92:5000/api/esl/tag/unbind
Content -Type: application/ json

Request parameters :

Parameter name	type	describe
shopCode	String	Systemstorenumber
shopCodeCst	String	Customerstorenumber,whichisemptybydefault. Note: If both the system store number and the customer store number exist, the customer store number will be matched first.
unbindType	Int	Default pass 1
idList	StringArray	Price tag ID

Return format:

Parameter name	type	illustrate
code	Int	0: Success Parameter validation 400: Parameter error 1001: Price tag ID is empty 3001: Tag format error 3004: Tag does not exist 1001: Unbinding failed
message	String	Success or error message
body	String	The default value is empty.

Example:
Request

```
{
   " shopCode ": "0001",
   " s hopCode Cst ": "",
"ap": "",
```


Limited Use Confidential

```
" idList ": [
"40000000EA40",
"40000000EA39"
],
" unbindType ": 1
}
```


Response


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


2.7 Price tag combination binding interface

Purpose: This interface is used to combine and bind a price tag ID with multiple product numbers, meaning that multiple product information can be displayed on a single price tag. This is a system integration-specific interface used to integrate eRetail 3.2 into the customer's system. If not necessary, the eRetail 3.2 backend management system or App can be used directly.

HTTP POST
URL : http://192.168.1.92:5000/api/esl/tag/groupBinding
Content -Type: application/ json

Request parameters :

Parameter name	type	describe
Parameter name	type	describe
shopCode	String	Systemstorenumber,canbeempty
shopCodeCst	String	Customerstorenumber,whichisemptybydefault. Note: If both the system store number and the customer store number exist, the customer store number will be matched first.
ap	String	DesignatedbasestationID
template	String	Templatetype(layoutscheme),suchas2x2,2x3, etc. PricetagID
tagID	String	Storenumber+productnumber
goodsCode	ArrayString	

Limited Use Confidential

Return format:

Parameter name	type	illustrate
code	Int	0: Success Parameter validation 400: Parameter error 3 001 : Invalid Tag ID 3004 : Tag does not exist. 700 1 : Goods do not exist . Error 8001 : Template does not exist . 1001 : GroupBind Fail
message	String	Success or error message The default value is empty.
body	String	

Example:

Request

```
{
" shopCode ": "0001",
   " s hopCode Cst ": "",
"ap": "",
"template": "1x2",
"              tagID              ":
"40000000EA40",           "
goodsCode ": [
"123456",
"123457"
]
}
```

Response

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

2.8 Price tag flashing light interface

Purpose: This interface is used to notify the price tag flashing light. This is a system integration-specific interface for integrating eRetail 3.2 into the customer's system. If not necessary, the eRetail 3.2 backend management system or app can be used directly.

HTTP POST

URL : http : // 192.168.1.92:5000/api/esl/tag/light
Content -Type: application/ json

Request parameters :

Parameter name	type	describe
shopCode	String	Systemstorenumber
shopCodeCst	String	Customerstorenumber,whichisemptybydefault. Note: If both the system store number and the customer store number exist, the customer store number will be matched first. Flashcolors(R-Red,G-Green,B-Bright)
rgb	string	String format: such as " R " represents a red light. " RG " indicates a red/green light. " RGB " means all red, green, and blue colors are lit.
times	int	Flashduration(seconds)
idList	Array	Pricetagorproductarray
ledType	Int	Flashingtype(1:specifypricetag,0:specify product)

Return format:

Parameter name	type	illustrate
code	Int	0: Success Parameter validation 400: Parameter error 1 001 : List data is null (The list data is empty) 1 001 : LED Fail
message	String	Successorerrormessage
body	String	Thedefaultvalueisempty.

Example:

Request

```
{
" shopCode ": "0001",
   "shopCodeCst": "",
   "rgb": "RGB", //R-Red,G-Green,B-Blue
   "times": 10,
   "idList": [
      "40000000EA40" //ledType 1-价签ID， 0->商品ID
   ],
```


Limited Use Confidential

```
  "ledType": 1
}
```

Response

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

2.9 Price tag data push refresh

Purpose: Used for customers to directly push product data and refresh to a specified price tag. The system backend does not guarantee the security of user data.

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,canbeempty
shopCodeCst	String	Customerstorenumber(canbeempty)
tagID	string	PricetagID
ap	string	Thespecifiedbasestationsendsadefaultempty message.
item	Object	Dataentity(correspondingtothefieldsbelow)
goodsCode	string	Productuniquecode
goodsName	string	ProductName
template	string	TemplateName
items	Array(characterDatadetailsarray(correspondingfieldsareused array)	tobindtothetemplatedisplay)

Return format:

Parameter name	type	illustrate
code	Int	0: Success Parameter validation 400 : Parameter error

Limited Use Confidential

		3 001 : Invalid Tag ID 3004 : Tag does not exist. 2 006 : This code does not exist . Error 8001 : Template does not exist . 1001 : Push tag data fail (failed to push price tag data)
message	String	Successorerrormessage
body	String	Thedefaultvalueisempty.

Example:

Request

```
{
   " shopCode ": "",
   " shopCodeCst ": "0002",
" tagID ": "4F000001320A",
"ap": "",
"item": {
      GoodsCode : 123456
      " GoodsName ": "Château Aubertley Red Wine"
      "Template " : "SAL",
      "Items ": [
            "0002",
            "9999",
            Shanghai
            "12315",
            "Zhang San",
            "39880",
            "Château O'Bartley Dry Red Wine",
            "Château O'Bartley Dry Red Wine",
            "",
            "240209",
            750ml
            "bottle",
            "France",
            "828",
            "88888",
            "0",
            "10",
            "qualified"

      ]
}
}
```


Limited Use Confidential

Response


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


2.10 Pricetagdatabatchpushrefresh

Purpose: Used for customers to directly push product data and refresh to a specified price tag. The system backend does not guarantee the security of user data.

HTTP POST
: http : // 192.168.1.92:5000/api/esl/tag/ pushList
Content -Type: application/ json

Request parameters : Push in array format

Parameter name	type	illustrate
shopCode	String	Systemstorenumber,canbeempty
shopCodeCst	String	Customerstorenumber(canbeempty)
tagID	string	PricetagID
goodsCode	string	ChangetheproductID-goodsIdtogoodsCode
goodsName	string	ProductName
template	string	TemplateName
items	StringArray	Datadetailsarray(correspondingfieldsareused to bind to the template display)

Return format:

Parameter name	type	illustrate
code	Int	0: success Parameter validation 400 : Parameter error 3001 : Invalid Tag ID {0} 2006 : {0} does not exist. 2007 : Data {0} is empty. 3004 : Tag {0} does not exist.

Limited Use Confidential

		8001 : Template does not exist {0},{1},{2} mistake 1001 : Push tag data failed (Price tag data push failed)
message	String	Success or error message
body	String	The default value is empty.

Example:
Request

```
{
   "shopCode": "0001",
   "shopCodeCst": "",
   "dataItems": [
      {
         "tagID": "4F000001320A",
         "template": "REG",
         "goodsCode": "123456",
         "goodsName": "xxx",
         "items": [
      "123456",
            "商品名称",
            "上海市",
            "12315",
            "Zhang San",
            "39880",
            "Château O'Bartley Dry Red Wine",
            "Château O'Bartley Dry Red Wine",
            "",
            "240209",
            750ml
            "bottle",
            "France",
            "828",
            "88888",
            "0",
            "10",
            "qualified"

]
},
...multiple price tags and product data
]
```


Limited Use Confidential

```
}
```

Response

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

2.11 Pricetagimagedatapush

Purpose: To push image data directly to the system and send it to the price tag (note that the image size should ideally match the price tag size).

HTTP POST
URL : http : // 192.168.1.92:5000/api/esl/tag/Image
Content -Type: application/ json

Request parameters :

Parameter name	type	describe
shopCode	String	Systemstorenumber,canbeempty
shopCodeCst	String	Customerstorenumber(canbeempty)
tagID	String	PricetagID
base64Data	string	Imagebase64string Supported image formats: bmp, jpeg , jpg, png
isDither	Bool	Doestheimageneedtobeshaken? The default jitter algorithm used is the Floyd-Steinberg diffusion jitter algorithm.
batchID	String	Can be empty, unique ID for the task
backUrl	String	Empty

Return format:

Parameter name	type	illustrate
code	Int	0: Success Parameter validation 400 : Parameter error 3001 : Invalid Tag ID {0} 3004 : Tag {0} does not exist.

Limited Use Confidential

		mistake 1001 : Preview Fail 1001 : PushImage Error (Failed to push image)
message	String	Successorerrormessage
body	String	Thedefaultvalueisempty.

Example:

Request

```
{
   "shopCode": "",
   "shopCodeCst": "",
   "tagID": "400001234567",
   "base64Data": "image base64 string",
   "batchID": "",
   "isDither": true
}
```

Response

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

HTTP POST

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

Content -Type: application/ json

Request parameters :

Parameter name	type	describe
shopCode	String	Systemstorenumber
shopCodeCst String		Customerstorenumber(canbeempty)
batchID	String	Canbeempty,uniqueIDforthetask
items	ObjectArray	Pricetagimagelist
tagID	String string	PricetagID
base64Data	Bool String	Imagebase64string
isDither		Shouldtheimagebejittered?(true/false)
backUrl		Feedbackaddress(canbeempty)

Return format:

Parameter name	type	illustrate
code	Int	0:Success

		Parameter validation 400 : Parameter error 3001 : Invalid Tag ID {0} 3004 : Tag {0} does not exist. mistake 1001 : Preview Fail 1001 : PushImage Error (Failed to push image)
message	String	Successorerrormessage Thedefaultvalueisempty.
body	String	

Example:

Request

```
{
" shopCode ": "0001",
   "shopCodeCst": "",
   "batchID": "",
   "items": [
      {
         "tagID": "400000000001",
         "base64Data":
"iVBORw0KGgoAAAANSUhEUgAAAZAAAAEsCAYAAADtt+XCAAAAAXNSR0IArs4c6Q
AAxxxx.....",
         "isDither": true,
         "backUrl": ""
      }
   ]
}
```

Response

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

2.12 Pricetaginquiry

Purpose: To query detailed information about a specified price tag.

HTTP GET

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

Content -Type: application/ json

Limited Use Confidential

Request parameters :

Parameter name	type	illustrate
id	string	Price tag ID

Return format:

Parameter name	type	illustrate
code	Int	0: Success
message	String	Success or error message
body	Objective	Price tag physical
id	string	Price tag ID
defaultAP	string	Finally, the base station is sent.
shopCode	string	Store number
shopCodeCst string		Customer store number
tagType	string	Price tag type
tagStatus	int	Price tag status (0 - Unused, 1 - Normal, 2 - Low battery, 3 - Communicating, 5 - Suspected loss, 6 - Final failure)
powerVal	int	Battery level (returns an integer, e.g., 31 represents 3.1V) Commonly used conversion rules for battery range: Battery level >= 30% -> 100% 28 <= Battery level < 30 -> 90% 27 <= battery level < 29 -> 80% 26 <= battery level < 28 -> 60% 25 <= battery level < 27 -> 60% 24 <= battery level < 26 -> 20% 23 <= battery level < 25 -> 10% Battery level <23 ms -> 0%
temperature	int	temperature
rssi	int	signal strength
token	int	Send TOKEN
goodsCode	Array	Binding Product ID
totalSend	int	Total number of sends
errorCount	int	Total number of failed transmissions
lastSendTime	s tring	Last sent time
lastRecvTime	s tring	Last return time

Example:
Request

Limited Use Confidential

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

Response


```
{
"code": 0,
"message": "success",
"body": {
      "id": "40000000E4A2",
      " shopCode ": "0001",
      " shopCodeCst ": "A050",
      " goodsCode ": ["100000"],
      "tagType": "40",
      "tagStatus": 1,
      "powerVal": 29,
      "temperature": 24,
      "rssi": -61,
      "totalSend": 8,
      "errorCount": 0,
      "lastSendTime": "2024-06-07 16:35:50",
      "lastRecvTime": "2024-06-07 16:38:10",
      " defaultAP ": " 01 " ,
      " token ": 0

}
}
```


2.13 Pricetaglistsearch

Purpose: To look up a price tag list.

HTTP POST

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

Content -Type: application/ json

Request parameters :

Parameternametype		Empty	illustrate
PageIndex	int int	no no	pagenumber
PageSize	string	yes	Pages
sort	string	yes	SortbyfieldssuchaslastSendTime
desc	string	yes	Sortingrulessuchas:desc/asc
id	string	yes	PricetagID
shopCode	string	yes	SystemStores
goodsCode			Bindproductcode

Limited Use Confidential

tagType	string	ye	Price tag type, such as: 36
tagStatus	string	s ye s	Price tag status: 0 - Initialization; 1 - Idle; 2 - Low battery; 3 - Working; 6 - Blacklist (attempt to refresh, failed due to reaching limit) ; 7 - Heartbeat.
powerValBegin	int	no	Battery range query minimum value -1 ignore query
powerValEnd	int	no	Battery range query maximum value - 1 ignore query
bindStatus	int	no	Is it bound? 0 - All 1 - Bound 2 - Not bound

Request:

```
{
    " PageIndex " : 1 ,
    " PageSize " : 20 ,
    "sort" : "" ,
    "desc" : " " ,
    "id" : "" ,
    " goodsCode " : "" ,
    " shopCode " : "" ,
    " tagType " : "" ,
    " tagStatus " : "" ,
    " taskStatus " : "" ,
    " powerValBegin " : -1 , minimum power range -1- ignore
    " powerValEnd " : -1 , maximum battery range -1- ignore
    " bindStatus " : 0 , //Whether to bind 0-All 1-Bound 2-Not bound

}
```

Response:

```
{
    "code" : 0 ,
    "message" : "success" ,
    "body" : {
        " totalCount " : 244 ,
        "items" : [
            {
                " goodsCodeList " : "00011234565" , // Bind product ID
                " shopCodeCst " : "9998" , // Customer store
                " goodsCode " : "1234565" , // Bind product code
                "id" : "820000AB1F44" , //Price tag ID
                " isDeleted " : false ,
                " createdBy " : "Admin "
```


Limited Use Confidential

```
                    " createdTime " : "2025-12-04 18:06:37 "
                    "lastUpdatedBy": "Admin"
                    " lastUpdatedTime " : " 2026-04-07
18:26:09"                    " shopCode " : "0001" , //System Store
                    " tagType " : "82" , //Price tag type
                    " tagStatus " : 1 , //Price tag status
                    "token" : 1 , // Price tag refresh token
                    " powerVal " : 30 , // Battery level
                    "temperature" : 24 , //Temperature
                    " rssi " : -63 , //signal
                    " totalSend " : 60 , // Number of items sent
                    " errorCount " : 5 , // Number of failures
                    " errorCountTemp " : 66 , // Total number of tasks
                    " lastSendTime " : "2026-04-07 18:25:53" , //Last sent time
                      " lastRecvTime " : "2026-04-07 18:26:08" , //Last received
time
                    "version" : "38" , // Firmware version
                    " areaID " : null , //area number
                    " templateID " : "" , // Combined template ID
                    " defaultAP " : "00DL" , //Last sent base station number
                    "best " 0 ,
                    " lowCount " : 0 ,
                    " lastHeartbeatTime " : "2026-03-10 12:05:03" , //Heartbeat
Time
                    " heartbeatCount " : 241 , // Heartbeat count
                    "orientation " 0 , // Installation direction
                    " aps " : null
                }
            ]
        }
}
```


2.14 Basestationfeedbackcallbackinterface

Purpose: When a base station goes online, offline messages are proactively pushed to the customer's system.

Note: This interface needs to be provided by the customer. After the system configuration enables feedback, this interface will be called back when the base station status changes.

use the following URL format: http://xxx.xxx.xxx.xxx/apheart

HTTP POST
URL: http:// client interface address /apheart
Content -Type: application/ json

Request parameters :

Parameter name	type	illustrate
	ObjectArray	Returnsamessagearrayobject
ShopCode	string string	Systemstorenumber
MAC	string Int	BasestationMAC
APID		BaseStationID
ApStatus		State0Initialization1Online2Heartbeat3 Offline
LastOnlineTime	DateTime	Lastupdated

Return format:

Parameter name	type	illustrate
code	Int	0:Success
message	String	Successorerrormessage
body	Objective	ReturnEntity

Example : http://127.0.0.1/api/apheart

Request

```
[
{
ShopCode : 9998
"MAC": "111111111",
"APID": "0 001 ",
"ApStatus":1
"     LastOnlineTime     ":     "2020-05-01
11:11:59"
} ,
{
ShopCode : 9998
"MAC": "11111111 2 ",
"APID": "0 002 ",
"ApStatus":1
"     LastOnlineTime     ":     "2020-05-01
11:11:59" }

]
```

Response

Limited Use Confidential

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

2.15 Pricetagfeedbackcallbackinterface

Purpose: To proactively push notifications to the customer's system whether the price tag refresh is successful or not.

Note: This interface needs to be provided by the customer. Once feedback is enabled in the system configuration, the system will call back to this interface after receiving feedback from the price tag.

the following URL format: http://xxx.xxx.xxx.xxx/callback

HTTP POST

URL: http:// client interface address /callback

Content -Type: application/ json

Request parameters :

Parameter name	type	illustrate
	ObjectArray	Returnsamessagearrayobject
ShopCode	string string	Systemstorenumber
TagID	string string	PricetagID
TagType	string string	Pricetagtype
BatchID	string string	Sendbatchnumber
APID	Int	TransmitbasestationID
GoodsCode		ProductID
GoodsNam		ProductName
e		TemplateName
TempType TaskStatus		Taskstatus:0Initialization,1Communicationin progress, 2 Success , 4 Failure, 5 Timeout , 8 Overwrite
TaskType	Int	Tasktype
Rssi	Int	signalstrength
PowerVal	Double	Battery
Temperature	Double	temperature
LastRecvTime	D ateTime	Finalfeedbacktime

Return format:

Limited Use Confidential

Parameter name	type	illustrate
code	Int	0:Success,Other:Error
message	String	Successorerrormessage
body	Objective	ReturnEntity

Request

```
[
{
ShopCode : 9998
"TagID":       "401234567890",
TagType : "40" ,
"              BatchID              ":
"1323dwafs11343",
   "APID": "0001",
   "GoodsCode": "123456",
   "GoodsName": "test",
   "TempType": "REG",
   "TaskStatus": 1,
   "TaskType": 1,
   "Rssi": -56,
   "PowerVal": 32,
   "Temperature": 32,
   "LastRecvTime": "2020-05-01 11:11:59"
},
{
   "ShopCode": "9998",
   "TagID": "401234567891",
   "TagType": "40",
   "BatchID": "1323dwafs11343",
   "APID": "0001",
   "GoodsCode": "123456",
   "GoodsName": "test",
   "TempType": "REG",
" TaskStatus ": 1,
TaskType : 1,
Rssi : -56 ,
" PowerVal ": 32,
"Temperature": 32,
" LastRecvTime ": "2020-05-01 11:11:59"
}

]
```


Limited Use Confidential

Response

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

2.16 Price tag refresh (specific price tags or store-wide)

Purpose: This interface is used to refresh the price tag screen without changing the displayed content.

HTTP POST
URL : http : // 192.168.1.92:5000/api/esl/tag/Refresh
Content -Type: application/ json

Request parameters :

Parameter name	type	Required	describe
shopCode	String	yes	Systemstorenumber
refreshType	Int	yes	RefreshType1:Template,2:PriceTagType,3: Specific Store, 4: Price Tag ID List
refreshName	String	no	WhenrefreshType=1,itrepresentsthetemplate name; when refreshType = 2, it represents the price tag type; otherwise, it is empty.
tags	String Array	no	WhenrefreshType=4,anarrayofpricetagIDsis specified. Other fields are empty.

Return format:

Parameter name	type	illustrate
code	Int	0: Success Others were failures.
message	String	Success or error message
body	String	The default value is empty.

示例：
Request

Limited Use Confidential

```
{
   "shopCode": "0001",
   "refreshType": 4,
   "refreshName": "",
   "tags": [
      "401234567890",
      "401234567891"
   ]
}
```

Response

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

2.17 基站查询

Purpose: To query the status of a specified store's base station.

HTTP GET
URL : http : // 192.168.1.92:5000/api/esl/Ap/queryShopAP/{ShopCode }
Content -Type: application/ json
Request parameters :

Parameter name	type	Requiredd	escribe
shopCode	String	yes	Systemstorenumber

Return format:

Parametername	type	illustrate
code	Int	0:Success Others were failures.
message	String	Successorerrormessage
body	ObjectArray	Basestationlist
id	string	PrimaryKeyID
shopCode	string	Systemstorenumber
apid	string int	BasestationID
apStatus		Basestationstatus: 0-init, 1-Online, 2-heartbeart, 3-Offline, 4-Working, 5-error, 6-blacklist, 9-other

Limited Use Confidential

apType	int string	Basestationtype
IP	string	Ip
Mac	string	Mac
hardware	string int	Firmwareversion
firmware	DateTime	Moduleversion
offlineCount	DateTime	Offlinecount
lastOnlineTime		Lastonlinetime
lastOfflineTime		Lastofflinetime
lastHeartbeatTime	DateTime	Lastheartbeattime
remark	string	备注

示例：

Request

```
http:// 192.168.1.92:5000/api/esl/Ap/queryShopAP/0001
```

Response

```
{
  "code": 0,
  "message": "success",
  "body": [
    {
      "id": "U007",
      "shopCode": "0001",
      "apid": "U007",
      "apStatus": 3,
      "apType": 5,
      "ip": "",
      "mac": "60:48:9C:40:34:80",
      "hardware": "1.0.1.0",
      "firmware": "303135_303",
      "offlineCount": 2,
      "lastOnlineTime": "2025-09-23 17:12:00",
      "lastOfflineTime": "2025-09-23 17:13:20",
      "lastHeartbeatTime": "2025-09-23 17:12:59",
"remark":
"0001" }
]
}
```

2.18 Template Query

Purpose: To look up price tag templates

HTTP GET

URL: http:// 192.168.1.92:5000/ api/Template/queryTemplate?
shopCode=&tagType= templateType=-1
Content -Type: application/ json

Request parameters :

Parameter name	type	Required	escribe
shopCode	String	no	Systemstorenumber,nullvalue-Queryallstores
tagType	String	no	Querytemplatesforaspecifiedpricetagtype
templateType	String	no	Template Type - 1 - All, 0 - ESL, 1 - LCD Template, 2 - Custom Template

Return format:

Parameter name	type	illustrate
code	Int	0: Success Others were failures.
message	String	Success or error message
body	ObjectArray	Template List
templateName	s tring	Template Name
tagType	s tring	Price tag type
shopCode	s tring int	Store number, empty indicates global access.
templateType		Template Type 0 - ESL 1 - LCD Template 2 - Custom Template
templateStatus	in	Template status
blocks	t in t	The number of combined template blocks is greater than 0, indicating a combined template.
preview	s tring	Preview address
width	int	Width
height	int	high
goodsCount	int	Number of products in the combined template Marketing space
openWidth	in	Marketing position
openHeight	t	The open area supports the following resource
openResType	in t in	types: 0 image, 1 video, 2 image+video. primary key
id	t nt	

Example:

Limited Use Confidential

Request

```
http:// 192.168.1.92:5000/ api/Template/queryTemplate? shopCode=0001&tagType=
templateType=-1
```

Response


```
{
"code": 0,
"message": "success",
   "body": [
      {
         "templateName": "REG",
         "tagType": "81",
         "shopCode": "",
         "templateType": 0,
         "templateStatus": 0,
         "blocks": 0,
         "preview":
"api/template/preview_template?shopCode=&tagType=81&name=REG&timeStamp
=1758524920398",
         "width": 250,
         "height": 122,
" goodsCount ": 0, "
openWidth ": 0,
" openHeight ": 0,
" openResType ": 1,
"id": 2,
}
]
}
```


3 Store Management

3.1 Store creation

Purpose: To create a store

HTTP POST
URL : http: // 192.168.1.92:5000/api/shop/add
Content -Type: application/ json

Request parameters :

Parameter name	type	Requiredd	escribe
shopCode	String	yes	Systemstorenumber
shopCodeCstInt		yes	Customerstorenumber
shopName	String	yes	StoreName
shopAddress	String Array	no	Storeaddress
sysOrgId	Int	yes	Default1-rootnode

Return format:

Parameter name	type	illustrate
code	Int	0:Success Others were failures.
message	String	Successorerrormessage
body	String	Thedefaultvalueisempty.

Example:

Request

```
{
" shopCode ": "0001",
" shopCodeCst ": "A001",
" shopName ": "Test Store"
shopAddress : Suzhou
   " sysOrgId ": 1
}
```

Response

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

3.2 Store deletion

Purpose: To delete a store. Deleting a store will result in the loss of existing data, so please use with caution.

Limited Use Confidential

HTTP GET
URL: http: // 192.168.1.92:5000/api/shop/delete/{id }
Content -Type: application/ json

Request parameters :

Parameter name	type	Requiredd	escribe
id	int	yes	ThestoreprimarykeyIDcanbefoundinsection 3.3 and can be obtained through store lookup.

Return format:

Parameter name	type	illustrate
code	Int	0:Success Others were failures.
message	String	Successorerrormessage
body	String	Thedefaultvalueisempty.

Example : http://192.168.1.92:5000/api/shop/delete/1
Request

Response

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

3.3 Store search

Purpose: To query the relationships of all stores under the current user.
HTTP GET
URL: http : // 192.168.1.92:5000/api/shop/ queryShopListByUser
Content -Type: application/ json

Request parameters :

Parameter name	type	describe

Return format:

Limited Use Confidential

Parameter name	type	illustrate
code	Int	0:Success
message	String	Successorerrormessage
body	ObjectArray	Storelistentities
shopCode	String	Systemstorenumber
shopCodeCst	String	Customerstorenumber
shopName	String	StoreName
shopAddress	String	address
shopStatus	String	Status:0Normal,1Off
Id	Int	StoreprimarykeyID

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

Example:

Request

```
```

Response

```
{
"code": 0,
"message": "success",
"body": [
{
" shopCode ": "0001",
" shopCodeCst ": "A050",
shopName : "23 Test Store" ,
        "shopAddress": "sz",
        "shopStatus": 0,
        "id": 22
    },
    {
        "shopCode": "0002",
        "shopCodeCst": "0002",
        "shopName": "0002",
        "shopAddress": "0002",
        "shopStatus": 0,
        "id": 29
    }
]
}
```

4 Digital Signage Interface

4.1 Signage list query interface

Purpose: This interface is used for querying the signage list. It is a system integration-specific interface for integrating eRetail 3.2 into the customer's system. If not necessary, the eRetail 3.2 backend management system or app can be used directly.

HTTP POST
URL : http://192.168.1.92:5000/api/tft/tft/queryList
Content -Type: application/ json
Request parameters :

Parameter name	type	describe
TFTId	String	DeviceID
shopCode	String	Storenumber
shopCode Cst	String	Customerstorenumber
pageIndex	Int	PageIndex
pageSize	Int	Pagesize

Return format:

Parametername	type	illustrate
code	Int	0:Success Parameter validation 400 : Parameter error
message	String	Successorerrormessage
Body	String Int	Thedefaultvalueisempty.
totalCount	Array String	Numberofqueries
items	String	Queryresultslist
tftId	String	DeviceID
custId	String	CustomerID
instId	String	EntityID
tftType	DateTime	Equipmenttype
tftVersion	DateTime	Deviceversionnumber
lastHeartbeatTime		Lastheartbeattime
lastOfflineTime		Lastofflinetime

Limited Use Confidential

lastOnlineTime	DateTime	Last online time
lastUpdatedTime	DateTime	Last updated
createdTime	DateTime	Creation time
tftToken	String	Communication token
videoTaskId	String	Task ID
shopCode	String	Store number
shopCode Cst	String	Customer store number
goods	Array	Product ID (array)
templateName	String	Template Name
tftBindRes	String	Device-bound resources
areaId	Int	Region ID
ssid	String	Wifi SSID
ssidpwd	String	Wi-Fi SSID Password
serverUrl	String	Service Address
certUrl	String	Certificate address
certMD5	String	Certificate MD5 verification
status	Int	Device status indicators: 0 Offline, 1 Online, 2 Downloading, 3 Not activated
tftName	String	Equipment Name
firmware	String	Firmware version
lastVideoTaskId	String	Final Task ID
firstDownloadTime	DateTime	First download time
lastDownloadTime	DateTime	First download time
lastConfirmTime	DateTime	Final confirmation time
downLoadCount	Int	Download count
confirmCount	Int	Number of confirmations
heartBeatCount	Int	Heart rate statistics
taskCount	Int	Total number of tasks
lastSpanTime	DateTime	Time taken for the last task (in seconds)
areaName	String	Area Name
statusStr	String	Device status name
goodsStr	String	Product Names (array)
lastHeartbeatTimeStr	String	Last heartbeat time (MM-dd HH:mm)
lastDownloadTimeStr	String	Last download time (MM-dd HH:mm) Number of queries
totalCount	Int	

Example:
Request

```
{
" TFTId ": "",
" shopCode ": "0001",
```


Limited Use Confidential

```
" pageIndex ": 1,
pageSize : 30
}
```

Response


```
{
"code": 0,
"message": "success",
"body": {
        "totalCount": 149,
        "items": [
        {
                "tftId": "C49C14844658",
                "custId": "ETAG",
                "instId": "0001",
                "tftType": "5C",
                "tftVersion": "61",
                "lastHeartbeatTime": "2022-09-06 13:17:09",
                "lastOfflineTime": "1900-01-01 00:00:00",
                "lastOnlineTime": "1900-01-01 00:00:00",
                "lastUpdatedTime": "2022-09-06 13:17:09",
                "createdTime": "2022-09-02 19:24:25",
                "tftToken": "",
                "videoTaskId": "b88d7987-4ad3-462a-9aa7-
                f917f7267d82", "shopCode": "0001",
                "goods": [
                    "1"
                ],
                "templateName": "显示区域视频",
                "tftBindRes": [],
                "areaId": 7,
                "ssid": "",
                "ssidpwd": "",
                "serverUrl": "",
                "certUrl": "",
                "certMD5": "",
                "status": 1,
                "tftName": "",
                "firmware": "",
                "lastVideoTaskId": "b88d7987-4ad3-462a-9aa7-f917f7267d82",
                "firstDownloadTime": "2022-09-06 13:16:10",
                "lastDownloadTime": "2022-09-06 13:16:10",
                "lastConfirmTime": "2022-09-06 13:17:02",
```

```
                    "downLoadCount": 152,
                    "confirmCount": 152,
                    "heartBeatCount": 257,
                    "taskCount": 152,
                    "lastSpanTime": 52,
                    "areaName": "视频",
                    "statusStr": "",
                    "goodsStr": "",
"   lastHeartbeatTimeStr   ":
"",  "   lastDownloadTimeStr
": "" }
]
}
}
```


4.2 Signage Query Interface

Purpose: This interface is used for signage lookup. It is a system integration-specific interface for integrating eRetail 3.2 into the customer's system. If not necessary, the eRetail 3.2 backend management system or app can be used directly.

HTTP GET

URL: http:// 192.168.1.92:5000/ api / tft / tft?tftId =C49C14844658

Content -Type: application/ json

Request parameters :

Parameter name	type	describe
TFTId	String	DeviceID

Return format:

Parameter name	type	illustrate
code	Int	0: Success 4.00 : Parameter error
message	String	Success or error message
body	String	The default value is empty.

Example:

Request

```
TFTId = C49C14844658
```

Response

Limited Use Confidential

```
{
"code": 0,
"message": "success",
"body": {
" tftId ": "C49C14844658",
" custId ": "ETAG",
" instId ": "0001",
" tftType ": "5C",
            "tftVersion": "61",
            "lastHeartbeatTime":              "2022-09-06
            13:17:09",   "lastOfflineTime":   "1900-01-01
            00:00:00",   "lastOnlineTime":    "1900-01-01
            00:00:00",  "lastUpdatedTime":  "2022-09-06
            13:17:09",     "createdTime":     "2022-09-02
            19:24:25", "tftToken": "",
" videoTaskId ": "b88d7987-4ad3-462a-9aa7-f917f7267d82",
" shopCode ": "0001",
            " shopCode Cst ": " A050 ",
"goods": [
"00011"
],
" templateName ": "Display Area Video",
" tftBindRes ": [],
" areaId ": 7,
            "ssid": "",
            "ssidpwd": "",
            "serverUrl": "",
            "certUrl": "",
            "certMD5": "",
            "status": 1,
            "tftName": "",
            "firmware": "",
            "lastVideoTaskId": "b88d7987-4ad3-462a-9aa7-f917f7267d82",
            "firstDownloadTime": "2022-09-06 13:16:10",
            "lastDownloadTime": "2022-09-06 13:16:10",
            "lastConfirmTime": "2022-09-06 13:17:02",
            "downLoadCount": 152,
            "confirmCount": 152,
            "heartBeatCount": 257,
            "taskCount": 152,
            "lastSpanTime": 52,
" areaName ": "video",
```

" statusStr ": "",
" goodsStr ": "",
" lastHeartbeatTimeStr ":
"", " lastDownloadTimeStr
": "" }
}

4.3 Signage Preview Interface

Purpose: This interface is used for signage preview. It is a system integration-specific interface for integrating eRetail 3.2 into the customer's system. Alternatively, the eRetail 3.2 backend management system or app can be used directly.

HTTP GET

URL: http:// 192.168.1.92:5000/ api / tft / tft / PreviewTft /C49C14844658

Content -Type: application/ json

Request parameters :

Parameter name	type	describe
TFTId	String	DeviceID

Return format:

Parameter name	type	illustrate
code	Int	0: Success 4.04 : Resource does not exist 1 001 : The device has not been bound to a template .
message	String	Success or error message
body	String	The default value is empty.

Example:

Request

```
TFTId = C49C14844658
```

Response

{
"code": 0,
"message": "success",
"body": [
{
"id": 0,

" createdBy ": null,
" createdTime ": "0001-01-01
00:00:00", "mac": null,
" displayId ": 0,
}
]
}

4.4 Signage Template Name Query Interface

Purpose: This interface is used to query the name of signage templates. This is a system integration-specific interface used to integrate eRetail 3.2 into the customer's system. If not necessary, the eRetail 3.2 backend management system or App can be used directly.

HTTP POST

URL: http:// 192.168.1.92:5000/ api /template/ queryTemplateByName

Content -Type: application/ json

Request parameters :

Parameter name	type	describe
gro upTem p		Whether to combine templates : 0 All, 1 Individual, 2 Combined
shop Code	String	Store number
tagType	String	Equipment Model
templateName	String	Template Name

Return format:

Parameter name	type	illustrate
code	Int	0: Success 4.00 : Parameter error
message	String	Success or error message
body	String	The default value is empty.

Example:

Request

{
" groupTemp ": 0,
" shopCode ": "0001",
" tagType ": "5C",

```
templateName : "Display Area Video "
}
```

Response


```
{
    "code": 0,
    "message": "success",
    "body": [
    {
            "templateName": "显示区域视频",
            "tagType": "5C",
            "shopCode": "0001",
            "templateType": 1,
            "templateStatus": 0,
            "blocks": 2,
            "preview":

      "api/template/preview_template?
      shopCode=0001&tagType=5C&name=%e6%98%be%
      e7%a4%ba%e5%8c%ba%e5%9f%9f%e8%a7%86%e9%a2%91&timeStamp
      =1661614464253",

            "templateData": null,
            "id": 60,
            "isDeleted": false,
            "createdBy": "System",
            "createdTime": "2022-08-16 16:58:12",
            "lastUpdatedBy": "System",
            "lastUpdatedTime": "2022-08-27 23:34:24"
}
]
}
```


4.5 Signage Template Query ID Interface

Purpose: This interface is used to query IDs for signage templates . This is a system integration-specific interface used to integrate eRetail 3.2 into the customer's system. If not necessary, the eRetail 3.2 backend management system or app can be used directly.

HTTP GET

URL: http:// 192.168.1.92:5000/ api /template/query/60

Content -Type: application/ json

Request parameters :

Parameter name	type	describe

I D	Int	Template I d

Return format:

Parameter name	type	illustrate
code	Int	0: Success
message	String	Success or error message
body	String	The default value is empty.

Example:

Request

```
```

Response

```
{
    "code": 0,
    "message": "success",
    "body": {
        " templateName ": "Display Area Video",
        "tagType": "5C",
        "shopCode": "0001",
        "templateType": 1,
        "templateStatus": 0,
        "blocks": 2,
      "preview": "api/template/preview_template?
      shopCode=0001&tagType=5C&name=%e6
      %98%be%e7%a4%ba%e5%8c%ba%e5%9f%9f%e8%a7%86%e9%a2%
      91&timeStamp=1 661614464253",
        }
}
```

4.6 标牌绑定接口

Purpose: This interface is used for signage binding. It is a system integration-specific interface used to integrate eRetail 3.2 into the customer's system. If not necessary, the eRetail 3.2 backend management system or app can be used directly.

HTTP POST

URL : http : // 192.168.1.92:5000/api/tft/tft/bind

Content -Type: application/ json

Request parameters :

Parameter name	type	describe
areaId	Int	Region ID
bindRes	Entity	List of entities for template elements
displayIndex	Int	Screen index (default value 0)
goods	Array	Product array
shopCode	String	Store number is optional. Note: Either the store number or the customer's store number is required. Template Name
templateName	String	Device ID
tftId		

Return format:

Parameter name	type	illustrate
code	Int	0: Success Parameter validation 400 : Parameter error 1100 : Tag not found . mistake 3201 : Goods {item} is not exist. 1001 : Binding error
message	String	Successorerrormessage
body	String	Thedefaultvalueisempty.

Example:

Request

```
{
" areaId ": 7,
" bindRes ": [],
    "displayIndex": 0,
    "goods": [
        "00011"
    ],
    "shopCode": "0001",
    "shopCodeCst": "A050",
    "templateName": "显示区域视频",
    "tftId": "C49C14844658"
}
```

Response

```
{
```


Limited Use Confidential

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

4.7 Signage Unbinding Interface

Purpose: This interface is used for unbinding signage. This is a system integration- specific interface used to integrate eRetail 3.2 into the customer's system. If not necessary, the eRetail 3.2 backend management system or app can be used directly.

HTTP POST

URL: http:// 192.168.1.92:5000/ api / tft / tft /unbind

Content -Type: application/ json

Request parameters :

Parameter name	type	describe
idList	Array	Devicearray
shopCode	String	Storenumberisoptional.Note:Eitherthestore number or the customer's store number is required.

Return format:

Parameter name	type	illustrate
code	Int	0: Success Parameter validation 400 : Parameter error 3102 : tft is null (tft does not exist)
message	String	Success or error message
body	String	The default value is empty.

Example:

Request

```
{
" idList ": [
"F49520998244"
],
    shopCode : " 0001"
}
```

Response

```
{
```


Limited Use Confidential

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

4.8 Signage screen off/on interface

Purpose: This interface is used for signage screen off/on. This is a dedicated system integration interface for integrating eRetail 3.2 into the customer's system. If not necessary, the eRetail 3.2 backend management system or app can be used directly.

HTTP POST
URL: http:// 192.168.1.92:5000/ api / tft /task / switchTask
Content -Type: application/ json

Request parameters :

Parameter name	type	describe
tftIds	Array	Devicearray
taskCode	Int	Taskcodes:11:Turnoffbacklight,12:Turnon backlight

Return format:

Parameter name	type	illustrate
code	Int	0: Success Parameter validation 400 : Parameter error mistake 1001 : TFTIds is not null (Color screen ID is not empty) 1001: Switch Task Settings HOST is null (Screen on/off task settings HOST is empty) 1001: Switch Task TFTIds is error: (Error message regarding device MAC address for screen on/off task)
message	String	Success or error message
body	String	The default value is empty.

Example:
Request

Limited Use Confidential

```
{
" tftIds ": [
"C49C14844658"
],
" taskCode ": "11"
}
```


Response


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


4.9 Signage Refresh Interface

Purpose: This interface is used for signage refresh. It is a system integration-specific interface used to integrate eRetail 3.2 into the customer's system. If not necessary, the eRetail 3.2 backend management system or app can be used directly.

HTTP POST

URL: http:// 192.168.1.92:5000/ api / tft / tft /refresh

Content -Type: application/ json

Request parameters :

Parameter name	type	illustrate
tftId	String	DeviceID
shopCode	String	Storenumber

Return format:

Parameter name	type	illustrate
code	Int	0:Success Parameter validation 400 : Parameter error mistake 1001: No items selected. 1001: Refresh error
message	String	Successorerrormessage
body	String	Thedefaultvalueisempty.

Example:

Request

```
{
" tftId ": [
"F4952099968E"
],
shopCode : " 0001"
}
```

Response

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

5. Product Management

5.1 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/save List
Content -Type: application/ json

Request parameters : -- array

URL parameter: NoTemplate: Whether to update the template parameters; true indicates that no update is needed.

Parameter name	type	describe
shopCode	String	The system store number requires corresponding store to be created in the system store list first.
template	String	Thetemplatenameneedstobecreatedinthe template list first.
items	ArrayString[27]	Productdataattributesarecurrentlyfixedat2and7 (expandable).

Note: For information on creating stores and templates, please contact Zhsunyco sales staff or pre-sales/after-sales support engineers.

Return format:

Parameter	type	illustrate

Limited Use Confidential

name		
code	Int	0:Success All others failed. 1001 : Shop does not exist. 9801 : ShopCode is null (Store ID is empty) 9801 : UPC index too long . 9801 : Cannot find goods code index. 9801 : Goods name index too long
message	String	Successorerrormessage
body	JsonNode	Messagebody

Example:

http:// 192.168.1.92:5000/ api /goods/ save List

Request

```
[
{
" shopCode ": "0001", // Customer store number
"template": "REG", // Price tag type: REG - Regular Sale, SAL - Promotion, NOR - Out of Stock, MER - Membership...
"items": [
"A050", // Customer store number
"123456", // Unique product code: can be the product number or product barcode.
"Product 1", //Product Name
"123456789012", //UPC1: Product barcode 1 or code
"123456789013", // UPC2: Product barcode 2 or code
"123456789014", // UPC3: Product barcode 3 or code
"8.98", // Price 1: Retail price or original price
"8.95", // Price 2: Promotional Price
"8.96", // Price 3: Member Price
"Shanghai", //Place of Origin
300ml, // Specification
"bottle", //unit
"Qualified", //Grade
"2021/12/20", //Promotion start date
"2021/12/25", // Promotion end date
http://www.baidu.com, //QR code
"Zhang San", //Price clerk
"5.1", // Inventory
"Extension 1", //Extension field 1 was not empty "
"Extension 2", //Extension field 2
"Extension 3", //Extension field 3
"Extension 4",
```


Limited Use Confidential

```
"Extension 5",
"Extension 6",
"Extended  7",
"Extended  8",
"Extension 9",
"Extended
10" ]
}
]
```

Response

```
{
"code": 0,
"message": "success",
"body": "121f5151fdfffds21cdf"
}
```

Note: By default, the field attributes in this example are fixed. In practice, the content and template of the product data fields can be freely set.

5.2 Product Update Interface - Specifying Fields

Purpose: This interface is used to update specified fields of a product. Note that modifications to the primary key, product name, and UPC are invalid.

HTTP POST
URL: http://192.168.1.92:5000/api/goods/ updateItems
Content -Type: application/ json

Request parameters : -- array

Parameter name	type	describe
shopCode	String	Systemstorenumber
updateIndex	IntArray	Updatefieldcorrespondingsequencenumber - starting from 0
updateList	Object rray	AProductDataList
goodsCode	String	Productuniquecode
template	String	TemplateName-Leavingthisfieldblank indicates no modification to the product template; can be left empty. Listofproductfieldvalues
items	StringArray	

Limited Use Confidential

Note: For information on creating stores and templates, please contact KCL Kuwait sales staff or pre-sales/after-sales support engineers.

Return format:

Parameter name	type	illustrate
code	Int	0:Success All others failed.
message	String	Successorerrormessage
body	JsonNode	Messagebody

Example:

http:// 192.168.1.92:5000/ api /goods/ updateItems

Request

```
{
" shopCode ": "0001", //Store Number
" updateIndex ": [        // Specifies the field to update the index]
7,
8
],
" updateList ": [ // Update product list]
{
" goodsCode ": "123456", //Unique product code
"template": "", // Template type can be empty
"items": [ // Update the specified data list]
"8.98",
8.95
]
},
     {
" goodsCode ": "123457",
"template": "",
"items": [
"8.99",
8.95
]
}
]
}
```

Response

```
{
"code": 0,
```


Limited Use Confidential

"message": "success",
"body": null
}

5.3 Product Inquiry

Purpose: Used to query product information based on the customer's store number.

HTTP POST
URL : http://192.168.1.92:5000/api/Goods/getList
Content -Type: application/ json

Request parameters :

Parameter name	type	illustrate
pageIndex pageSize	Int Int	pagenumber
sort	string	Numberofpages
desc shopCodeCst	string	Empty
string		canbeemptyasc/desc
		Emptycustomerstorenumber
shopCode	string	Emptysystemstorenumber
goodsCode	string	Emptyproductuniquecode

Return format:

Parametername	type	illustrate
code	Int	0:Success
message	String	Successorerrormessage
body	Objective	ReturnEntity
totalCount	int	total
itemlist	ObjectArray	ProductEntityList
id	string	Productprimarykey
shopCodeCst	string	Customerstorenumber
shopCode	string	Systemstorenumber
goodsCode	string	Productuniquecode
goodsName	string	ProductName
template	string	TemplateName
upc	StringarraySearchcode	

Limited Use Confidential

items	String Array	Product details array, subject to actual data.

Example:

Request


```
{
" pageIndex ": 1,
" pageSize ": 100,
sort": "",
"desc": "",
   "shopCodeCst": "012412",
   "shopCode": "",
   "goodsCode": ""
}
```


Response


```
{
   "code": 0,
   "message": "success",
   "body": {
      "totalCount": "999",
      "itemList": [
      {
      "shopCodeCst": "012412",
      "id": "00011000786815",
      "shopCode": "0001",
      "goodsCode": "1000786815",
      "goodsName": "A",
      "template": "REG",
      "upc": [
         "1000786815"
      ],
      "items": [
         "14297",
         "1000786815",
         "A",
         "",
         "6903148083109",
         "0.00",
         "34.90",
"0",
"0.00",
2.8KG
```


Limited Use Confidential

```
"",
"",
"qualified",
"",
"",
"bag",
"bag",
"0518",
"60081373
",          "",
Guangzhou
      ]},
      ......
         ]
      }
}
```

5.4 Product Deletion

HTTP GET
URL: http://192.168.1.92:5000/api/Goods /delete/{id}
Content -Type: application/ json

Request parameters :

Parameter name	type	illustrate
id	string	Aproduct'suniqueIDistypically:storenumber+ product code.

Return format:

Parametername	type	illustrate
code	Int	0:Success 1001: Failure
message	String	Successorerrormessage
body	String	Thedefaultvalueisempty.

Example:
Request

```
http:// 192.168.1.92:5000/ api /Goods/delete/0001123456
```


Limited Use Confidential

Response

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

5.5 Product Template Switching Plan

Purpose: This interface is used to specify timed template switching for products in batches.

HTTP POST
URL : http://192.168.1.92:5000/api/Schedule/GoodsTemplateSwitch
Content -Type: application/ json

Request parameters :

Parameter name	type	illustrate
goodsList	ObjectArray	Product array System
shopCode	string	store number Product
goodsCode	string	Code
scheduleList	ObjectArray	Plan array
startTime	dateTime	Start time
endTime	dateTime	End time
template	string	Template Name

Return format:

Parametername	type	illustrate
code	Int	0: Success Other: Failure
message	String	Success or error message
body	String	The default value is empty.

Example:
Request

```
{
" goodsList ": [
{
```


Limited Use Confidential

```
" shopCode ": "A0001",
" goodsCode ": "100"
},
{
        "shopCode": "A0001",
        "goodsCode": "1000"
      }
   ],
   "scheduleList": [
      {
         "startTime": "2024-09-18 02:11:58",
         "endTime": "2024-09-20 02:11:58",
         "template": "REG"
      }
   ]
}
```


Response


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


6. Data Synchronization Instructions

Data synchronization includes common methods: based on database/intermediate table/view query statements or stored procedures; based on files of a specific format (such as Excel, CSV/TXT, XML, JSON, etc.); based on FTP file systems, etc.

6.1 D2M Dynamic Data Model

For simple data structures, a dynamic model (D2M) can be used for data synchronization. This is typically the responsibility of the implementation team, who will select the appropriate model based on 1) the data structure definition and 2) the template switching logic definition. See eRetail for details. D2M configuration page.

6.2 Importing Products from Excel

After setting up the dynamic model for product data, you can export the corresponding data template from "Data Management - Product Management". Then, add or edit product data and import it into the system to adjust prices.

Note: The Excel file format is xlsx, the file size is less than 10MB, and the number of product data entries is around 100,000.
The template field is fixed as "template" and cannot be modified.

6.3 Customize Customized development

Customised development is available for situations where the above-mentioned solutions cannot effectively address the customer's system integration needs. For specific solutions, please contact the KCL Kuwait project department for further information.

Limited Use Confidential

شركة كويت كاربوريشن 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 ®: 16437921

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