2023

eRetail3.1 Docking Scheme Documentation

V1.2

DALIAN SERTAG TECHNOLOGY CO., LTD

Historic Version

Version Number	Date	Describe	Framer	Reviewer
1.0	2022/3/31	First edition		
1.1	2023/10/25	Active docking, UI interface modification	Merlin	Tony
1.2	2023/11/15	eRetail3.1 Docking Scheme Documentation translate	Kris	Tony

Contents

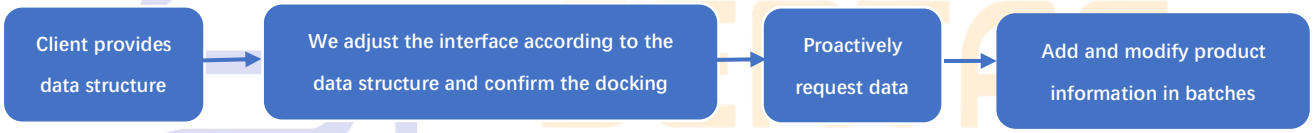
1. Summary

eRetail3.1 supports docking methods: Data Sync/API, hardware equipment, API is Restful Web API, which provides an interface for customer systems to actively push product data; Data Sync is a background service that actively captures data from customer systems (Database, Web API, Excel).

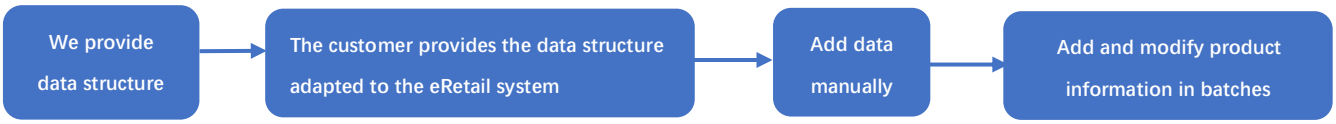
2. Product Data Docking Method

Data docking: active docking, passive docking, and no docking.

- Active docking: eRetail3.1 actively requests data, requiring the client to provide data access rights, connection information, data structure, product sales logic, and active docking: database docking, files, FTP files, customization (API interface).



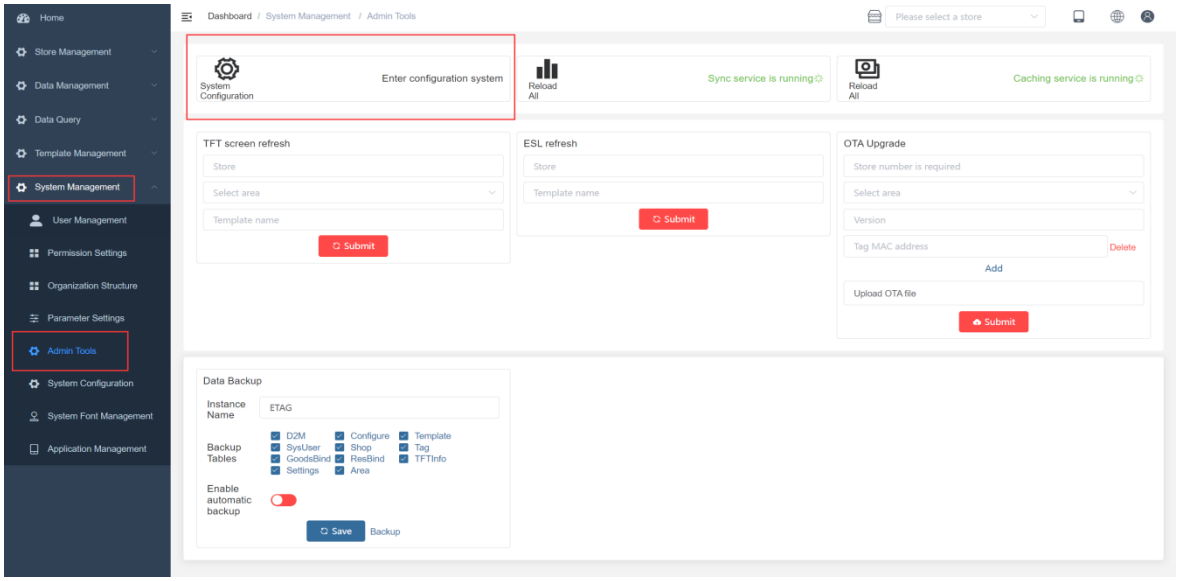
- Passive docking: To receive data sent by customers, we provide corresponding API interface documents and data structures.



- No docking: Product data is maintained in eRetail3.1, and product data is added and modified through background tables.

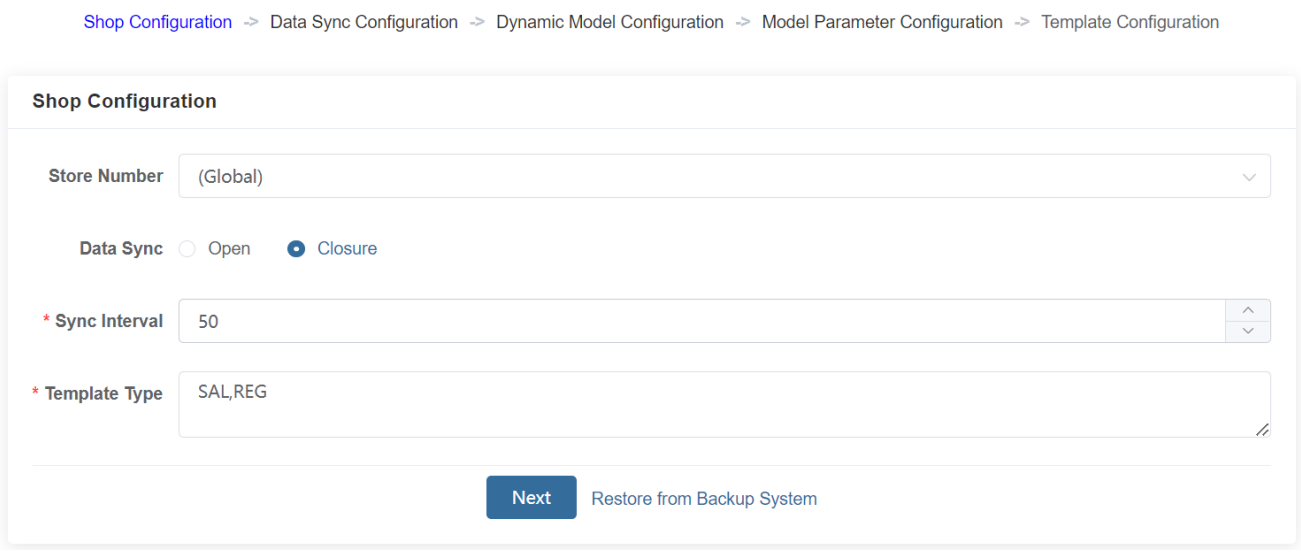
3. Active Docking

Parameter configuration: Log in eRetail3.1 - System Management - Admin Tools - Fill in the docking parameters in the System Configuration.

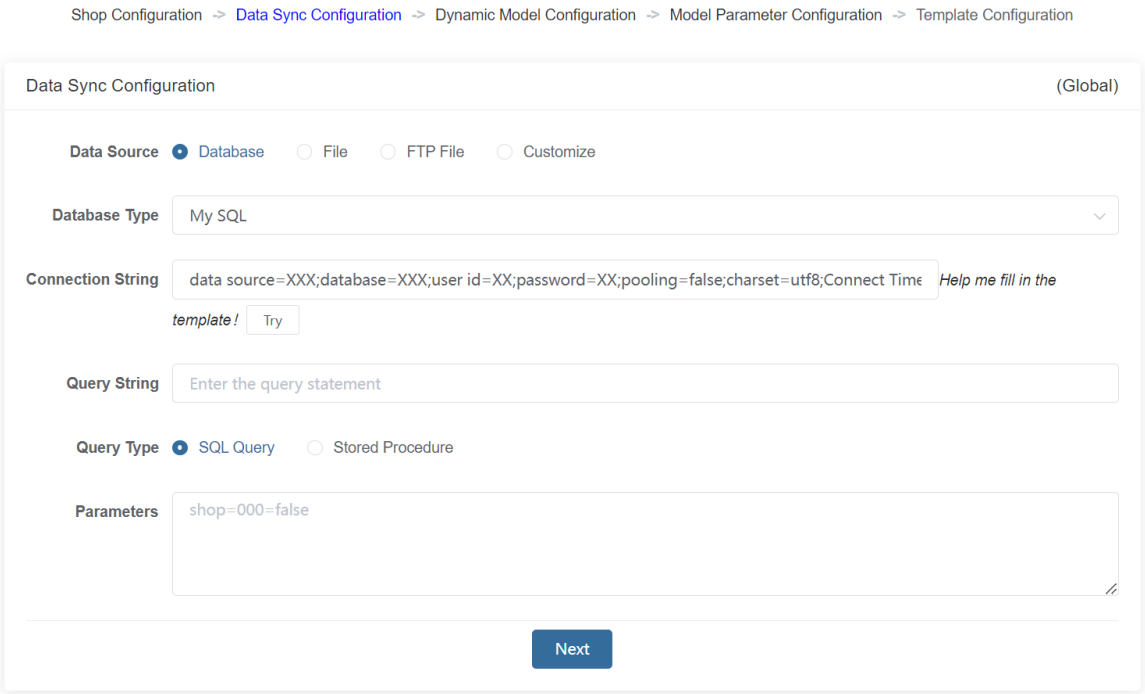


3.1 Store Configuration

- Store Number: The global configuration is the configuration information of all eRetail3.1 stores; the single store configuration is the configuration information of the specified store.
- Data Sync: Active docking must be turned on.
- Sync Interval: Synchronization period, unit second.
- Template type: ESL template name, needed when making templates.



3.2 Data Sync Configuration



Database: The customer uses a database containing product data; provides database connection parameters and product data structure information.

- Database type: SQL Server, My SQL, Sybase, Oracle, and Sqlite.
- Connection string: Click to help me fill in the template and change parameters.

Database	SQL Server
Params	Meaning
server	DB IP address
uid	DB username
pwd	DB password
database	DB instance

Database	Sybase
Params	Meaning
Data Source	DB IP address
Initial Catalog	DB instance
Userid	DB username
Password	DB password

Database	Sqlite
Params	Meaning
Data Source	DB file path + DB file name

Database	My SQL
Params	Meaning
data source	DB IP address
database	DB instance
userid	DB username
password	DB password
pooling	Connection limitations
charset	Encoding type
Connect Timeout	Maximum expiration time

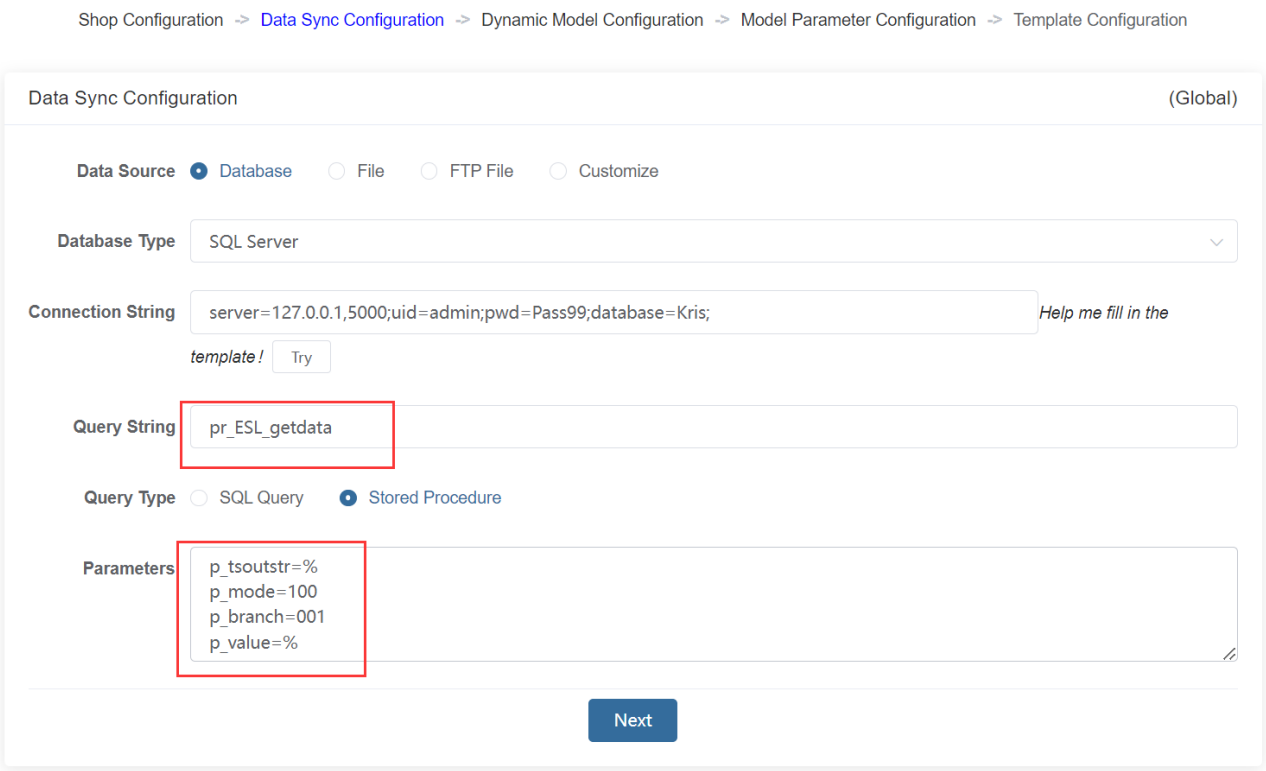
Database	Oracle
Params	Meaning
HOST	DB IP address
PORT	DB port
SERVICE_NAME	DB instance
UserId	DB username
Password	DB password
Min Pool Size	Connection limitations
Connection Timeout	Maximum expiration time

➢ Try: Test the database connection information filled in. Click the upper right corner of Try to prompt success.

➢ Query String: Fill in the corresponding String according to the selected Query Type.

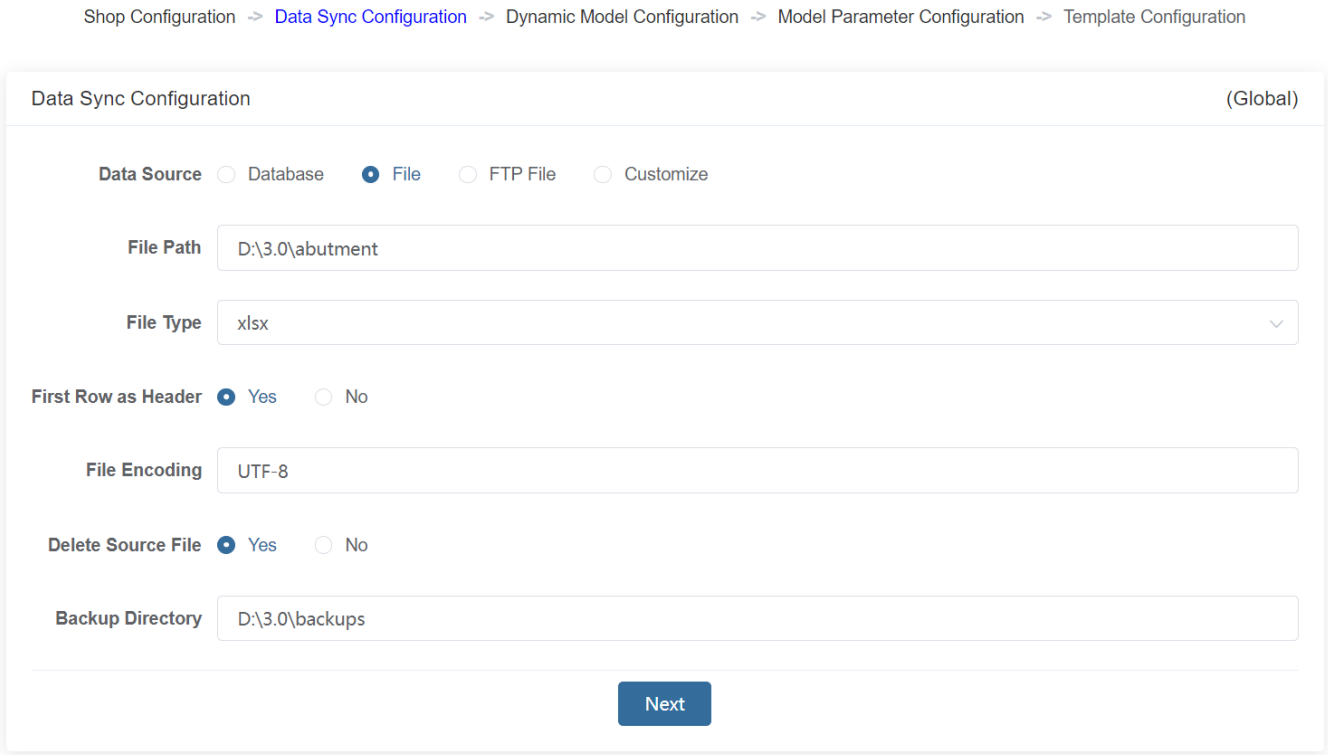
➢ Parameters: When Query Type selects a Stored Procedure, fill in the stored procedure parameters.

(1) SQL Server 2000 requires tools to synchronize data from the 2000 version to the 2012 version in real time (the tools are provided by us), and then use the 2012 version data to connect.



Configure based on stored procedure data in the customer database.

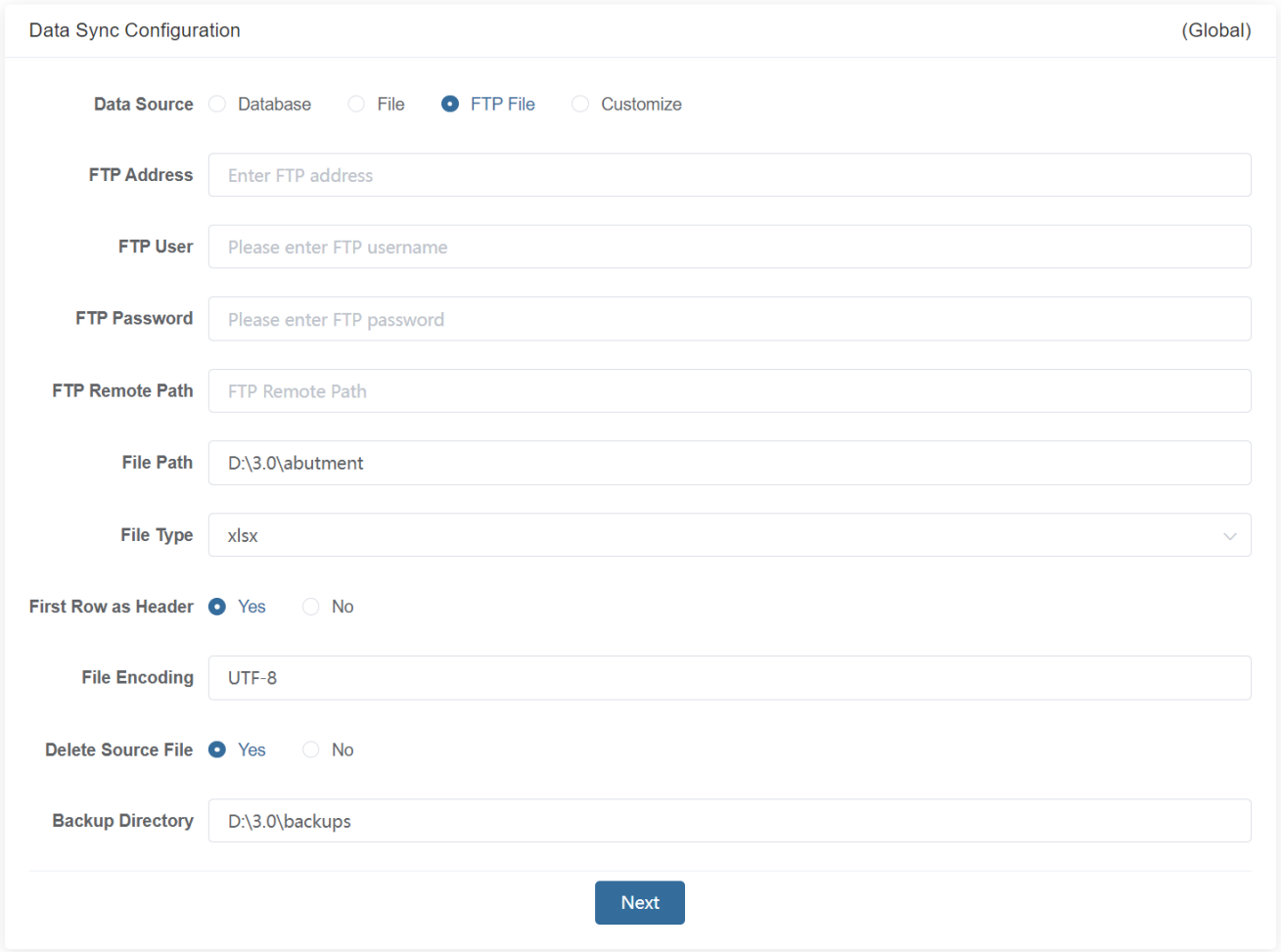
File: The product data file in the specified file path, and eRetail3.1 reads the files in the path in real time.



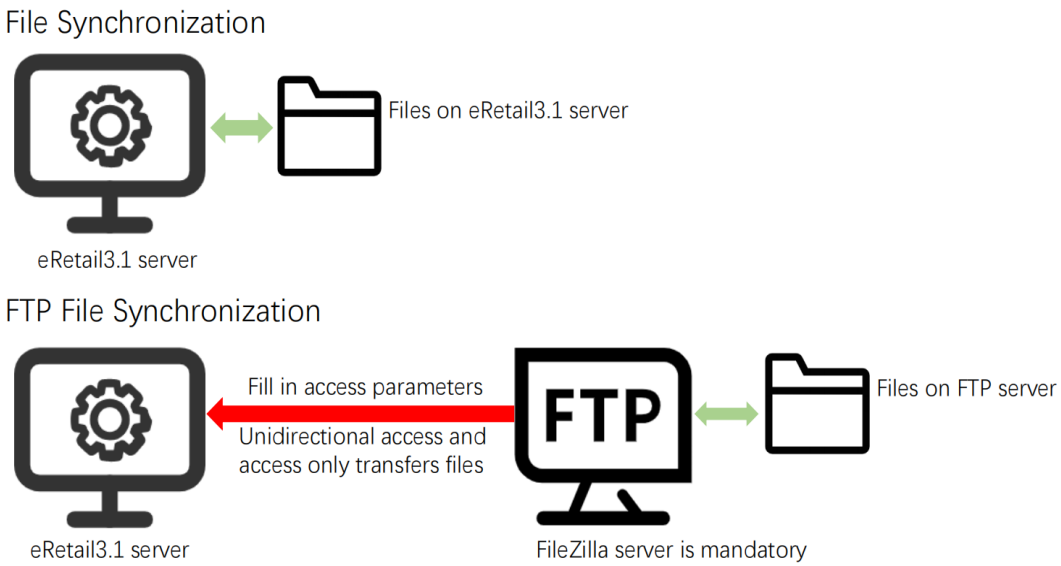
File: The product data file in the specified file path, and eRetail3.1 reads the files in the path in real time.

- File Path: Sync file path.
- File Type: Supports xlsx, xls, csv, txt file formats.
- First Row as Header: The first line is the field name, select Yes; or the product data, select No.
- File Encoding: The default format is UTF-8.
- Delete Source Files.
- Backup Directory: The synchronized files will be deleted, and the deleted files will be stored in this path.
- File fields cannot contain a field named template.

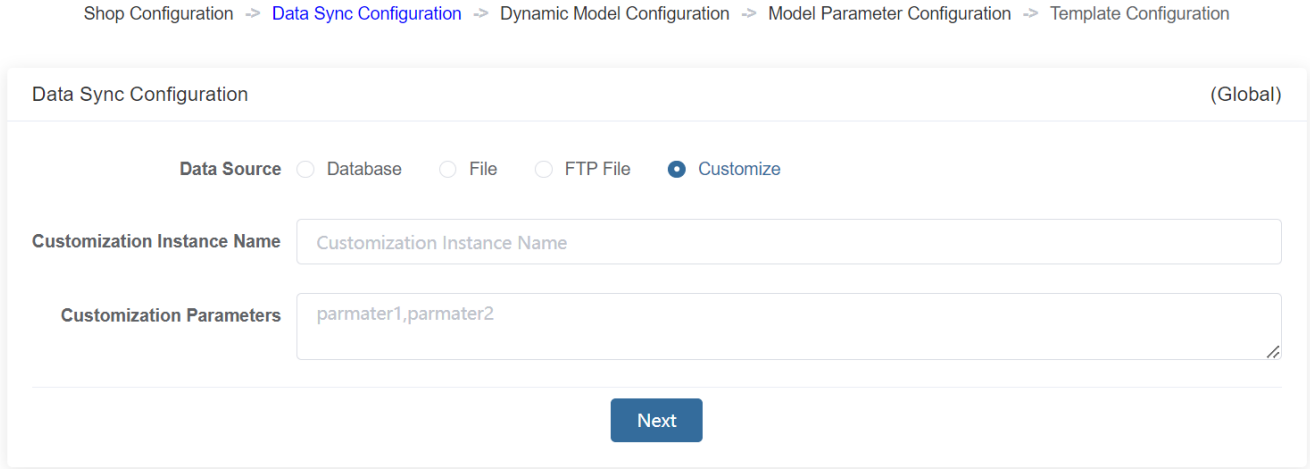
FTP files:



- FTP Address: IP of FTP software server.
- FTP User: FTP software server login name.
- FTP password: FTP software server login password.
- FTP Path:FTP software server storage data file path.
- Other settings: same as file.
- Differences between files and FTP files:



Customize: The customer provides the product data API interface, and we obtain the data through the interface. Fill in the connection parameters with the customized instance name and customized parameters.



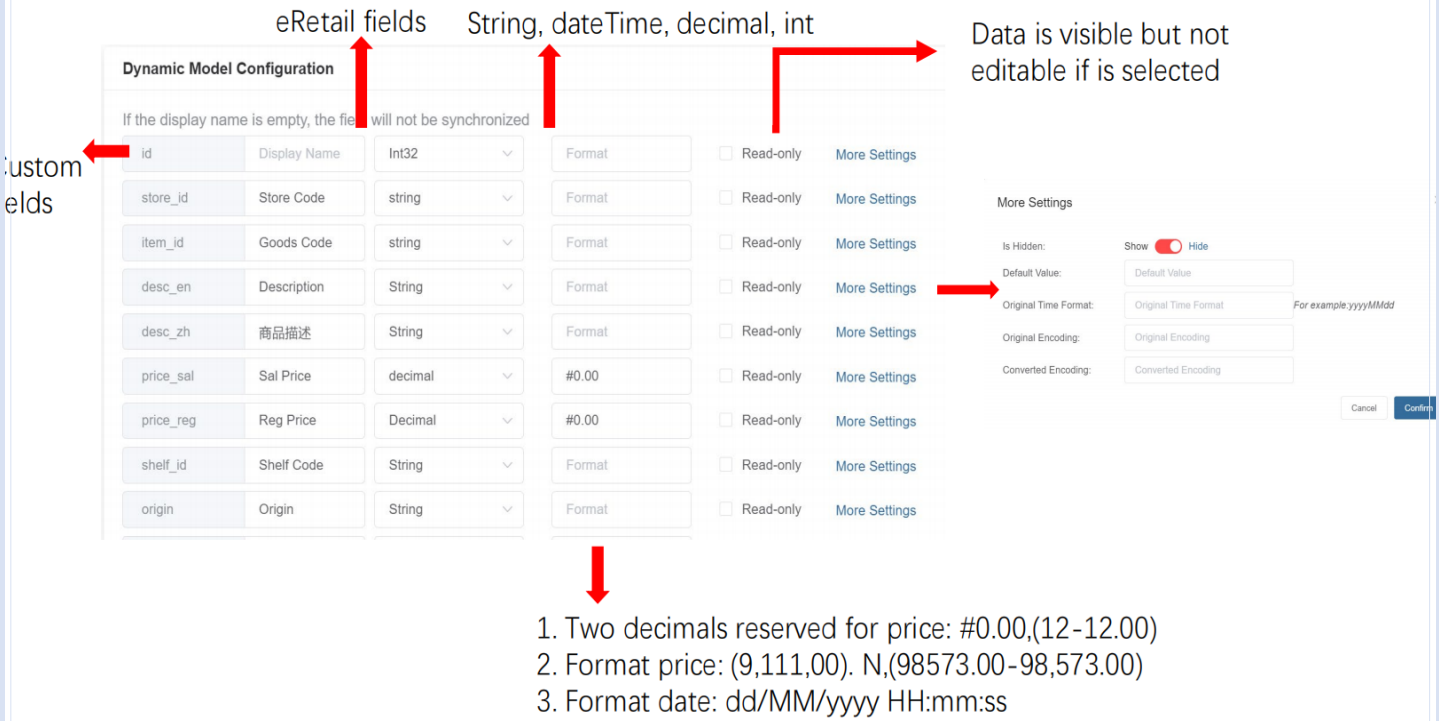
Customer provides information:

- Various request addresses related to ESL: Store Number, Product Data, Key, etc.
- Various request parameters related to ESL: Store Number, Product Data, Key, etc.
- Various response parameters related to ESL: Store Number, Product Data, Key, etc.
- ESL require fields and field meanings.
- Product sales method judgment logic.

➢ Product field data conversion logic.

3.3 Dynamic Model Configuration

Customized docking requires no settings.



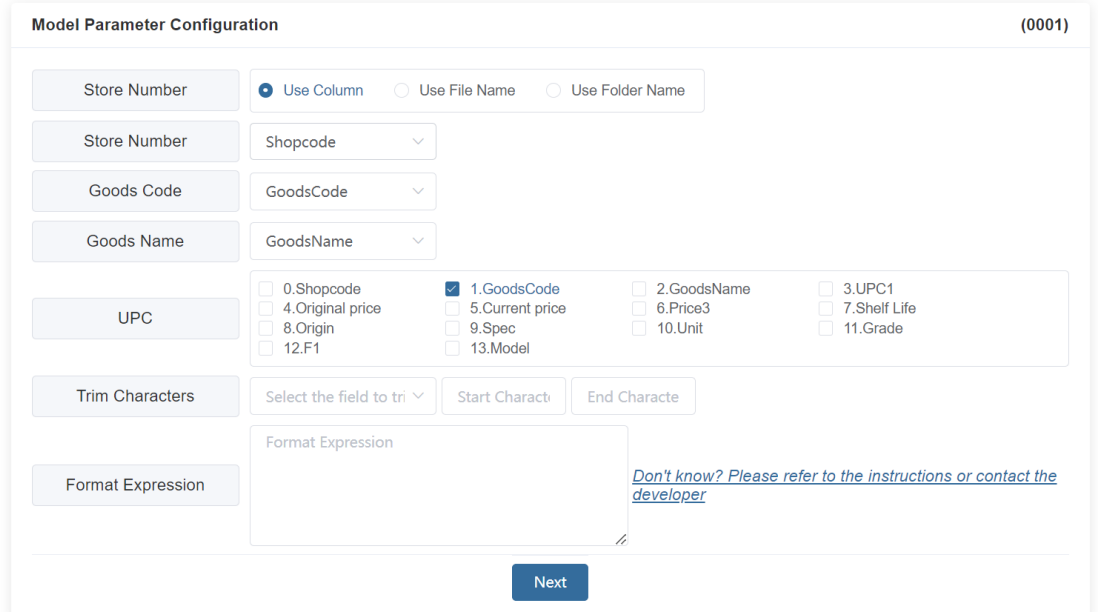
➢ Is Hidden: Green means display, red means hide. After hiding, the product query on the APP can't display this field and field information.

➢ Default Value: Filling in the format data will replace the original data.

➢ Original Time Format: For example, time 02122023 can be filled in MMddyyyy

➢ Original Encoding\Converted Encoding: Some product data encoding formats can't be recognized by the eRetail3.1, and direct acquisition will result in garbled codes. Fill in the customer's product data encoding format with the original code,

and fill in the encoding format supported by the eRetail3.1 with the converted code.



3.4 Model Parameter Configuration

- Store Number: Use Column, Use File Name or Use Folder Name.
- Store Number: You must be choose Use Column if you use to select Store Number.
- Goods Code: The value of this field must be unique under the same store.
- Goods Name: Select the connected product name field.
- UPC: The field used to unlock the product when ESL is bound to the product. Usually the default Goods code is UPC.
- Trim Characters:

Remove fixed characters at the beginning or end of the selected field

Beginning Char0	Beginning Char 12	Beginning Char 12, End Char 4
Custom data Goods UPC: 00789123 Goods UPC: 00789443	Custom data Goods UPC: 123478954 Goods UPC: 123478664	Custom data Goods UPC: 123478954 Goods UPC: 123478664
Converted data Goods UPC: 789123 Goods UPC: 789443	Converted data Goods UPC: 3478954 Goods UPC: 3478664	Converted data Goods UPC: 347895 Goods UPC: 347866

Note: Goods Code is not supported for trim!

- Format Expressions: Transform customer data.

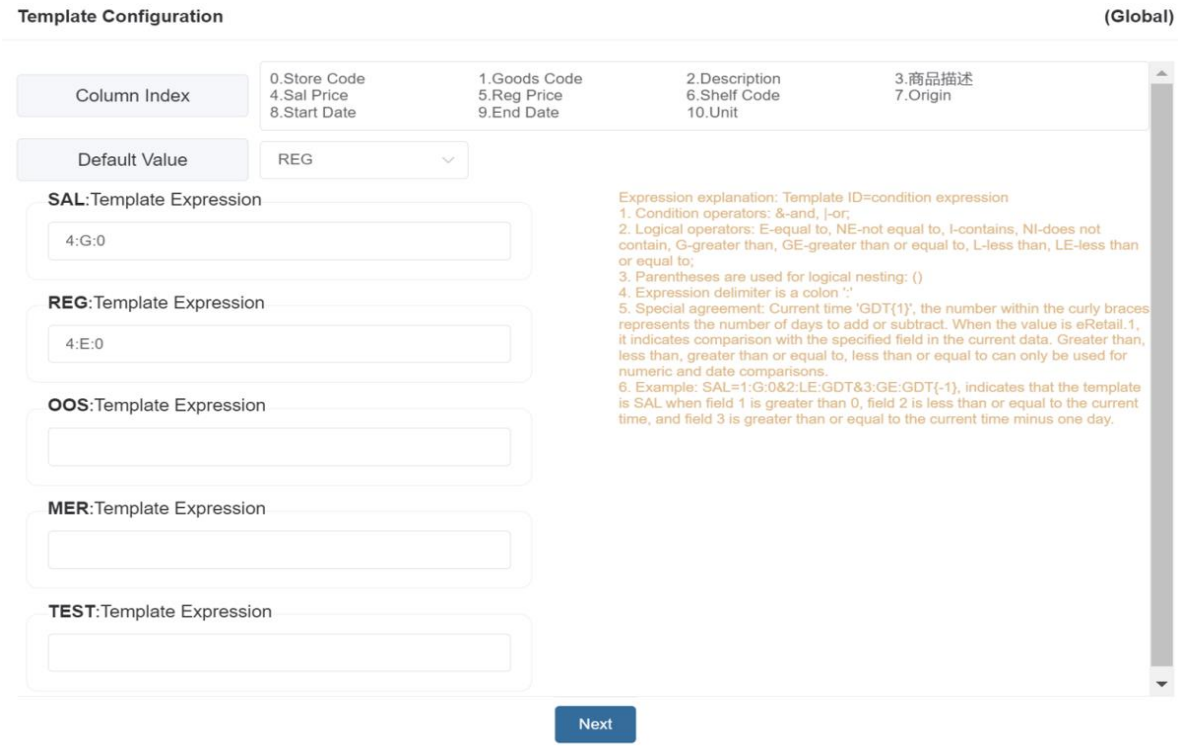
1. 4#0.5=10:E:KG|10:E:kg|10:E:Kg|10:E:kG|10:E:公斤
Sal price will be divided by two when Unit meets KG/kg/Kg/kG/公斤

2. 5#0.5=10:E:KG|10:E:kg|10:E:Kg|10:E:kG|10:E:公斤
Reg price will be divided by two when Unit meets KG/kg/Kg/kG/公斤

3. 10~500g=10:E:KG|10:E:kg|10:E:Kg|10:E:kG|10:E:公斤
Unit will be fixed as "500g" when Unit meets KG/kg/Kg/kG/公斤

4. Some rounded fields use @ instead of #, to make sure the calculated result keep in Integer.
For example:
When the inventory number is greater than 500, multiply the number of storage by #Inventory Value@2, and the calculation result of @ is an integer.

3.5 Template Configuration



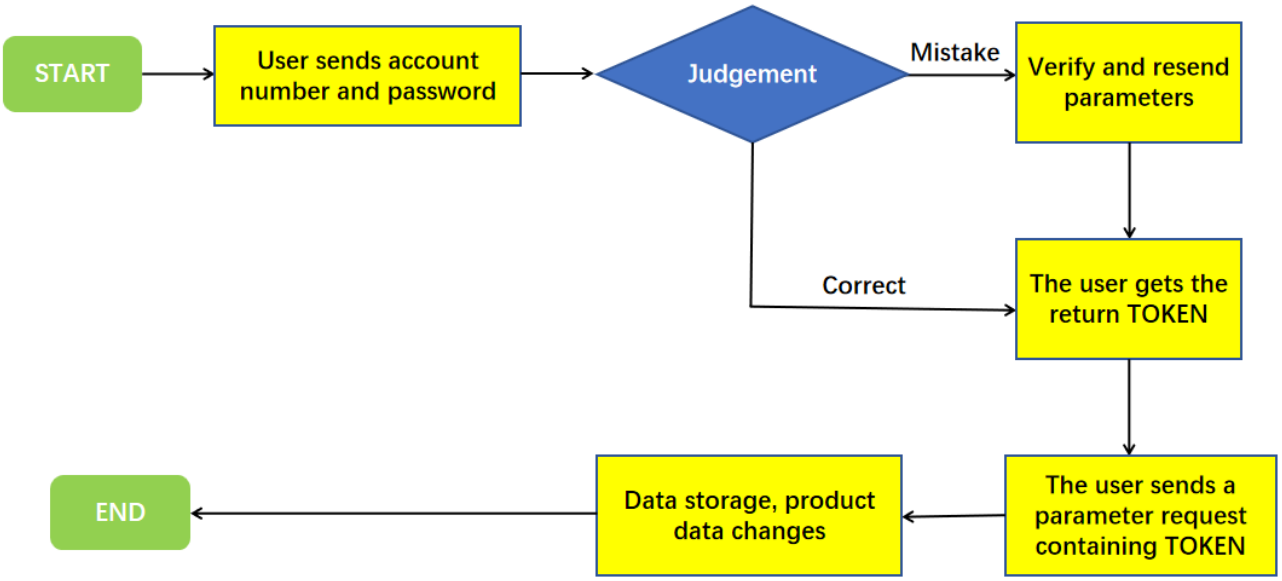
- Expression explanation: Template ID=condition expression.

- Default value: Usually empty and default REG type.

- Condition operators: &-and, |-or;
- Logical operators: E-equal to, NE-not equal to, I-contains, NI-does not contain, G-greater than, GE-greater than or equal to, L-less than, LE-less than or equal to;
- Parentheses are used for logical nesting: ()
- Expression delimiter is a colon ':'
- Special agreement: Current time 'GDT{1}', the number within the curly braces represents the number of days to add or subtract. When the value is eRetail.1, it indicates comparison with the specified field in the current data. Greater than, less than, greater than or equal to, less than or equal to can only be used for numeric and date comparisons.
- Example: SAL=1:G:0&2:LE:GDT&3:GE:GDT{-1}, indicates that the template is SAL when field 1 is greater than 0, field 2 is less than or equal to the current time, and field 3 is greater than or equal to the current time minus one day.
- Note: The above list can be used as a reference.

4. Passive Docking

Data Sync option is turned off.



4.1 Calling Method

Handshake interface:This interface is used for authentication. All subsequent interface accesses 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	Describe
userName	String	Username (provided by us)
password	String	Password (provided by us)

- Return Format:

Parameter Name	Type	Describe
code	Int	0: success, other: error
message	String	success or error message
body	Json Node	message body
token	String	Token for subsequent sessions

Notice: After obtaining the Token, this content needs to be added to the header of subsequent HTTP requests. Like:"Authorization: Bearer {token}"

- Example:

Request

```
{
  "userName":"port",
  "password":"Port99"
}
```

Response

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

```
    }
}
```

Product data interface: 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	Describe
shopCode	String	Store number, create the corresponding store.
template	String	Template name, create the corresponding template.
items	Array String[26]	Product data attributes, fixed at 26 (**expandable**)

◆ Return format:

Parameter Name	Type	Describe
code	Int	0: success, other: error
message	String	success or error message
body	Json Node	message body

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

◆ Example:

Request

```
[
  {
    "shopCode":"0001",//Customer store number
    "template":"REG",// ESL type: REG-General Sales, SAL-Promotion, NOR-Out of Stock....
    "items":[
      "A050",/Store number
      "123456",//Product code
      "Product 1",//Product name
      "1",//Product abbreviation
      "123456789012",//UPC1
```

```
        "123456789013",// UPC2
        "123456789014",// UPC3
        "8.98",// REG price
        "8.95",/SAL price
        "8.96",//Member price
        "8.96",//Original price
        "Shanghai",//Origin
        "300ml",//Specification
        "qualified",//Grade
        "bottle",//Unit
        "2021/12/20",//Promotion start date
        "2021/12/25",//Promotion end date
        "2021/12/20",//member start date
        "2021/12/25",//member end date
        "Zhang San",//Price clerk
        "5.1",//In stock
        "Delicatessen",//Category code
        "http://www.baidu.com",//QR code
        "state",//state
        "Extension 1",
        "Extension 2",
        "Extension 3",
        "Extension 4",
        "Extension 5",
        "Extension 6",
        "Extension 7",
        "Extended 8",
        "Extension 9",
        "Extension 10"
    ]
  }
]
```

Response

```
{
```

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


Note: The field properties in this example are fixed. In fact, product data field content and templates can be set freely.

4.2 System Integration Interface

Integrate eRetail3.1 into the customer system, and the customer system operates, binds, and unbinds... eRetail3.1. ESL continues to work in eRetail3.1.

ESL data push refresh: Customers push product data and refresh to the specified ESL. The system background will not save user data.

- HTTP POST

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

Content-Type: application/json

- Request parameters:

Parameter Name	Type	Describe
shopCode	string	Store number
tagID	string	ESL ID
ap	string	Send from designated base station(Default empty)
item	Object	Data entity
GoodsCode	string	Product unique code
GoodsName	string	Product name
Template	string	Template name
Items	Array	Data details array (Fields are used to bind to template display)

- Return format:

Parameter Name	Type	Describe
code	Int	0: success, other: error
message	String	success or error message
body	String	Default is empty

- Example:

Request

```
{
  "shopCode":"0002",
  "tagID":"4F000001320A",
  "ap": "",
  "item":{
    "GoodsCode" :"123456",
    "GoodsName" :"Obatley Castle Dry Red",
    "Template" :"SAL",
    "Items" :[
        "0002",/Store number
        "123456",//Product code:Unique product code
        "Obatley Castle Dry Red", //product name
        "Castle Dry Red",//Product abbreviation
        "123456789012",//UPC1
        "123456789013",// UPC2
        "123456789014",// UPC3
        "118.98",//REG price
        "110.95",/Sale price
        "118.96",//Member price
        "118.96",//Original price
        "Shanghai",//Origin
        "300ml",//Specification
        "qualified",//Grade
        "bottle",//Unit
        "2021/12/20",//Promotion start date
        "2021/12/25",//Promotion end date
        "2021/12/20",//Member start date
        "2021/12/25",//Member end date
        "Zhang San",//Price clerk
        "50",//In stock
        "Liquor",//Category code
        "http://www.baidu.com",//QR code
        "state",
        "Extension 1",
```

```
            "Extension 2",
            "Extension 3",
            "Extension 4",
            "Extension 5",
            "Extended 6",
            "Extension 7",
            "Extended 8",
            "Extension 9",
            "Extension 10"
        ]
    }
}
```

Response

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

Other interfaces: Refer to Sertag eRetail3.1 System Integration Manual.

5. No Docking

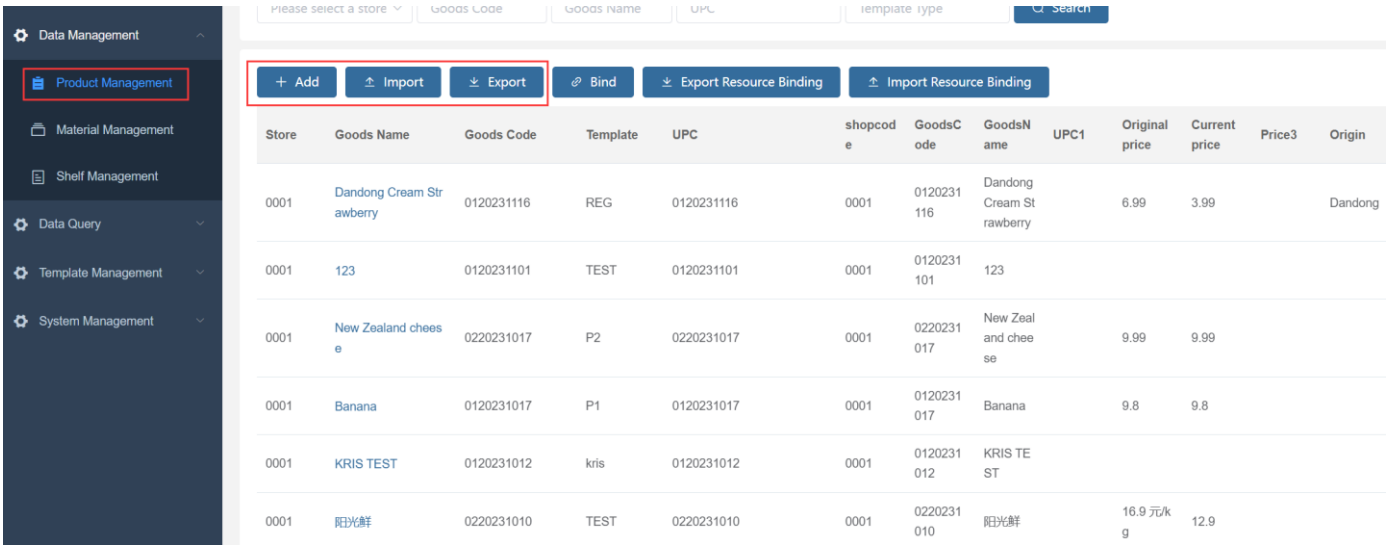
Data Sync option is turned off

Data structure: The following field information is used by default and can be modified or added.

Store Number
Goods code
Goods Name
Product Abbreviation
UPC1
UPC2
UPC3

REG price
SAL price
Member price
Original price
Origin
Specification
Grade
Unit
Promotion start date
Promotion end date
Member start date
Member end date
Price clerk
Inventory
Category code
QR code
Status
Extension 1
Extension 2
Extension 3
Extension 4
Extension 5
Extension 6
Extension 7
Extension 8
Extension 9
Extension 10

5.1 Product Data Maintenance



- Product data is maintained in [eRetail3.1-Data Management-Product management]
- Download the data structure form, fill in the product information in the form, and then import the form into the system to implement batch import and modification of product data.
- Add and modify individual products through the add and modify buttons.