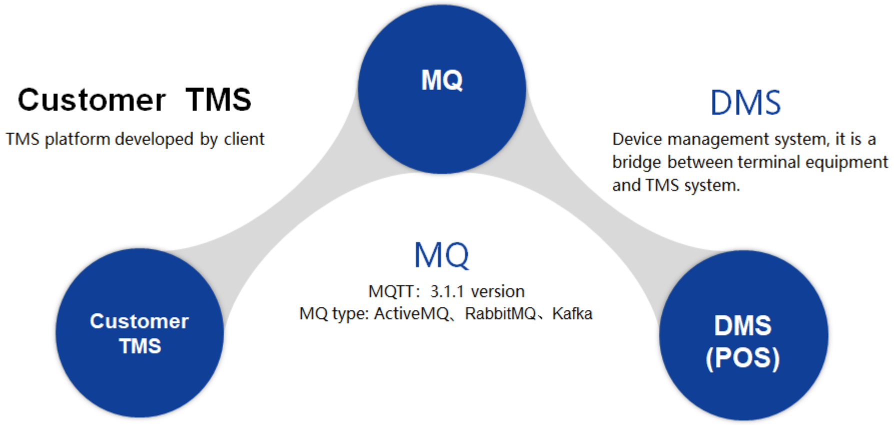
DMS standard document V1.0

Time	Version	author	**content**
2020-09-16	1.0	Allen	First version of DMS interface document

1、Project Background

In order to meet the international market and regional policy requirements, DMS International Standard Version v1.0 was officially released.

2、System Architecture



TMS platform developed by client

Device management system, it is a bridge between terminal equipment and TMS system.

MQTT: 3.1.1 version
MQ type: ActiveMQ、RabbitMQ、Kafka

3、Communication protocol

3.1 Rules of agreement

Calling the API must follow the following rules:

transmission mode	In order to ensure the security of the data, the formal environment should adopt the HTTPS transmission in principle, and the test environment should be HTTP transmission；The server push message adopts MQTT protocol

Submission method	Submit by post method
Data format	Except for some file upload and download interfaces, the request and return data are in JSON format and content- Type:application/json
Character encoding	Unified use of UTF-8 character coding
signature algorithm	At present, the signature is MD5, and other signature methods may be supported in the future.
Signature requirements	Both request return and asynchronous notification need to verify the signature. See 5.3 for detailed signature method4.3
Judgment logic	First judge the return of protocol field (HTTP status code), then judge the message return code, and finally judge the data status
Language support	Accept language: the language that the client can accept, such as en US, Zh CN, etc., currently in Chinese and English. The value is passed through the HTTP header

3.2 Parameter specification

necessity

- M – Required parameter
- C – It is a required parameter when some conditions are satisfied
- O – Optional parameters

Parameter type

Parameter KEY	parameter Type	Examples	explain
NUMBER	Digital class	123	
AMOUNT	Amount category	88.05	

TEXT	Text class	Allen	
DATE	Time class	2018-08-02 15:16:51	Greenwich mean time (utc-0) format: yyyy-mm-dd HH: mm: SS
BOOLEAN	Boolean class	true	true or false
JSONObject	JSON object class	{"key":"value"}	
JSONArray	JSON array class	[1,2,3,4,5]	

Request message - pbulic parameter

Parameter KEY	parameter Name	Type	Necessity	describe
signType	signature type	TEXT(16)	M	default value MD5
signValue	signature value	TEXT(32)	M	Prevent message tampering
version	interface version number	TEXT(8)	M	default value： 1.0
isEncrypted	Encryption or not	NUMBER	M	default value: 1(ciphertext); 0（Plaintext)

Response message - public parameter

Parameter KEY	parameter Name	Type	Necessity	describe
code	status code	TEXT(16)	M	Status code, 0 means successful request, others indicate failure
msg	error message	TEXT(128)	M	When an error occurs, this msg contains error information
data	retrurn all the data	JSONArray	M	The array contains one or more jsonobjects, and the specific parameters are defined by each business API

An example of the requested message format is as follows:

```
{
    "signType": "MD5",
    "signatureValue": "xxxxxxxxxxxxx",
    "version": "1.0",
    "isEncrypted": 1,
    "data": "ciphertext(xxxxxxxxxxx)"
}
```

The response message format is as follows:

```
{
    "code": "0",
    "data": "ciphertext(xxxxxxxxxx)",
    "msg": "success",
    "total": 0
}
```

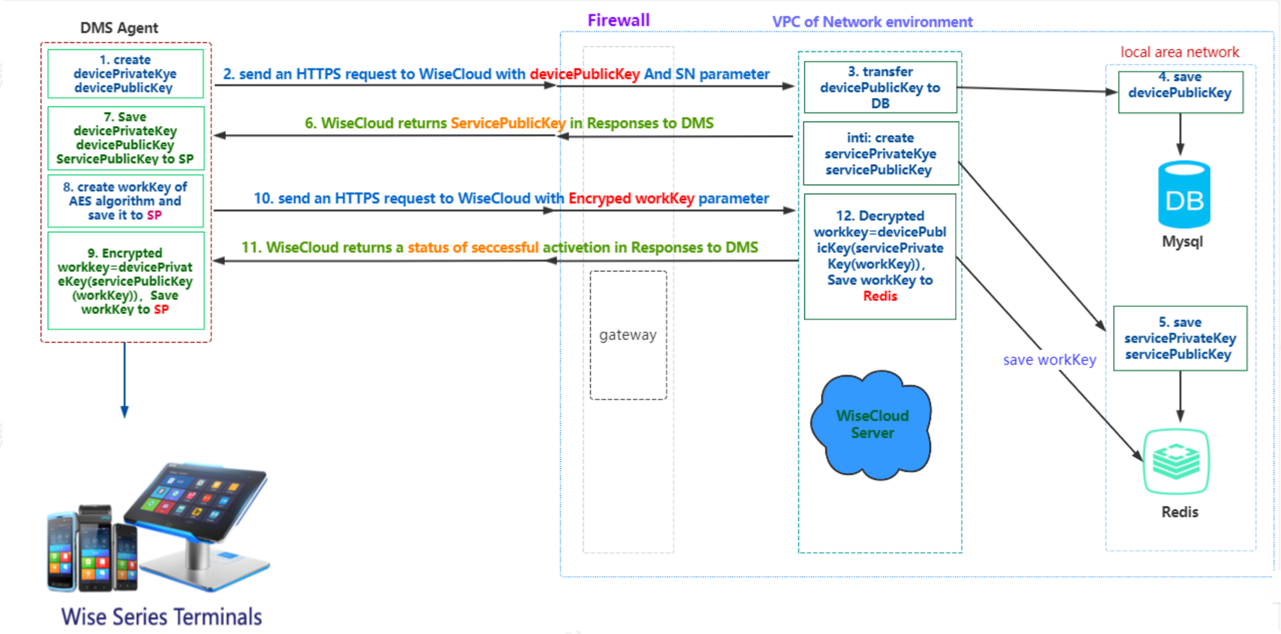
4.Communication encryption technology description

The main management functions of WiseCloud platform include: network access activation, device management, OTA upgrade, application management, etc
Communication secret key: AES (Advanced Encryption Standard in cryptography, a block encryption standard adopted by the federal government of the United States) symmetric encryption mode
Digital signature: one of the security means of data transmission in the network, which is used to prevent tampering and verify the identity of both sides.
Unique device identification: SN is the unique identification of the device in the micro smart cloud system;
The device type is wpos-3x
Work key: workkey

5、Key exchange scheme

In order to ensure the relative security of the key generated by AES, the device side and the server side will generate RSA public and private keys, and then exchange public keys. When reporting the AES key, the RSA public-private key combination algorithm will be used for encryption, which will be reported to the server, and the server will obtain the AES key through reverse decryption.

The key exchange scheme is shown in the figure


DMS Agent
1. create devicePrivateKye devicePublicKey
7. Save devicePrivateKey devicePublicKey ServicePublicKey to SP
8. create workKey of AES algorithm and save it to SP
9. Encrypted workkey=devicePrivateKey(servicePublicKey(workKey)), Save workKey to SP
2. send an HTTPS request to WiseCloud with devicePublicKey And SN parameter
6. WiseCloud returns ServicePublicKey in Responses to DMS
10. send an HTTPS request to WiseCloud with Encryped workKey parameter
11. WiseCloud returns a status of seccessful activetion in Responses to DMS
Firewall
gateway
VPC of Network environment
3. transfer devicePublicKey to DB
inti: create servicePrivateKye servicePublicKey
12. Decrypted workkey=devicePublicKey(servicePrivateKey(workKey)), Save workKey to Redis
WiseCloud Server
save workKey
local area network
4. save devicePublicKey
DB
Mysql
5. save servicePrivateKey servicePublicKey
Redis
Wise Series Terminals

6.MQTT — topic introduction

Topic Name	producer	consumer	description
INSTRUCTION/DMS/{SN}	back-up services	DMS	This topic needs to be subscribed to by the DMS. The server pushes the message to the topic, and the DMS can receive the message

7、DMS Instruction list

instruction name	instruction function	description
apkInstall	The instruction identifier of the	The DMS executes the

	push install application	instructions to install the application according to this key
uninstallApp	The instruction identifier of the push uninstall application	
OTAUpgrade	The instruction identifier of the push OTA upgrade	
animation	The instruction identifier of push boot animation	
wallpaper	The instruction identifier of push the wallpaper	
desktop	The instruction identifier of push the desktop	

8.Rules of agreement

8.1 message format

The device communicates with the micro intelligent cloud platform in JSON format.

8.2 character coding

Unified use of UTF-8 format coding

8.3 message signature and verification

Because the message transmission distinguishes whether to encrypt or not, the signature rules of encrypted message and unencrypted message are different. Details are as follows.

When the message is encrypted in the following steps: first, the message is encrypted in the following steps
When the encrypted message is signed, the encrypted message string can be signed directly;

The data message transmitted by the whole system is in JSON format, which will be signed by MD5 encryption technology before transmission. Then the MD5 is used to sign, and then the data signature is compared.

In MD5 signature, the master key workkey is required to participate in the signature. The client and server store workkey at the same time.

8.3.1 non encrypted message signature process

Step 1: establish the parameters to be signed

In the process of communication between the terminal and the background, the data message body in the message is encrypted by workkey, and the ciphertext string is obtained;

In the client request parameter list, all message nodes data of API request parameters need to participate in signature.

```
{
    "seqNo": "20200508065120000000389",
    "commandKey": "installApk",
    "downloadUrl": "http://xxx.wiseasy.com/dmr/dms/report/app_20200902.apk",
    "MD5": "abxyx2312378bds334oe0",
    "size": 1726300
}
```

Step 2: parameter sorting

The parameter name is sorted from small to large in ASCII code (sort from a to Z. if the same initial letter is encountered, see the second letter, and so on).

The array sorted in the first step is

```
{
    "commandKey": "installApk",
    "downloadUrl": "http://xxx.wiseasy.com/dmr/dms/report/app_20200902.apk",
    "MD5": "abxyx2312378bds334oe0",
  "size": 1726300,
    "seqNo": "20200508065120000000389"
}
```

Step 3: parameter splicing

Use the "&amp;" character to connect sorted parameters. The string after connecting in the previous example is as follows:

```
commandKey=installApk&downloadUrl=http://xxx.wiseasy.com/dmr/dms/report/app_20200902.apk&MD5=abxyx2312378bds334oe0&size=1726300&seqNo=2020050806512000000389
```

Step 4: sign the above string through workkey; then assign the signature string obtained from "to" signatureValue "; fill the signature string into the message, as shown in the following example

```
{
    "signType": "MD5",
    "version": "1.0",
    "isEncrypted": 1,
    "signatureValue": "ab04ccd0093aff344dco43f0",
    "data": "ciphertext(xxxxxxxxxxx)"
}
```

9. Service access

This project includes two environments: Test and production

Test environment: it is mainly used for testing and external debugging of testers;

Production environment: formal online operation environment

Service purpose	**test Service enviroment**	production environment
http request service	http://47.93.151.57:8086/	https://xx-cn.yy.com/data
MQTT Message service	url: ssl://mqtt-dev.test.com:18883 userName: test password: xxyyyy	

10.HTTPS request interface document is as follows

10.1 Device public key exchange

Interface Description: device and server exchange public key

URL: service domain name + /dms/report/devicepublickey

Request parameters:

Parameter KEY	Parameter Name	Parameter Type	Necessity	describe
devicePublicKey	device public key	TEXT	M	
sn	Device SN	TEXT	M	
deviceTypeKey	Device Type	TEXT	M	

request JSON format follow:

```
{
    "signType": "MD5",
    "version": "1.0",
    "isEncrypted": 0,
    "signatureValue": "ab04ccd0093aff344dco43f0",
    "data": {
        "devicePublicKey":"3810232asdffdd123456xxxx",
        "sn":"P320001235823",
        "deviceTypeKey":"WISELING"
    }
}
```

Response parameter:

Parameter KEY	Parameter Name	Parameter Type	Necessity	describe
code	status code	TEXT(16)	M	
msg	error message	TEXT(128)	M	
total	return the total number of data	NUMBER	M	
data	retrurn all the data	JSONArray	M	

Response data format

```
{
    "code": "0",
    "data": [],
    "msg": "success",
    "total": 0
}
```

10.2 equipment activation

Interface Description: the device reports the work key, and the device is updated to the active state after success

- URL: service domain name + dms/report/register
- request parameters:

Parameter KEY	Parameter Name	Parameter Type	Necessity	describe
deviceSn	device SN number	TEXT	M	
workKey	secret key	TEXT	M	Key to encrypt data

Workkey needs to be encrypted by RSA key. The encryption algorithm is as follows:

```
{
"topic":"wiseLing/register",
"Data": data string
}
The data string format is:
Base64 encoding (server side RSA public key encryption ({content: Base64 encoding (device RSA private key encryption ({original request data}), "csum": MD5 signature (original data)}) &amp; &amp; Base64 (deviceid)
Encryption process:
1. Assemble the original request data: {"deviceid": "xeb23cde", "workkey": "xabcde0012see5678"} to get the JSON string STR1;
2. Encrypt STR1 with device private key to get STR2;
3. Encode STR2 with Base64 to get str3. At the same time, MD5 signature is performed on the original JSON string STR1 to get sign
```

```
   = MD5 (STR1);
4. Assemble data {"content": str3, "csum": sign}, and get str4;
5. Use the public key of server to encrypt str4 to get str5;
6. Code str5 with Base64 to get str6, and Base64 to deviceid to get STR7;
7. Combine str6 and STR7, for example: str6 &amp; &amp; STR7; to get str8
8. Str8 is the value of data in the above JSON;
```

Response parameter:

Parameter KEY	Parameter Name	Parameter Type	Necessity	describe
code	status code	TEXT(16)	M	
msg	error message	TEXT(128)	M	
total	return the total number of data	NUMBER	M	
data	retrurn all the data	JSONArray	M	

10.3 reporting equipment information

Interface Description: equipment details interface

- URL: service domain name + dms/report/detail
- request parameters:

Parameter KEY	Parameter Name	Parameter Type	Necessity	describe
signatureValue	signature value	TEXT	M	
data	message text	TEXT	M	it need encryption
deviceSN	device SN	TEXT	M	
signType	signautre type	TEXT	M	md5
version	version	TEXT	M	such as :1.0
IsEncrypted	Encryption or not	TEXT	M	1 equals yes: 0equals no

Data format

Parameter KEY	Parameter Name	Parameter Type	Necessity	describe
sn	device SN	TEXT	M	
StorageCount	Size storage space	TEXT	M	
freeStoreCount	Remaining storage space	TEXT	M	
networkType	network type	int	M	network type:1、wifi，2、2G，3、3G，4、4G
signalStrength	signal intensity	int	M	
spVersion	Hardware version number	TEXT	O	
otaVersion	OTA version number	TEXT	M	
cpuInfo	CPU information	TEXT	O	

The format of push message is as follows:

```
{
    "signatureValue": "ad123456dff23d56",
    "data": "ciphertext(xxxxxxxxxxx)",
    "deviceSN": "aes12348",
    "signType": "MD5",
    "version": "1.0",
    "isEncrypted": 1
}
```

The data plaintext format is the following JSON format

```
{
  "sn":"WNET3512789000006",
  "StorageCount": 111111,
    "freeStoreCount": 971500,
    "networkType": 1,
  "signalStrength":20,
  "spVersion":"2020-04-25",
    "otaVersion": "0.0.2",
    "voiceVersion": "TTS_1.0.0"
}
```

Response parameter:

Parameter KEY	Parameter Name	Parameter Type	Necessity	describe
code	status code	TEXT(16)	M	
msg	error message	TEXT(128)	M	
total	return the total number of data	NUMBER	M	
data	retrurn all the data	JSONArray	M	

Response data format

```
{
    "code": "0",
    "data": [],
    "msg": "success",
    "total": 0
}
```

10.4 Report app information

Interface Description: report app information

- URL: service domain name + dms/report/appInfor
- request parameters:

Parameter	Parameter	Parameter	Necessity	describe

KEY	Name	Type		
signatureValue	signature value	TEXT	M	
data	message text	TEXT	M	it need encryption
deviceSN	device SN	TEXT	M	
signType	signautre type	TEXT	M	md5
version	version	TEXT	M	such as :1.0
IsEncrypted	Encryption or not	TEXT	M	1 equals yes: 0; equals no

Data format

Parameter KEY	Parameter Name	Parameter Type	Necessity	describe
packageName	packageName	TEXT	M	
appName	appName	TEXT	M	
versionNumber	versionNumber	NUMBER	M	2
versionInfo	versionInfo	TEXT	M	V_1.0.3
type	signal intensity	int	M	1:install apps; 2:runing Apps
installTime	installTime	TEXT	M	
updateTime	updateTime	TEXT	M	

The format of push message is as follows:

```
{
    "signatureValue": "ad123456dff23d56",
    "data": "ciphertext(xxxxxxxxxxx)",
    "deviceSN": "PP3526003236",
    "signType": "MD5",
    "version": "1.0",
    "isEncrypted": 1
}
```

The data plaintext format is the following JSON format

```
[{
  "packageName":"com.wiseasy.wiscashier",
  "appName": Wiscashier,
    "versionNumber": 2,
    "versionInfo": "v_1.0.2",
  "type":2,
  "installTime":"2020-04-25",
    "updateTime": "2020-08-25"
}]
```

Response parameter:

Parameter KEY	Parameter Name	Parameter Type	Necessity	describe
code	status code	TEXT(16)	M	
msg	error message	TEXT(128)	M	
total	return the total number of data	NUMBER	M	
data	retrurn all the data	JSONArray	M	

Response data format

```
{
    "code": "0",
    "data": [],
    "msg": "success",
    "total": 0
}
```

11.The server push message format is as follows（Mqtt message）

After the device is activated, mqtt is initialized and messages are subscribed. Message subject: topic ="INSTRUCTION/DMS";
Basic process: background push message, terminal receives message, terminal executes message, terminal reports message execution result

11.1 OTA upgrade message

Message format: JSON string
Function: contains a URL string, the device will download and upgrade after receiving the message.
Request parameters:

Parameter KEY	Parameter Name	Parameter Type	Necessity	describe
signatureValue	signature value	TEXT	M	
data	message text	TEXT	M	it need encryption
deviceSN	device SN	TEXT	M	
signType	signautre type	TEXT	M	md5
version	version	TEXT	M	such as :1.0
IsEncrypted	Encryption or not	TEXT	M	1 equals yes: 0; equals no

Data plaintext parameters are as follows

Parameter KEY	Parameter Name	Parameter Type	Necessity	describe
seqNo	message number	TEXT	M	
deviceTypeKey	device type	TEXT	M	
callBackUrl	Callback interface url	TEXT	O	
instructionKey	instruction key	TEXT	M	
list	message list	TEXT	M	

The format of push message is as follows:

```
{
    "signatureValue": "xxxxxxxxxxxxx",
    "data": "ciphertext(xxxxxxxxxxx)",
    "signType": "MD5",
    "version": "1.0",
    "isEncrypted": 1
}
```

data明文格式为如下JSON格式

```
{
  "seqNo": "20200508065120000000390",
    "deviceTypeKey": "WPOS-3 X",
    "callBackUrl": "http://xxx.wiseasy.com/dms/report/executeStatus",
    "instructionKey": "WISELINGOTA",
    "list":[{
        "downloadPath": "http://xxxx.wiseasy.com/dms/ota/yyyyyyy.zip",
        "filesize": 1236,
        "otaVersion": "v_1_ota_20200820",
    "otaVersionNumber": 1
    }]
}
```

11.2 app install

Message format: JSON string

Function: contains a URL string, the device will download and install after receiving the message.

Request parameters:

Parameter KEY	Parameter Name	Parameter Type	Necessity	describe
signatureValue	signature value	TEXT	M	
data	message text	TEXT	M	it need encryption
signType	signautre type	TEXT	M	md5

version	version	TEXT	M	such as :1.0
IsEncrypted	Encryption or not	TEXT	M	1 equals yes: 0; equals no

Data plaintext parameters are as follows

Parameter KEY	Parameter Name	Parameter Type	Necessity	describe
seqNo	message number	TEXT	M	
deviceTypeKey	device type	TEXT	M	
callBackUrl	Callback interface url	TEXT	O	
instructionKey	instruction key	TEXT	M	
list	message list	TEXT	M	

The format of push message is as follows:

```
{
    "signatureValue": "xxxxxxxxxxxxx",
    "data": "ciphertext(xxxxxxxxxxx)",
    "signType": "MD5",
    "version": "1.0",
    "isEncrypted": 1
}
```

data明文格式为如下JSON格式

```
{
    "seqNo": "20200508065120000000390",
    "deviceTypeKey": "WPOS-3 X",
    "callBackUrl": "http://xxx.wiseasy.com/dms/report/executeStatus",
    "instructionKey": "apkInstall",
    "list": [{
        "appName": "cashier",
        "packageName": "com.wiseasy.cashier",
        "downloadPath": "http://xxxx.wiseasy.com/dms/ota/yyyyyyy.apk",
        "filesize": 1236,
        "appVersion": "v_1.2.3",
        "appVersionNumber": 1
    }, {
        "appName": "wechat",
        "packageName": "com.wiseasy.wechat",
        "downloadPath": "http://xxxx.wiseasy.com/dms/ota/yyyyyyy.apk",
        "filesize": 1238,
        "appVersion": "v_1.2.3",
        "appVersionNumber": 1
    }]
}
```

11.3 App uninstall

Message format: JSON string

Function: After receiving the message, the device executes the unload command

Request parameters:

Parameter KEY	Parameter Name	Parameter Type	Necessity	describe
signatureValue	signature value	TEXT	M	
data	message text	TEXT	M	it need encryption
signType	signautre type	TEXT	M	md5
version	version	TEXT	M	such as :1.0
IsEncrypted	Encryption or not	TEXT	M	1 equals yes； 0 equals no

Data plaintext parameters are as follows

Parameter KEY	Parameter Name	Parameter	Necessity	describe

		Type		
seqNo	message number	TEXT	M	
deviceTypeKey	device type	TEXT	M	
callBackUrl	Callback interface url	TEXT	O	
instructionKey	instruction key	TEXT	M	
list	message list	TEXT	M	

The format of push message is as follows:

```
{
    "signatureValue": "xxxxxxxxxxxxx",
    "data": "ciphertext(xxxxxxxxxxx)",
    "signType": "MD5",
    "version": "1.0",
    "isEncrypted": 1
}
```

data明文格式为如下JSON格式

```
{
    "seqNo": "20200508065120000000390",
    "deviceTypeKey": "WPOS-3 X",
    "callBackUrl": "http://xxx.wiseasy.com/dms/report/executeStatus",
    "instructionKey": "uninstallApp",
    "list": [{
        "appName": "cashier",
        "packageName": "com.wiseasy.cashier"
    }, {
        "appName": "wechat",
        "packageName": "com.wiseasy.wechat"
    }]
}
```

11.4 Send a message with a single command (restart, restore factory settings, etc.)

Message format: JSON string

Function: After receiving the message, the device executes this command

Request parameters:

Parameter KEY	Parameter Name	Parameter Type	Necessity	describe
signatureValue	signature value	TEXT	M	
data	message text	TEXT	M	it need encryption
signType	signautre type	TEXT	M	md5
version	version	TEXT	M	such as :1.0
IsEncrypted	Encryption or not	TEXT	M	1 equals yes ； 0 equals no

Data plaintext parameters are as follows

Parameter KEY	Parameter Name	Parameter Type	Necessity	describe
seqNo	message number	TEXT	M	
deviceSN	device SN	TEXT	M	
instructionKey	instruction key	TEXT	M	

The format of push message is as follows:

```
{
    "signatureValue": "xxxxxxxxxxxxx",
    "data": "ciphertext(xxxxxxxxxxx)",
    "signType": "MD5",
    "version": "1.0",
    "isEncrypted": 1
}
```

data明文格式为如下JSON格式

```json
{
    "seqNo": "20200508065120000000390",
    "instructionKey": "pushMessage/restart/restoreSettings",
}
```