Communication Protocol

Between

Charging Station and Server

(MQTT)

**King-Meter Technology Co., Ltd.**

**July 2023 Tianjin, China**

# Revision history record

|  |  |  |
| --- | --- | --- |
| Version | Date | Description |
| V1.0.0 | 7/24/2023 | Original Version |
| V1.0.1 | 8/4/2023 | Add restart topic |
| V1.0.2 | 8/15/2023 | remove {dock\_id} from all topics if exist, add {dock\_id} into the body |
| V1.0.3 | 8/22/2023 | Add sha1 cert for upgrade function |
| V1.0.4 | 9/30/2023 | Modify querying version, remove hardware version |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# MQTT Communication protocol description

## Topic format

**Topic format : {protocol\_version}/{product\_name}/{topic}/{station\_id}**

**Data format : String, comma delimited**

**protocol\_version : current value is v1**

**product\_name: station**

**station\_id: the unique id of staiton**

**QoS of all topic is 1**

## Topic list

|  |  |
| --- | --- |
| Topic | Description |
| {protocol\_version}/station/login/#/{Id} | Log in command,the device will send this command after each connection built . |
| {protocol\_version}/station/logout/#/{Id} | Last will , once the device offline on its own initiative or not, the broker will notify the server |
| {protocol\_version}/station/sync/#/{Id} | Sync the current timestamp |
| {protocol\_version}/station/unlock/#/{Id} | Unlocking command |
| {protocol\_version}/station/force\_unlock/#/{Id} | Force unlocking command , no matter there is bike exist in the dock or not |
| {protocol\_version}/station/lock/#/{Id} | Locking command , no matter there is bike exist in the dock or not |
| {protocol\_version}/station/card\_permission/#/{Id} | unlock by rfid card, request permission from server |
| {protocol\_version}/station/card\_notify/#/{Id} | unlock by rfid card,report result to server. |
| {protocol\_version}/station/bike\_in/#/{Id} | Somebody push the bike into the dock,the station would notify the server |
| {protocol\_version}/station/info/#/{Id} | Report info of station |
| {protocol\_version}/station/malfunction/#/{Id} | Report malfunction of station |
| {protocol\_version}/station/clear\_malfunction/#/{Id} | Clear malfunction |
| {protocol\_version}/station/check/#/{Id} | Server get info of station |
| {protocol\_version}/station/setting/#/{Id} | Server configure the station |
| {protocol\_version}/station/version/#/{Id} | Server query version of all components |
| {protocol\_version}/station/upgrade/#/{Id} | Server send ota command |
| {protocol\_version}/station/upgrade\_result/#/{Id} | Station report ota result |

## Common parameters

|  |  |
| --- | --- |
| name | description |
| station\_id | The unique id of station |
| dock\_id | The unique id of dock , several dock compose one station |
| bike\_id | The unique id of e-bike,when the bike is pushed into the dock ,the dock would recognize the id of the e-bike |
| company\_code | The code for different company, default as 2113, length 4, Integer type |
| password | Password for station to login to server, Integer value,length 6,range from 100000 to 999999,default as 100419 |
| hardware\_version | The version of hardware |
| software\_version | The version of software, important for OTA function |
| timestamp | UTC timestamp. The station keep sync with server on this value. |
| secret\_key | When station login, the server would create one randomly for the station , after that the station and server would communicate with each other with this value , avoid illegal station or server,  String value ,length depend on server design  If secret\_key is not correct, the other side would not reply anything |
| card\_no | The number of RFID card , length 8 , range from 00000000 to FFFFFFFF |
| user\_id | The unique id of user from server side,String value, the length depends on the design of server |
| sequence\_number | Set it with current timestamp, and keep both side with same logic process . |
| expire\_time | In some cases,such as unlocking, server send this value to station,the station would verify whether this command is expired or not . |
| timeout | Take unlocking for example,server send this value to station,the station make one dock unlocked,if the customer hasn’t get out of the bike within the require timeout,the dock would lock the bike itself. |
| min\_soc | When you try to unlock the dock, the capacity of bike battery should be larger than this value |

## Common Result

Every time the server send command to the wifimaster,if the wifimaster need to reply, there must be one parameter named “result” included in the payload

|  |  |
| --- | --- |
| name | description |
| 0 | command valid |
| 1 | command expired |
| 2 | dock id not exist |

## Topics about login

### Login

#### Request

Topic: v1/station/login/request/{station\_id}

Publisher: station

Payload: {company\_code},{password},{hardware\_version},{software\_version}

Description: Once the station power on,the station will publish this topic

Example:

Topic : v1/station/login/9000000000001

Payload: 2113,100419,20230724,20230724

#### Reply

Topic: v1/station/login/reply/{station\_id}

Publisher: server

Payload: {result},{secret\_key},{timestamp}

Description: secret\_key is created by server,it is used by some important topics, such as unlocking,locking ,swing card to unlock

Example:

Topic: v1/station/login/reply/9000000000001

Payload: 0,abcd,1691028779

### Logout (LWT last will and Testament)

Topic: v1/station/logout/{station\_id}

Publisher: station

Payload: None

Description: the station set topic of LWT when it try to connect to the broker

Example:

Topic: v1/station/logout/9000000000001

Payload:

## Topics about business logic

### Sync timestamp every 10 minutes

#### Station Request

Topic: v1/station/sync/request/{station\_id}

Publisher: station

Payload: {secret\_key},{device\_send\_timestamp}

Description: this topic is used for sync timestamp with server side, the station will publish this topic once it get login response,and will publish it every 10 minutes later.

If server doesn’t reply anything within 3 minutes , the station would restart,publish login topic again .

Example:

Topic: v1/station/sync/request/9000000000001

Payload: 123,1691028779

#### Server Reply

Topic: v1/station/sync/reply/{station\_id}

Publisher: server

Payload: {secret\_key},{device\_send\_timestamp},{server\_receive\_timestamp},{server\_send\_timestamp}

Description:

Example:

Topic: v1/station/sync/reply/9000000000001

Payload: 123,1691028779,1691028782,1691028785

device\_receive\_timestamp means the time when station receive this topic

So the station would calculate the real current time by the following equation:

Current\_Time = (server\_receive\_timestamp + server\_send\_timestamp

+ device\_receive\_timestamp - device\_send\_timestamp)/2

The station would set Current\_Time as system time itself.

### Customer Unlocking the dock

#### Server Request

Topic: v1/station/unlock/request/{station\_id}

Publisher: server

Payload:

{secret\_key},{dock\_id}，{user\_id},{min\_soc},{sequence\_number},{expire\_time},{timeout}

Description:

expire\_time: expire time , second unit ,the station

if (sequence\_number + expire\_time) < current\_device\_time , then station will reply result value with 3

Example:

Topic: v1/station/unlock/request/9000000000001

Payload: 123,1000000000001,2341314,10,1691028782,10,20

#### Station Reply

Topic: v1/station/unlock/reply/{station\_id}

Publisher: station

Payload: {secret\_key},{dock\_id},{user\_id},{bike\_id},{sequence\_number},{result},{rent\_status},{bike\_status},{lock\_status}

Description:

result: 0 : action correct

1: command expired

2: dock id not exist

rent\_status: 0: no bike

1: rent succeed

18: capacity of battery of bike is too low to rent

19: rent failed,unlock failed

bike\_status: 0: no bike exist in the dock now

1: bike exist in the dock now(maybe the user didn’t pull out the bike right now)

lock\_status : 0: unlocked

1: locked

3: unlock failed

Example:

Topic: v1/station/unlock/request/9000000000001

Payload: 123,1000000000001,2341314,2000000000001,1691028782,0,1,0,0

### Operator Force Unlocking the dock

#### Server Request

Topic: v1/station/force\_unlock/request/{station\_id}

Publisher: server

Payload: {secret\_key},{dock\_id},{user\_id},{sequence\_number},{expire\_time},{timeout}

Description:

expire\_time: expire time , second unit ,the station

if (sequence\_number + expire\_time) < current\_device\_time , then station will reply result value with 3

Example:

Topic: v1/station/force\_unlock/request/9000000000001

Payload: 123,1000000000001,2341314,1691028782,10,20

#### Station reply

Topic: v1/station/force\_unlock/reply/{station\_id}

Publisher: station

Payload: {secret\_key},{dock\_id},{user\_id},{bike\_id},{sequence\_number},{result}

Description:

result: 0: unlock succeed;

1: unlock failed;

2: secret validated failed

3: command expired

Example:

Topic: v1/station/force\_unlock/reply/9000000000001/1000000000001

Payload: 123,2341314,2000000000001,1691028782,0

### Locking Request

#### Server Request

Topic: v1/station/lock/request/{station\_id}

Publisher: server

Payload: {secret\_key},{dock\_id},{user\_id},{sequence\_number},{expire\_time}

Description:

expire\_time: expire time , second unit ,the station

if (sequence\_number + expire\_time) < current\_device\_time , then station will reply result value with 3

Example:

Topic: v1/station/lock/request/9000000000001

Payload: 123,1000000000001,2341314,1691028782,10

#### Station reply

Topic: v1/station/lock/reply/{station\_id}

Publisher: station

Payload: {secret\_key},{dock\_id},{user\_id},{bike\_id},{sequence\_number},{result}

Description:

result: 0: lock succeed;

1: lock failed;

2: secret validated failed

3: command expired

Example:

Topic: v1/station/lock/request/9000000000001

Payload: 123,1000000000001,2341314,2000000000001,1691028782,0

### Swing RFID card,request permission of unlocking

#### Station Request

Topic: v1/station/card\_permission/request/{station\_id}

Publisher: station

Payload: {secret\_key},{dock\_id},{card\_no},{bike\_id},{sequence\_number}

Description:

sequence\_number equal device\_send\_timestamp here

Example:

Topic: v1/station/card/request/9000000000001

Payload: 123,1000000000001,AE34DF78,2000000000001,1691028782

#### Server reply

Topic: v1/station/card\_permission/reply/{station\_id}

Publisher: server

Payload: {secret\_key},{dock\_id},{user\_status},{user\_balance},{min\_soc},{sequence\_number},{expire\_time},{timeout},{result}

Description:

user\_status: account status

0: account ok

11: get account info failed

17: the balance of user is not enough to rent bike

21: this RFID card has been frozen

23: no bike in this dock , only query info of user account by this card ,will show some info on the display of this dock .

user\_balance: balance of the user which bond this rfid card,unit 0.01 $

sequence\_number: same as the value from 2.5.5.1

result: 0: server allow unlocking by this RFID card;

2: secret validated failed

3: command expired

Example:

Topic: v1/station/card\_permission/request/9000000000001/1000000000001

Payload: 123,0,199,10,1691028782,10,10,0

### Unlocked notify after swinging card

#### Station Request

Topic: v1/station/card\_notify/request/{station\_id}

Publisher: station

Payload: {secret\_key},{card\_no},{bike\_id},{sequence\_number},{result}

Description:

sequence\_number: same as the value from 2.5.5.1

result: 0: unlock succeed

29: low battery,cannot rent this bike

If the user hasn’t get out this bike , the dock would try to lock itself, if lock succeed, the station would publish bike\_in\_dock topic

Example:

Topic: v1/station/unlock/request/9000000000001

Payload: 123,1000000000001,AE34DF78,2000000000001,1691028782,0

#### Server reply

Topic: v1/station/card\_notify/reply/{station\_id}

Publisher: server

Payload: {secret\_key},{dock\_id},{sequence\_number}

Description:

sequence\_number: same as the value from 2.5.5.1

Example:

Topic: v1/station/unlock/request/9000000000001/1000000000001

Payload: 123,1691028782

### Bike in Dock notify

#### Station Request

Topic: v1/station/bike\_in/request/{station\_id}

Publisher: station

Payload: {secret\_key},{dock\_id},{bike\_id},{return\_state},{sequence\_number}

Description:

rent\_state : 0 return succeed(locked succeed)

1 return failed(locked failed)

Example:

Topic: v1/station/bike\_in/request/9000000000001

Payload: 123,1000000000001,2000000000001,0,1691028782

#### Server reply

Topic: v1/station/bike\_in/reply/{station\_id}

Publisher: server

Payload: {secret\_key},{dock\_id},{user\_balance},{consumption\_money},{result}

Description:

result : 0:user account ok

1: user balance back charge

Example:

Topic: v1/station/bike\_in/request/9000000000001

Payload: 123,1000000000001,180,19,0

## Topics about other things

### Report information of station every 5 minutes

#### Station Request

Topic: v1/station/info/request/{station\_id}

Publisher: station

Payload: {secret\_key}#{dock\_id},{dock\_sort},{bike\_id},{bike\_soc};{dock\_id},{dock\_sort},{bike\_id},{bike\_soc};{dock\_id},{bike\_id},{bike\_soc};

Description:

The amount of {dock\_id},{dock\_sort},{bike\_id},{bike\_soc}; depends on how many docks this station manage.

Example:

Topic: v1/station/info/request/9000000000001

Payload: 123#1000000000001,1,2000000000001,56;1000000000002,2,2000000000002,15;1000000000003,2000000000003,100;

#### Server reply

Topic: v1/station/info/reply/{station\_id}

Publisher: server

Payload: {secret\_key}

Description:

Example:

Topic: v1/station/info/reply/9000000000001

Payload: 123

### Query infomation of station remotely

#### Server Request

Topic: v1/station/query\_info/request/{station\_id}

Publisher: server

Payload: {secret\_key}

Description:

the server can also query information actively.no need to wait the station update info by 2.7.1.1

Example:

Topic: v1/station/query\_info/request/9000000000001

Payload: 123

#### Station reply

Topic: v1/station/query\_info/reply/{station\_id}

Publisher: station

Payload: {secret\_key}#{dock\_id},{dock\_sort},{bike\_id},{bike\_soc};{dock\_id},{dock\_sort},{bike\_id},{bike\_soc};{dock\_id},{bike\_id},{bike\_soc};

Description:

The amount of {dock\_id},{dock\_sort},{bike\_id},{bike\_soc}; depends on how many docks this station manage.

Example:

Topic: v1/station/query\_info/reply/9000000000001

Payload: 123#1000000000001,1,2000000000001,56;1000000000002,2,2000000000002,15;1000000000003,2000000000003,100;

### Report malfunction when something happen wrong or recover

#### Station Request

Topic: v1/station/malfunction/request/{station\_id}

Publisher: station

Payload: {secret\_key},{dock\_id},{dock\_error},{dock\_lock\_error},{charging\_error},{bms\_error},{battery\_error},{display\_error};

Description:

dock\_error:

dock\_lock\_error:

charging\_error:

bms\_error:

battery\_error:

display\_error:

Example:

Topic: v1/station/malfunction/request/9000000000001

Payload: 123,1000000000001,0,0,0,0,0,0

#### Server reply

Topic: v1/station/malfunction/reply/{station\_id}

Publisher: station

Payload: {secret\_key}

Description:

Example:

Topic: v1/station/malfunction/reply/9000000000001

Payload: 123

### Server clear malfunction

#### Server Request

Topic: v1/station/clear\_malfunction/request/{station\_id}

Publisher: server

Payload: {secret\_key},{dock\_id},{error\_code}

Description:

error\_code: 21(carrier error) ,12(lock error), 13(unlock error)

Example:

Topic: v1/station/clear\_malfunction/request/9000000000001

Payload: 123,1000000000001,21

#### Station reply

Topic: v1/station/clear\_malfunction/reply/{station\_id}

Publisher: station

Payload: {secret\_key},{dock\_id},{result}

Description:

Example:

Topic: v1/station/malfunction/reply/9000000000001

Payload: 123,1000000000001,0

### Sync version of all components of station

#### Server Request

Topic: v1/station/version\_async/request/{station\_id}

Publisher: server

Payload: {secret\_key},{component\_type}

Description:

component\_type: 1: wifimaster

2: icpu

3: display

4: eicc

5: font

6: pcrd

Example:

Topic: v1/station/version/request/9000000000001

Payload: 123,1

#### Station reply

Topic: v1/station/version\_async/reply/{station\_id}

Publisher: station

Payload: {secret\_key},{component\_type}

Description:

Example:

Topic: v1/station/version\_async/reply/9000000000001

Payload: 123,1

### Query version of all components of station

#### Server Request

Topic: v1/station/version/request/{station\_id}

Publisher: server

Payload: {secret\_key},{component\_type}

Description:

component\_type: 1: wifimaster

2: icpu

3: display

4: eicc

5: font

6: pcrd

Example:

Topic: v1/station/version/request/9000000000001

Payload: 123,1

#### Station reply

Topic: v1/station/version/reply/{station\_id}

Publisher: station

Payload:

the format depends on the component\_type

wifimaster: {secret\_key},{component\_type};{wifimaster\_sw},{wifimaster\_hw};

pcrd: {secret\_key},{component\_type};{pcrd\_sw},{pcrd\_hw};

icpu: {secret\_key},{component\_type};{dock\_id},{icpu\_sw},{icpu\_hw};...

display:{secret\_key},{component\_type};{dock\_id},{display\_sw},{display\_hw};...

eicc:{secret\_key},{component\_type};{dock\_id},{eicc\_sw},{eicc\_hw};...

font:{secret\_key},{component\_type};{dock\_id},{font\_sw},{font\_hw};...

Description:

Example:

Topic: v1/station/version/reply/9000000000001

Payload:

wifimaster:123,20230724;20201201,20180912;

icpu:(if there are 3 docks)

123,20230724;1000000000001,20200101,20200202;1000000000002,20200101,20200202;1000000000003,20200101,20200202;

### Setting configuration of station

#### Server Request

Topic: v1/station/setting/request/{station\_id}

Publisher: server

Payload: {secret\_key},{new\_secret\_key},{sync\_timestamp}

Description:

Example:

Topic: v1/station/setting/request/9000000000001

Payload: 123

#### Station reply

Topic: v1/station/setting/reply/{station\_id}

Publisher: station

Payload: {secret\_key},{dock\_id},{result}

Description:

result: 0 : succeed

1: failed

Example:

Topic: v1/station/setting/reply/9000000000001

Payload: 123,1000000000001,0

### Server restart station

#### Server Request

Topic: v1/station/restart/request/{station\_id}

Publisher: server

Payload: {secret\_key}

Description:

Example:

Topic: v1/station/restart/request/9000000000001

Payload: 123

#### Station reply

Topic: v1/station/restart/reply/{station\_id}

Publisher: station

Payload: {secret\_key},{dock\_id}

Description: the station publish this topic, and restart itself,connect mqtt broker, then public login topic .

Example:

Topic: v1/station/restart/reply/9000000000001

Payload:

## Topics of OTA

### Upgrading component of station

#### Server Request

Topic: v1/station/upgrade/request/{station\_id}

Publisher: server

Payload: {secret\_key},{component\_type},{firmware\_url},{SHA1\_Cert}

Description:

Example:

Topic: v1/station/upgrade/request/9000000000001

Payload: 123,1,http:\*\*\*\*\*\*\*\*,333

Enum of component\_type:

1: wifimaster

2: ipcu

3: display

4: eicc

5: font

6: pcrd

#### Station reply

Topic: v1/station/upgrade/reply/{station\_id}

Publisher: station

Payload: {secret\_key},{component\_type},{result}

Description:

Example:

Topic: v1/station/upgrade/reply/9000000000001

Payload: 123,1,0

Enum of result:

1: download file failed

2: download file succeed,but verify by sha1 failed

3: download file succeed,and verify by sha1 succeed,try to upgrade later

4: download file succeed,and verify by sha1 succeed,but fond none of compoents to be ugpraded

### Notify upgrading process

#### Station Request

Topic: v1/station/upgrade\_result/request/{station\_id}

Publisher: station

Payload: {secret\_key},{component\_type},{dock\_id},{result}

Description:

wifimaster upgarde itself would not publish this topic

Enum of component\_type:

2: ipcu

3: display

4: eicc

5: font

6: pcrd

Example:

Topic: v1/station/upgrade\_result/request/9000000000001

Payload: 123,1,1000000000001,1

Enum of result:

1, succeed

2, none of this type of component is upgraded

#### Station reply

Topic: v1/station/upgrade\_result/reply/{station\_id}

Publisher: station

Payload: {secret\_key},{component\_type},{dock\_id},{result}

Description:

Example:

Topic: v1/station/upgrade\_result/reply/9000000000001

Payload: 123

## Appendix

# Bluetooth Communication protocol description

## //todo , you could config the ip and port of the mqtt broker by App through bluetooth.

## Instruction format

## Data encryption

## Communication process

## Characteristic under the service

## Command details and examples

## Appendix 1