1. **Communication Protocol Contents**

1.1 The nomenclature used in this document

|  |  |
| --- | --- |
| Time Attendance terminal | The time attendance terminal which will link the internet and the HTTP protocol |
| Operator | The one who send instruction to the specific machine  The operator send instruction and wait for the result through Web APP on the PC |
| The operator instructions | The instructions which was send by the operator  Eg. Set time and change user’s ID |
| Register data | The data on the machine which is used to identify the user, eg. Fingerprint data, password, ID number, facial data and so on. |

1.2 Base on the HTTP protocol and server to communication the machine.

Machine used HTTP request in the POST way to send all the requests to the server.

The responding part of this POST request HTTP contains binary data.

1.3 The format of binary data in request and responding

Following is the format of binary data.

The binary data in the subsequent place and the string data in front of it.

String data belongs to the one in the format of JSON with its code UTF-8.

Actually,it can express all the data format by using JSON. However,JSOM may cause some problems, like the capacity of the whole data,speed of communication and so on.

In this communication protocol, we do take advantage of binary to express data under the circumstance of using JSON with least adverse.

the front string data will be signed when using the binary data, which is corresponding to the later one.

Such as, if the type of a syllable is binary data, it will be instead by Bin\_n.

N is the serial no of later binary data. It starts from 1.

For instance, “log\_array”:BIN\_1

This JOSN string data instructions the recorded one belonging to binary data and put at the first place after the string data.

1.4 Communicating process of machine and its server

Communication with WEB server roughly divided into two kinds. One is the process that machine to receive and implement instruction

instructions. The other is the process that machine notifies some events of server, such as the generation of a new record.

2.The general process of operator instructions

The process of operator instructions is that users choose the attendance machine needed to be managed by log in server. The server will distribute the instruction to attendance machine and accept the outcome of instructions. The process is as follow.

1)Operators select attendance machine and get the machine number(device\_id).

2)Operator on the WEB sever linkage database maintain records of instruction executed on the machine.

Those records include below messages.

Task identification number(trans\_id),

Machine identification number(device\_id)

Order identification number(cmd\_code)

Directive parameter data(cmd\_param)

Task status(trans\_status)

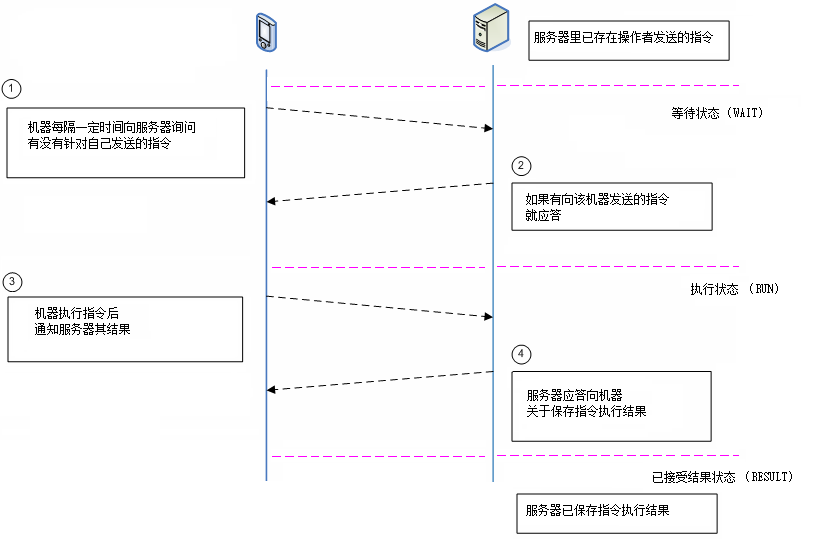
Task status last updated time(trans\_status\_update\_time)

1)Machine will ask the server at regular intervals if there are some instructions for it. If yes, bring it to execute and upload the output to Web server.

2)Operator will ask the executive status of service instruction at regular intervals. If there are executed identifications, they deal with the results.

This trans\_idis is the task identification number.Namely when returning the instruction execution result, use trans\_id to judge which instruction this result corresponding to.

Chart 2-1 is the process.



If there is any instruction, sent to the machine

The server responds to the device about the result of the save instruction

The machine feedback the result to the server after executing the instruction

Device ask server if there is any instruction for at set intervals

Server stored instruction device will send

Server saves execution results

P2-1.The communication process in the command processing.

The command was send by the operator has been existence in the server

1. The terminal will ask the server whether the sever send command to itself or not at regular intervals

2. If there is a command for the terminal,it will reply

3. After finish the command,the terminal will give notice to the server

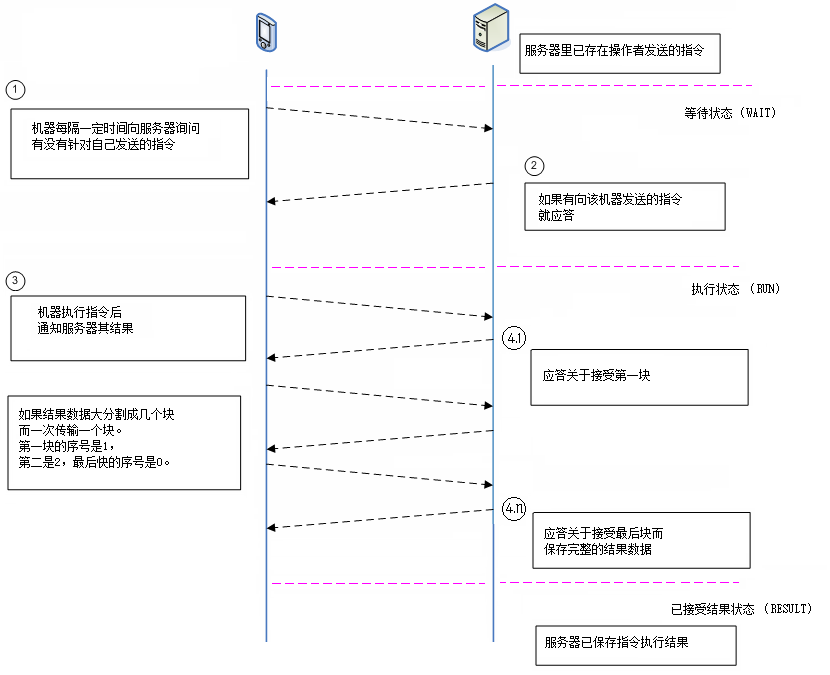
4.The server responds to the device about the result of the save instruction

If the result is more than 10KB when the machine uploading instruction result, it may be uploaded in the way of having been divided into several parts.

At this time, the machine will separate data into several blocks and then transmit. The server receives those blocks by order and stores in a temporary buffer.

If the serial number of the last block, server received, is zero,it will joint the before receiving result together to the completed data and store at the database.

Such process as shown in figure 2-2



Server saves execution results

Server stored instruction device will send

To answer the last block and save the complete result data

If the result data is divided into several blocks, one block is transmitted every time. The first block serial number is 1, the number is 2, and the last block is the serial number of 0

Responding to device for receiving first block

The machine feedback the result to the server after executing the instruction

If there is any instruction, sent to the machine

Device ask server if there is any instruction for at set intervals

2-2. If the result data is large, it can be divided into several blocks and one block at a time

Received instruction and execution process are similar to the format of HTTP request and response in most parts but there still exist some difference on instruction.

2.1 Request and response needed for attendance machine to receive operators instructions Attendance machine will send HTTP request to WEB server at regular intervals and receive response in purpose of receiving instructions. Following is the format in details.

**2.1.1 Request for machine to receive operator instruction**

Attendance machine would send out signal to the server for instructions.

HTTP POST request to WEB server at regular intervals.

The following field will be put within the HTTP header.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field name** | **Field meaning** | **Necessary options** | **restrictions** | **Detailed instructions** |
| request\_code | Demand\_code | must | Must be a string as follow “receive\_cmd” | Indicating the machine asks WEB server for instruction aiming at itself. |
| dev\_id | Identification number for machine | must | The number of words, maximum 24bits | All the attendance machine having connected to the same WEB server, must have a unique identification number.  This parameter refers to the attendance machine’s unique number. |
| Content-type | MIME type | must | Must be a string as follow  “application/octet-stream” | Content-Type generally refers to the one existing in web page,which is used to define the type of network and web page code,determin what form the browser will be in and what code to read the file. Those are the reasons why the result often seen on some Asp web pages by clicking is downloaded to a file and a picture. |
| Content-Length | The length of message transmission | must | number | Content-Length must comply with the transmission length of message exactly. |

For example the below HTTP header is uploaded to the server from machine.

|  |
| --- |
| **POST / HTTP/1.0**  Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/vnd.ms-excel, application/msword, application/vnd.ms-powerpoint, \*/\*  Accept-Language: en-us  Accept-Encoding: gzip, deflate  User-Agent: Mozilla/4.0  Connection: close  **Content-Type: application/octet-stream**  **request\_code: receive\_cmd**  **dev\_id: 001**  **Content-Length: 201** |

The bold words in front was used to mark the field that shall be noted.

The data described in 1.3 was placed in the portion of HTTP body.

The content of string in body part is as below.

{

“fk\_name”:<1>,

“fk\_time”:<2>,

“fk\_info”:

{

"supported\_enroll\_data":<3.1>,

"fk\_bin\_data\_lib":<3.2>,

"firmware":<3.3>

}

}

fk\_name field ：the name of machine

fk\_time field ：submit the the time of machine, HTTP request.The form of time string is YYMMDDhhmmss。

fk\_info field ：the machine information.

supported\_enroll\_data field：the type of registrating data used by machine. For instance, if use fingerprint data,you shall mark FP.

fk\_bin\_data\_libfield： explaining dynamic library name used when uploading binary data.  
Such as, FKDataHS001 means that if there is a need of explaining binary data, name FKDataHS001 database will be used.

firmwarefield：firmware version of the machine.

For instance,

|  |
| --- |
| {  "supported\_enroll\_data":["FP","PASSWORD","IDCARD","FACE"],  "fk\_bin\_data\_lib":"FKDataHS001",  "firmware":"FK725HS001"  } |

No binary data will be put into the HTTP body when summit the HTTP request.

**2.1.2 Server’s response to receive instruction request**

The server will check if there are some instructions aiming at the machine itself, after having received the request.

If yes, the server will download the response.

It contains below information in response header and response body.

Response header includes following field.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field name** | **field meaning** | **Necessary options** | **restriction** | **Detailed instructions** |
| response\_code | Response code | must | Words’ number, maximum 64bits  The capital form of English letter | The result of receiving instructions  OK : success  ERROR : failure |
| trans\_id | Task recognition number | optional | Words’ number, maximum 16bits。 | Identified number of tracking the instruction executed process |
| cmd\_code | Order identification number | optional | ords’ number, maximum 32bits  The capital form of English letter | Indicating the type of instructions  Such as： GET\_ENROLL\_DATA |
| Content-type | MIME type | must | Must be below string “application/octet-stream” | Content-Type general indicates the one existing in web page, which is used to define the type of network files and web page code, decide what form the browser will be in and the code to read the file.Those are the reasons why some results often seem clicked in many Asp web page but finally they were downloaded to a file or a picture. |
| Content-Length | Transmission length of message | must | number | If exist and valid, it must keep the same length with the message. |

For example, the below response is downloaded.

|  |
| --- |
| **HTTP/1.1 200 OK**  Cache-Control: private  Server: Microsoft-IIS/7.5  Set-Cookie: ASP.NET\_SessionId=453lmc45jaelft45glb2mdre; path=/; HttpOnly  X-AspNet-Version: 2.0.50727  X-Powered-By: ASP.NET  Date: Wed, 10 Dec 2014 04:47:42 GMT  Connection: close  **Content-Length: 39**  **Content-Type: application/octet-stream**  **response\_code: OK**  **trans\_id: 201**  **cmd\_code: GET\_ENROLL\_DATA** |

Data in response body is different according to orders.

**2.2 The need of request and response for uploading instruction executive result**

After having executed instruction, machine will send HTTP request, receive response and transmit the result to WEB server. Below is the detailed format.

**2.2.1 Uploading the request needed for operator to execute instruction**

Response header includes following field.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field name** | **Field meaning** | **Necessary option** | **restriction** | **Detailed instruction** |
| request\_code | Request code | must | Must be the following string.  “send\_cmd\_result” | Indicating the executive result for machine uploading instruction to server |
| dev\_id | Machine identifing code | must | Words’number, the maximum 24 bits | Reference 2.1.1 |
| trans\_id | Task identification number | must | Words’number, the maximum 16bits | Reference 2.1.2 |
| cmd\_return\_code | Instructive result code | must | Words’number, the maximum 64bits | Indicating the result fr machine to carry out instruction  OK means success  If there is a string like ERROR coming out on the process of execution ,it means the error string. |
| blk\_no | Block number | optional | number | The serial number of block machine sending  If the result data was divided into several blocks and transmit one block each time, the serial number of first block is 1,the second is 2 and the last one is 0. The size of block data is consistent with content\_length string. |
| Content-type | MIME type | must | Must be below string  “application/octet-stream” | Content-Type generally indicates to the the one existing in web page, which is used to define the type of network files and the code number of web page. It decides what form the browser will be in and what code number to read this file. Those are the reasons why the result constantly seen in some Asp web page through clicking is downloaded to a file or a picture. |
| Content-Length | Transmission length of message | optional | Number | Content-Length must be consistent with the transmission length of message completely. |

Data in the request nody is different according to the instruction.

**2.2.2 The servers response to the request for the machine to upload the result**

The server saves the data to database, executes the instruction and then download below response.

Below field will be put in the response header.

|  |  |  |  |
| --- | --- | --- | --- |
| **Field name** | **Necessary option** | **Restrictions** | **Detailed instruction** |
| response\_code | Must | Words number,the maximum is 64bits.  All the English letters are capital. | Indicating server having received and saved the result data successfully  OK : success  ERROR : failure |
| trans\_id | Must | Words number, the maximum is 16bits. | Reference2.1.2 |

**3. Operator Command**

The operator sends the following commends to terminal.

|  |  |
| --- | --- |
| Command Name | Command Code |
| [Get Terminal Enrollment Data](#bkm_3_1) | GET\_ENROLL\_DATA |
| [Set Database Enroll Data To Terminal](#bkm_3_2) | SET\_ENROLL\_DATA |
| Set Synchronization Time | SET\_TIME |
| Reset Terminal | RESET\_FK |
| Delete user | DELETE\_USER |
| Rename | SET\_USER\_NAME |
| Change User Privilege | SET\_USER\_PRIVILEGE |
| [Ge](#bkm_3_10)t Enroll ID List | GET\_USER\_ID\_LIST |
| Get Log Data | GET\_LOG\_DATA |
| Set Terminal name | SET\_FK\_NAME |
| [Clear](#bkm_3_14) Log Data | CLEAR\_LOG\_DATA |
| [Clear](#bkm_3_15) Enroll Data | CLEAR\_ENROLL\_DATA |
| Get Terminal Status | GET\_DEVICE\_STATUS |
| Set the user enroll data and information | SET\_USER\_INFO |
| Get the user information from terminal | GET\_USER\_INFO |
| Set the server address and port NO. | SET\_WEB\_SERVER\_INFO |

**3.1. Get Terminal Enrollment Data(GET\_ENROLL\_DATA)**

|  |  |  |
| --- | --- | --- |
| **Terminal Request** |  | **WEB Server Response** |
| -- HTTP header --  request\_code:receive\_cmd  dev\_id:<2>  -- HTTP body --  Means the string of the terminal information |  |  |
|  |  | -- HTTP header --  response\_code:<1>  trans\_id:<2>  cmd\_code:GET\_ENROLL\_DATA  -- HTTP body --  {<4>}  <4>Format  {"user\_id":"<4.1>","backup\_number":<4.2>}  user\_id field: The register NO. Of the user  backup\_number field means the NO. Of the register data form.  Set one of the following value  0 ~ 9 : Ten fingers’ data for user  10 : password for user  11 : ID No. For user  12 : Facial data for user |
| -- HTTP header –  request\_code:send\_cmd\_result  dev\_id:<2>  trans\_id:<3>  cmd\_return\_code:<4>  blk\_no:<5>  Value of <4> can be set as follow  OK : Get the use data successful  EEROR\_NOT\_EXIST : In command parameter, the designated registration data isn’t be enrolled.  There is no data in HTTP body part.  <5> means serial number of each part when the result data is divided into multi-parts .  The size of execute results data of the register data command according to fingerprint, password and facial are different, some times it will up to 20KB.  So sometimes need to cut into different part to transmit.  -- HTTP body –-  The following format of the result data is divided into multi-parts, then transmit.  <6> + bin\_1  {"enroll\_data":"BIN\_1"} |  |  |
|  |  | -- HTTP header --  response\_code:<1>  trans\_id:<2> |

**3.2. Set Database Enroll Data To Terminal(SET\_ENROLL\_DATA)**

Request to set database enrollment data to terminal

|  |  |  |
| --- | --- | --- |
| **Terminal Request** |  | **WEB Server Response** |
| -- HTTP header --  request\_code:receive\_cmd  dev\_id:<2>  -- HTTP body --  Means the string of the terminal information |  |  |
|  |  | -- HTTP header --  response\_code:<1>  trans\_id:<2>  cmd\_code: SET\_ENROLL\_DATA  -- HTTP body --  {<4>} + bin\_1  Format of <4>  {  "user\_id":"<4.1>",  "backup\_number":<4.2>,  "enroll\_data":"BIN\_1"  }  The meaning and the format of each field, please check 3.1。 |
| -- HTTP header --  request\_code: send\_cmd\_result  dev\_id:<2>  trans\_id:<3>  cmd\_return\_code:<4>  blk\_no:<5>  -- HTTP body --  For this command , the execute results with no any data, so there is no any data in body part. |  |  |
|  |  | -- HTTP header --  response\_code:<1>  trans\_id:<2> |

**3.3. Set Synchronization Time(SET\_TIME)**

|  |  |  |
| --- | --- | --- |
| **Terminal Request** |  | **WEB Server Response** |
| -- HTTP header --  request\_code:receive\_cmd  dev\_id:<2>  -- HTTP body --  Means the string of the terminal information |  |  | |
|  |  | -- HTTP header --  response\_code:<1>  trans\_id:<2>  cmd\_code:SET\_TIME  -- HTTP body --  {<4>}  The format of <4>  {"time":"<4.1>"}  Put <4.1> into the time string of the server, the format is YYYYMMDDhhmmss。 | |
| -- HTTP header --  request\_code: send\_cmd\_result  dev\_id:<2>  trans\_id:<3>  cmd\_return\_code:<4>  blk\_no:<5>  -- HTTP body --  For this command , the execute results with no any data, so there is no any data in body part. |  |  | |
|  |  | -- HTTP header --  response\_code:<1>  trans\_id:<2> | |

**3.4.** **Reset the terminal (RESET\_FK)**

Please send this command if you want to reset the terminal due to certain reasons.

For example, a certain command status is always under {RUN} status.

If can’t confirm the reason, reset terminal is the best solution for you.

After reset the terminal, the terminal ignores all previous commands, and waiting to receive new command.

|  |  |  |
| --- | --- | --- |
| **Terminal Request** |  | **WEB Server Response** |
| -- HTTP header --  request\_code:receive\_cmd  dev\_id:<2>  -- HTTP body --  Means the string of the terminal information |  |  |
|  |  | -- HTTP header --  response\_code:RESET\_FK  trans\_id:<2> |

**3.5. Delete the user (DELETE\_USER)**

|  |  |  |
| --- | --- | --- |
| **Terminal Request** |  | **WEB Server Response** |
| -- HTTP header --  request\_code:receive\_cmd  dev\_id:<2>  -- HTTP body --  Means the string of the terminal information |  |  |
|  |  | -- HTTP header --  response\_code:<1>  trans\_id:<2>  cmd\_code:DELETE\_USER  -- HTTP body --  {<4>}  The format of <4>  {"user\_id":"<4.1>"}  user\_id : Delete user’s register ID |
| -- HTTP header --  request\_code: send\_cmd\_result  dev\_id:<2>  trans\_id:<3>  cmd\_return\_code:<4>  blk\_no:<5>  -- HTTP body --  For this command , the execute results with no any data, so there is no any data in body part. |  |  |
|  |  | -- HTTP header --  response\_code:<1>  trans\_id:<2> |

**3.6. Rename the terminal(SET\_USER\_NAME)**

|  |  |  |
| --- | --- | --- |
| **Terminal Request** |  | **WEB Server Response** |
| -- HTTP header --  request\_code:receive\_cmd  dev\_id:<2>  -- HTTP body --  Means the string of the terminal information |  |  |
|  |  | -- HTTP header --  response\_code:<1>  trans\_id:<2>  cmd\_code:SET\_USER\_NAME  -- HTTP body --  {<4>}  The format of <4>  {  "user\_id":"<4.1>",  "user\_name":<4.2>  }  user\_name ： The user name is the code UTF-8 |
| -- HTTP header --  request\_code: send\_cmd\_result  dev\_id:<2>  trans\_id:<3>  cmd\_return\_code:<4>  blk\_no:<5>  -- HTTP body --  For this command , the execute results with no any data, so there is no any data in body part. |  |  |
|  |  | -- HTTP header --  response\_code:<1>  trans\_id:<2> |

**3.7.Change User Privilege (SET\_USER\_PRIVILEGE)**

|  |  |  |
| --- | --- | --- |
| **Terminal Request** |  | **WEB Server Response** |
| -- HTTP header --  request\_code:receive\_cmd  dev\_id:<2>  -- HTTP body --  Means the string of the terminal information |  |  |
|  |  | -- HTTP header --  response\_code:<1>  trans\_id:<2>  cmd\_code:SET\_USER\_PRIVILEGE  -- HTTP body --  {<4>}  The format of <4>  {"user\_id":"<1>","user\_privilege":"<2>"}  user\_privilege : Means users privilege to operate the computer. Set one of the following strings. MANAGER : manager  REGISTER : register  OPERATOR : operator  USER : normal user |
| -- HTTP header --  request\_code: send\_cmd\_result  dev\_id:<2>  trans\_id:<3>  cmd\_return\_code:<4>  blk\_no:<5>  -- HTTP body --  For this command , the execute results with no any data, so there is no any data in body part. |  |  |
|  |  | -- HTTP header --  response\_code:<1>  trans\_id:<2> |

**3.9.** **Get the user ID list (GET\_USER\_ID\_LIST)**

|  |  |  |
| --- | --- | --- |
| **Terminal Request** |  | **WEB Server Response** |
| -- HTTP header --  request\_code:receive\_cmd  dev\_id:<2>  -- HTTP body --  Means the string of the terminal information |  |  |
|  |  | -- HTTP header --  response\_code:<1>  trans\_id:<2>  cmd\_code: GET\_USER\_ID\_LIST  -- HTTP body --  No need the parameter in this part, so there is no data put in the body |
| -- HTTP header –  request\_code:send\_cmd\_result  dev\_id:<2>  trans\_id:<3>  cmd\_return\_code:<4>  blk\_no:<5>  When assign the user ID, the result data is the user ID list, sometimes you can assign more than 100000 users ID list, so we need to divide the result data into several modules, then transmit.  -- HTTP body –-  The following format is the result date will be divided into several modules, then transmit.  {<6>} + bin\_1  The following strings are put into <6>.  {“user\_id\_count”:<6.1>,”one\_user\_id\_size”:<6.2>,”user\_id\_array”:”BIN\_1”}  user\_id\_count field: Assign the amount of the user  one\_user\_id\_size field: Assign the size of the byte units of a user\_id structure data.  user\_id\_array field: This field set“BIN\_1”as the string,means the ID data is put hind in the first binary data.  In binary data means user\_id of the structure data is arranged by sequence.  All structure data interpretation using specific libraries. |  |  |
|  |  | -- HTTP header --  response\_code:<1>  trans\_id:<2> |

**3.10.Get the record data (GET\_LOG\_DATA)**

|  |  |  |
| --- | --- | --- |
| **Terminal Request** |  | **WEB Server Response** |
| -- HTTP header --  request\_code:receive\_cmd  dev\_id:<2>  -- HTTP body --  Means the string of the terminal information |  |  |
|  |  | -- HTTP header --  response\_code:<1>  trans\_id:<2>  cmd\_code:GET\_LOG\_DATA  -- HTTP body --  {<4>}  <4>의 격식  {"begin\_time":”<1>”,"end\_time":”<2>”}  The format of time string is YYYYMMDDhhmmss。  When begin\_time field is blank or no field, it will get all the attendance record data before end\_time record When end\_time field is blank or no field, it will get all the attendance record data after begin\_time record  If the 2 fields are invalid value, it will get all the attendance record data from the terminal. |
| -- HTTP header –  request\_code:send\_cmd\_result  dev\_id:<2>  trans\_id:<3>  cmd\_return\_code:<4>  blk\_no:<5>  When get the record data,get any record data will be a great result data. So we need to divide the result data into several modules, then transmit.  -- HTTP body –-  The following format is the result date will be divided into several modules, then transmit.  {<6>} + bin\_1  Set the strings in <6> as follow。  {“log\_count”:<6.1>,”one\_log\_size”:<6.2>,”log\_array”:”BIN\_1”}  log\_count field : The amount of the assigned gains the record data.  one\_log\_size field : Assign the size of the byte units of a record data.  log\_array field: In this field set“BIN\_1”as the string, means the record data is put hind in the first binary data.  In binary data, means record data of the structure data is arranged by sequence.  All structure data interpretation using specific libraries. |  |  |
|  |  | -- HTTP header --  response\_code:<1>  trans\_id:<2> |

**3.12. Setup Terminal Name (SET\_FK\_NAME)**

|  |  |  |
| --- | --- | --- |
| **Terminal Request** |  | **WEB Server Response** |
| -- HTTP header --  request\_code:receive\_cmd  dev\_id:<2>  -- HTTP body --  Means the string of the terminal information |  |  |
|  |  | -- HTTP header --  response\_code:<1>  trans\_id:<2>  cmd\_code:SET\_FK\_NAME  -- HTTP body --  {<4>}  The format of <4>  {"fk\_name":”<4.1>”}  fk\_name must be English. |
| -- HTTP header --  request\_code: send\_cmd\_result  dev\_id:<2>  trans\_id:<3>  cmd\_return\_code:<4>  blk\_no:<5>  -- HTTP body --  For this command , the execute results with no any data, so there is no any data in body part |  |  |
|  |  | -- HTTP header --  response\_code:<1>  trans\_id:<2> |

**3.13.Clear Log Data (CLEAR\_LOG\_DATA)**

|  |  |  |
| --- | --- | --- |
| **Terminal Request** |  | **WEB Server Response** |
| -- HTTP header --  request\_code:receive\_cmd  dev\_id:<2>  -- HTTP body --  Means the string of the terminal information |  |  |
|  |  | -- HTTP header --  response\_code:<1>  trans\_id:<2>  cmd\_code:CLEAR\_LOG\_DATA  -- HTTP body --  This command no need parameters, so there are no data in body. |
| -- HTTP header --  request\_code: send\_cmd\_result  dev\_id:<2>  trans\_id:<3>  cmd\_return\_code:<4>  blk\_no:<5>  -- HTTP body --  For this command , the execute results with no any data, so there is no any data in body part |  |  |
|  |  | -- HTTP header --  response\_code:<1>  trans\_id:<2> |

**3.14.Clear Enroll Data (CLEAR\_ENROLL\_DATA)**

|  |  |  |
| --- | --- | --- |
| **Terminal Request** |  | **WEB Server Response** |
| -- HTTP header --  request\_code:receive\_cmd  dev\_id:<2>  -- HTTP body --  Means the string of the terminal information |  |  |
|  |  | -- HTTP header --  response\_code:<1>  trans\_id:<2>  cmd\_code:CLEAR\_ENROLL\_DATA  -- HTTP body --  This command no need parameters, so there are no data in body. |
| -- HTTP header --  request\_code: send\_cmd\_result  dev\_id:<2>  trans\_id:<3>  cmd\_return\_code:<4>  blk\_no:<5>  -- HTTP body --  For this command , the execute results with no any data, so there is no any data in body part |  |  |
|  |  | -- HTTP header --  response\_code:<1>  trans\_id:<2> |

**3.15.****Get the Status information form the terminal (GET\_DEVICE\_STATUS)**

|  |  |  |
| --- | --- | --- |
| **Terminal Request** |  | **WEB Server Response** |
| -- HTTP header --  request\_code:receive\_cmd  dev\_id:<2>  -- HTTP body --  Means the string of the terminal information |  |  |
|  |  | -- HTTP header --  response\_code:<1>  trans\_id:<2>  cmd\_code: GET\_DEVICE\_STATUS  -- HTTP body --  This command no need parameters, so there are no data in body. |
| -- HTTP header --  request\_code: send\_cmd\_result  dev\_id:<2>  trans\_id:<3>  cmd\_return\_code:<4>  blk\_no:<5>  -- HTTP body --  {<6>}  There are string in <6>, as follow.  {  “total\_user\_count”:<6.1>,  “user\_count”:<6.2>,  “manager\_count”:<6.3>,  “fp\_count”:<6.4>,  “face\_count”:<6.5>,  “password\_count”:<6.6>,  “idcard\_count”:<6.7>,  “total\_log\_count”:<6.8>  }  <6.1> ~ <6.8>are numeric |  |  |
|  |  | -- HTTP header --  response\_code:<1>  trans\_id:<2> |

**3.16.****Set the user enroll data and information (SET\_USER\_INFO)**

Set the user fingerprint, facial, password, card, name and the privilege from operational data base into the attendance terminal

|  |  |  |
| --- | --- | --- |
| **Terminal Request** |  | **WEB Server Response** |
| -- HTTP header --  request\_code:receive\_cmd  dev\_id:<2>  -- HTTP body --  Means the string of the terminal information |  |  |
|  |  | -- HTTP header --  response\_code:<1>  trans\_id:<2>  cmd\_code:SET\_USER\_INFO  -- HTTP body --  {<4>} + bin\_1 + bin\_2 + … + bin\_k  <4>的格式  {  "user\_id":<4.1>,  "user\_name":<4.2>,  "user\_privilege":<4.3>,  "user\_photo":”BIN\_1”,  "enroll\_data\_array":  [  {“backup\_number”:<5.4.1>,”enroll\_data”:”BIN\_2”},  {“backup\_number”:<5.4.2>,”enroll\_data”:”BIN\_3”},  …,  {“backup\_number”:<5.4.k>,”enroll\_data”:”BIN\_k+1”},  ]  }  user\_id : users enroll number  user\_name : user name, through UTF-8 compile the string  user\_privilege : The privilege of string about user operate the attendance terminal  enroll\_data\_array :The JSON array, use the enroll data which is the user fingerprint, facial, password, ID card as a unit.  The array unit is backup\_number, enroll\_data field JSON object  enroll\_data field identification means the relationship between the register data and which 2 hexadecimal data  Put JSON string <4> fist, then according to the string details put the 2 hexadecimal data  The 2 hexadecimal data saves the actual enroll data. |
| -- HTTP header --  request\_code: send\_cmd\_result  dev\_id:<2>  trans\_id:<3>  cmd\_return\_code:<4>  blk\_no:<5>  -- HTTP body --  For this command , the execute results does not exit any data, so there is no any data in body part |  |  |
|  |  | -- HTTP header --  response\_code:<1>  trans\_id:<2> |

**3.17. Get the fingerprint data, facial data, password, ID card, name and the privilege from attendance terminal (GET\_USER\_INFO)**

|  |  |  |
| --- | --- | --- |
| **Terminal Request** |  | **WEB Server Response** |
| -- HTTP header --  request\_code:receive\_cmd  dev\_id:<2>  -- HTTP body --  Means string of the computer information |  |  |
|  |  | -- HTTP header --  response\_code:<1>  trans\_id:<2>  cmd\_code:GET\_USER\_INFO  -- HTTP body --  {<4>}  <4> format  {"user\_id":”<1>”}  user\_id : to obtain a user enroll number |
| -- HTTP header --  request\_code: send\_cmd\_result  dev\_id:<2>  trans\_id:<3>  cmd\_return\_code:<4>  blk\_no:<5>  -- HTTP body --  {<6>} + bin\_1 + bin\_2 + … + bin\_k  <6> format  {  "user\_id":<6.1>,  "user\_name":<6.2>,  "user\_privilege":<6.3>,  "user\_photo":”BIN\_1”,  "enroll\_data\_array":  [  {“backup\_number”:<7.4.1>,”enroll\_data”:”BIN\_2”},  {“backup\_number”:<7.4.2>,”enroll\_data”:”BIN\_3”},  …,  {“backup\_number”:<7.4.k>,”enroll\_data”:”BIN\_k+1”},  ]  }  The format of the command result data,as the parameter data format of SET\_USER\_INFO |  |  |
|  |  | -- HTTP header --  response\_code:<1>  trans\_id:<2> |

**3.18.Set** **Server IP Address and change the port number (SET\_WEB\_SERVER\_INFO)**

Change the Web Server IP address and port number which need to communication with the time attendance terminal

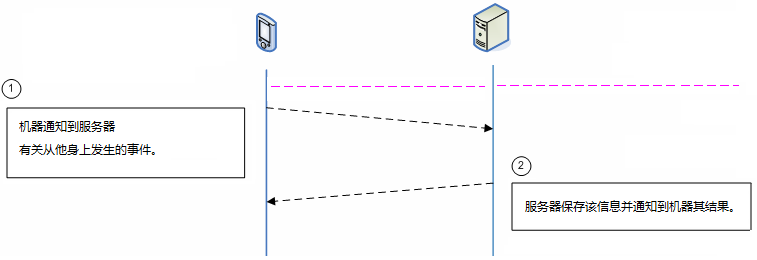
Use the instruction when link the communicating attendance terminal with another Web server

Once the attendance terminal is set another server address, the original communication server address will be cut down. It’s only can communicate with the new server After set the WEB server address and port number, reboot the attendance terminal can take effect.

|  |  |  |
| --- | --- | --- |
| **Terminal Request** |  | **WEB Server Response** |
| -- HTTP header --  request\_code:receive\_cmd  dev\_id:<2>  -- HTTP body --  Means string of computer information |  |  |
|  |  | -- HTTP header --  response\_code:<1>  trans\_id:<2>  cmd\_code:SET\_WEB\_SERVER\_INFO  -- HTTP body --  {<4>}  <4>The format  {"server\_ip":"<1>","server\_port":<2>}  server\_ip ：means the sting of server ip4 address  eg：192.168.0.1  server\_port ：means number value of the server http port |
| -- HTTP header --  request\_code: send\_cmd\_result  dev\_id:<2>  trans\_id:<3>  cmd\_return\_code:<4>  blk\_no:<5>  -- HTTP body --  For this command , the execute results with no any data, so there is no any data in body part |  |  |
|  |  | -- HTTP header --  response\_code:<1>  trans\_id:<2> |

**4. Real Time General Logs (RealTime GLog)**

Real time transmission logs is not done by operator command, that is the terminal initiatively send request to WEB server and response. The request and response text column to WEB server is much different from operator command.



The server save records and

notifies the device

Active upload event log to server

**4.1 Real Time Transmission Logs**

|  |  |  |
| --- | --- | --- |
| **Terminal Request** |  | **WEB Server Response** |
| -- HTTP header –  request\_code:realtime\_glog  dev\_id:<2>  -- HTTP body –-  <3> + bin\_1  In <3> place the JSON sting means record information  Some of the terminal will upload the record data and the record image, so sometimes maybe place the binary data in body part  Format of <3> as follow  {  “user\_id”:”<3.1>”,  “verify\_mode”:<3.2>,  “io\_mode”:<3.3>,  “io\_time”:”<3.4>”,  “log\_image”:”BIN\_1”  }  user\_id :leave the record of the user’s enroll number  verify\_mode : leave the record of identify way  May be to place JSON array as follow  Element of the array means the identify way and the order  Eg [“FP”, “PASSWORD”] means first to identify fingerprint, and then password  io\_mode : Purpose of In/Out （at work or out of work）  io\_time : In/Out time. The format is YYYYMMDDhhmmss  log\_image: means the record image which is placed on the back. |  |  |
|  |  | -- HTTP header --  response\_code:<1>  response\_code Indicate whether real time transmission data is successful or not  OK : Success  ERROR : Failed |

**4.2Real Time Transmission Enroll Data**

|  |  |  |
| --- | --- | --- |
| **Terminal Request** |  | **WEB Server Response** |
| -- HTTP header –  request\_code:realtime\_enroll\_data  dev\_id:<2>  -- HTTP body –-  <3> + bin\_1 + bin\_2 + … + bin\_k  <3>Place the JSON character string which is means enroll data  The format of <3> as follow.  {  “user\_id”:”<3.1>”,  “user\_name”:”<3.2>”,  "user\_privilege":<3.3>,  "user\_photo":”BIN\_1”,  "enroll\_data\_array":  [  {“backup\_number”:<3.5.1>,”enroll\_data”:”BIN\_2”},  {“backup\_number”:<3.5.2>,”enroll\_data”:”BIN\_3”},  …,  {“backup\_number”:<3.5.k>,”enroll\_data”:”BIN\_k+1”},  ]  }  user\_id : Enroll user ID number  user\_name : Enroll user name and UTF-8 code  user\_privilege : Indicate that whether the user has the privilege to operate the terminal.  enroll\_data\_array : Used the user’s enroll data as the element of JSON array  The element of the array contains backup\_number, enroll\_data field of JSON object.  Place the sting of the corresponding binary data into the enroll\_data field  Behind JSON string <3>, continue place the corresponding binary data  These binary data contains the real enroll data |  |  |
|  |  | -- HTTP header --  response\_code:<1>  response\_codeIndicate whether real time transmission data is successful or not  OK : Success  ERROR : Failed |