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.

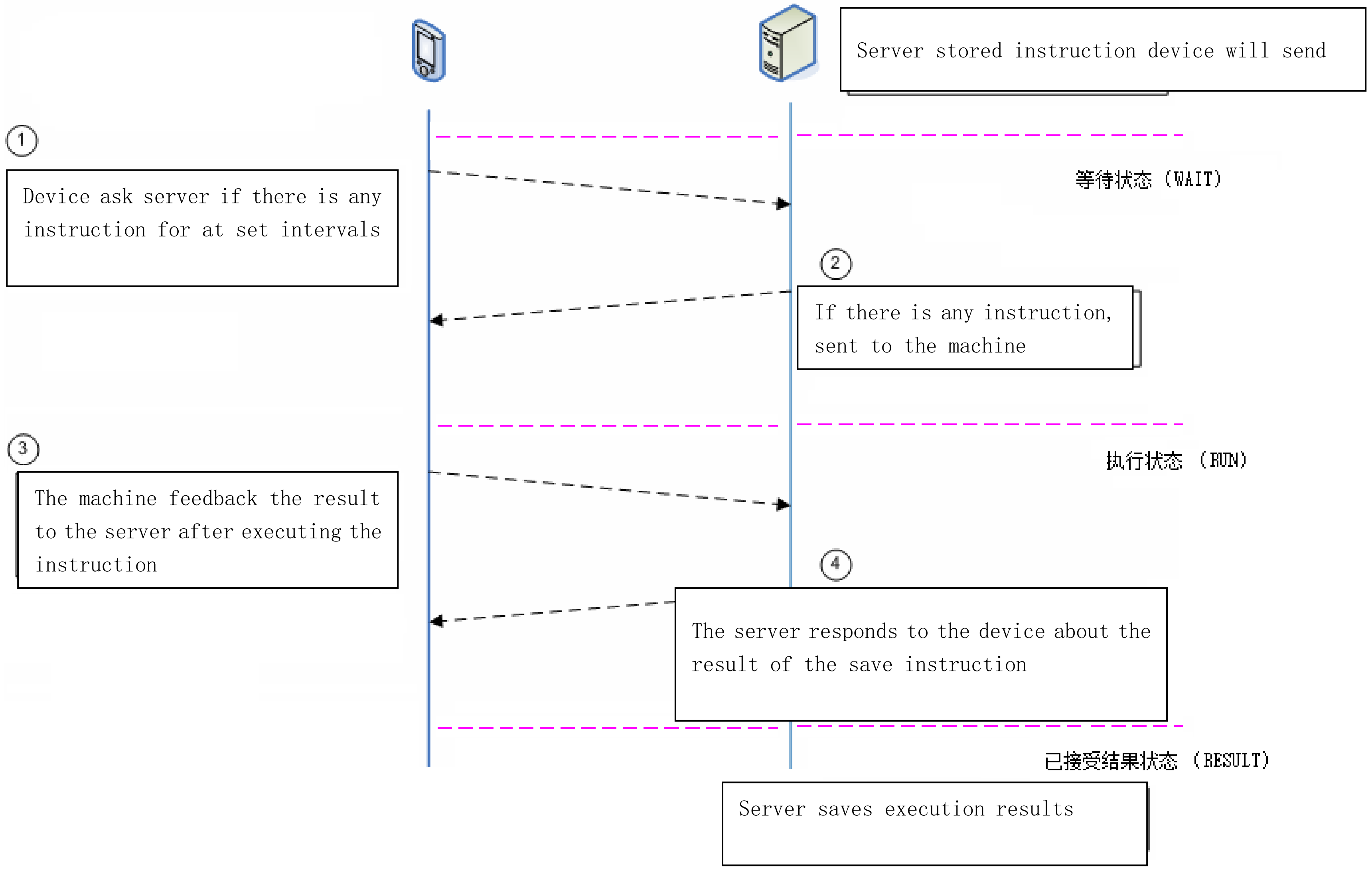
1.4 Communicating process of machine and its server

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.


Server stored instruction device will send
1
Device ask server if there is any instruction for at set intervals
等待状态（WAIT）
2
If there is any instruction, sent to the machine
3
The machine feedback the result to the server after executing the instruction
执行状态（RUN）
4
The server responds to the device about the result of the save instruction
已接受结果状态（RESULT）
Server saves execution results

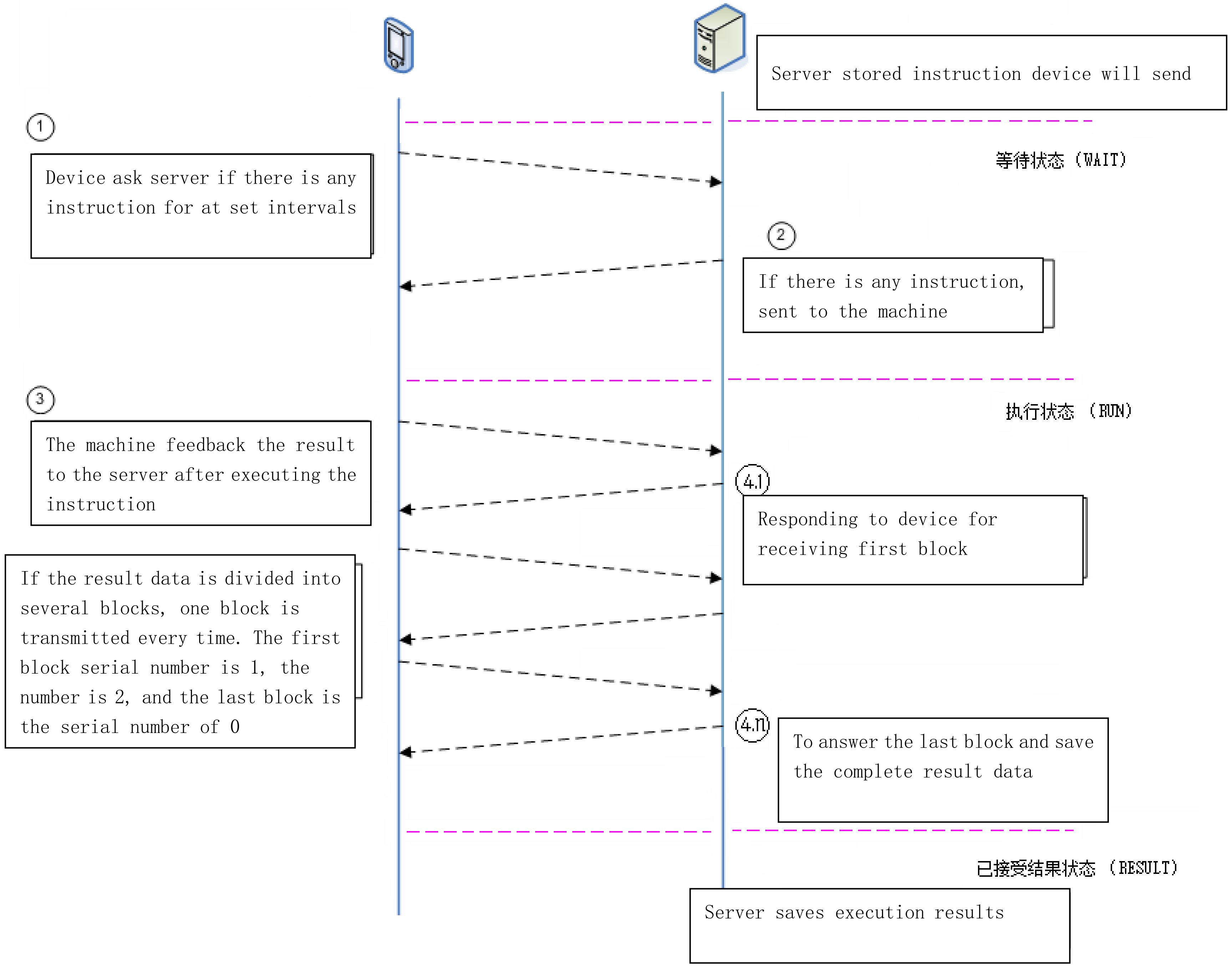
P2-1.The communication process in the command processing.

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 stored instruction device will send

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

If there is any instruction, sent to the machine

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

Responding to device for receiving first block

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

To answer the last block and save the complete result data

Server saves execution results

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	GET_ENROLL_DATA
Set Database Enroll Data To Terminal	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
Get Enroll ID List	GET_USER_ID_LIST
Get Log Data	GET_LOG_DATA
Set Terminal name	SET_FK_NAME
Clear Log Data	CLEAR_LOG_DATA
Clear 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)

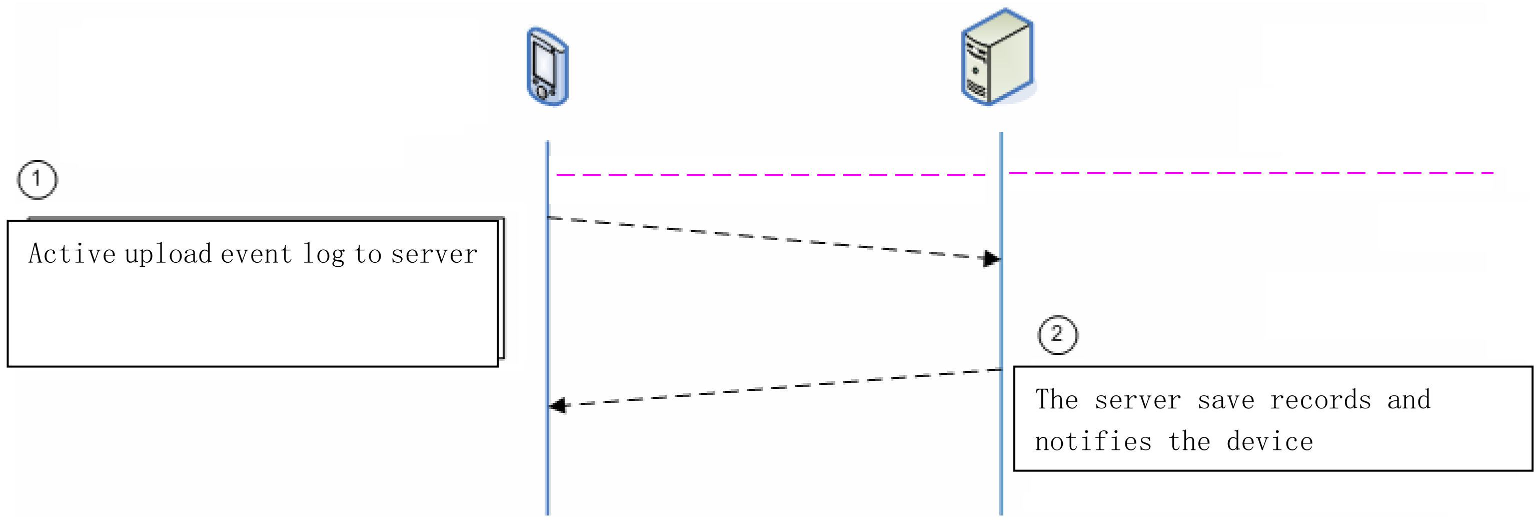
Change the Web Server IP address and port number which need to communication with the time attendance terminal
Use the instruction when link the communicating attendance terminal with another Web server
Once the attendance terminal is set another server address, the original communication server address 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.


①
Active upload event log to server
②
The server save records and notifies the device

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