

WEBSTER

Easy Data Centralization with Webster



FINGERTEC

Webster SDK
Describe Webster demo function

• CONTENTS

3-4 Webster SDK Introduction

Requisite Knowledge

Webster Table

- webster_terminal
- webster_userinfo
- webster_checkout
- webster_command
- webster_usertemplate_fp10
- webster_usertemplate

5-11 Webster SDK Functions

General Data Management

- check_terminal_log_user()
- clear_terminal_all ()

Terminal Management

- check_terminal_info()
- reboot_terminal()
- trigger_terminal()

User Management

- upload_user_one()
- db_delete_user()
- delete_user_one()

Template Management

- upload_fptemplate_one()
- db_delete_fptemplate()
- delete_fptemplate_one()

Transaction Log Management

- clear_terminal_log ()
- db_delete_log()

• Introducing Webster

Webster: An Overview

Before starting to use this Webster SDK Manual, it is important to understand how Webster functions.

Webster, an online web based platform, that is used to consolidate and centralize the database for time and attendance. With the Webster platform, you can access raw time attendance data anywhere and everywhere using an internet browser. Webster also has the capacity to act as a platform between a reader and the TCMS V2 bundled software or even TimeTec, the newly released web-based attendance solution. (For more information on these software applications, visit <http://www.fingertec.com/software/>)

This manual guides system integrators on how to use the Webster SDK to integrate FingerTec systems to a third party application, to create a customized solution when needed. Before proceeding, make sure you have all the components listed under Getting Started. The Webster SDK functions are then listed on the following chapter.

Getting Started

First things first. In order to use the guide successfully, you should have:

- Installed Webster or TimeTec.
- FingerTec terminals that are connected to Webster or TimeTec.
- Installed Microsoft Visual Studio 2008 or above OR Microsoft Visual Studio 2008 Express or above.
- Knowledge and familiarity with .NET languages such as C# and VB.

Webster SDK Table

Now that you've done the above, its time to proceed. The following are the tables involved in Webster SDK Functions:

[webster_terminal](#)

Table to store terminal

[webster_userinfo](#)

Table to store terminal user information

[webster_checkinout](#)

Table to store terminal transaction log

[webster_command](#)

Table to store request toward terminal which triggered from Webster SDK Functions. Terminal will look up pending request from this table and execute constantly.

[webster_usertemplate_fp10](#)

Table to store Webster Vx10.0 fingerprint template

[webster_usertemplate](#)

Table to store Webster Vx9.0 fingerprint template

• Webster SDK Functions

General Data Management

`check_terminal_log_user()`

[Method]

`check_terminal_log_user(conn: MySqlConnection; serialno: String): Bool`

[Feature]

Get terminal's info, user information and transaction log from terminal and insert into table `webster_terminal`, `webster_userinfo` and `webster_checkout` respectively.

[Parameter Declaration]

`conn` (required): mysql connection

`serialno` (required): serial number

[Return Value]

Return true for success, or false for fail

[Affected Table]

`webster_terminal`

`webster_userinfo`

`webster_checkout`

`clear_terminal_all ()`

[Method]

`clear_terminal_all(conn: MySqlConnection; serialno: String): Bool`

[Feature]

Clear all data in terminal including user information and transaction log

[Parameter Declaration]

`conn` (required): mysql connection

`serialno` (required): serial number

[Return Value]

Return true for success, or false for fail

Terminal Management

`check_terminal_info()`

[Method]

`check_terminal_info(conn: MySqlConnection; serialno: String): Bool`

[Feature]

Get terminal info from terminal and insert into table `webster_terminal`

[Parameter Declaration]

`conn` (required): mysql connection

`serialno` (required): serial number

[Return Value]

Return true for success, or false for fail

[Affected Table]

`webster_terminal`

`reboot_terminal()`

[Method]

`reboot_terminal(conn: MySqlConnection; serialno: string): Bool`

[Feature]

Reboot terminal

[Parameter Declaration]

`conn` (required): mysql connection

`serialno` (required): serial number

[Return Value]

Return true for success, or false for fail

trigger_terminal()

[Method]

trigger_terminal(conn: MySqlConnection; serialno: string): Bool

[Feature]

Trigger terminal to look up pending request from table webster_command and execute immediately.

This function only works in LAN environment.

[Parameter Declaration]

conn (required): mysql connection

serialno (required): serial number

[Return Value]

Return true for success, or false for fail

User Management

upload_user_one()

[Method]

upload_user_one(conn: MySqlConnection; serialno: string, enrollno: int, name: string, privilege: int, password: string, cardno: long, accgroup: int, u_timezone: string): Bool

[Feature]

Upload a user to terminal

[Parameter Declaration]

conn (required): mysql connection

serialno (required): serial number

enrollno (required): user id

name: user name

privilege: user privilege represent by integer. Default value is 0.

0 – User

1 – Enroller

2 – Admin

3 – Supervisor

password: user password

cardno: card number. Default value is 0.

accgroup: user group. Default value is 0.

u_timezone: user timezone. Default value is 00000000

[Return Value]

Return true for success, or false for fail

db_delete_user()

[Method]

db_delete_user(conn: MySqlConnection; serialno: string): Bool

[Feature]

Delete all user of given serial number from table webster_userinfo

[Parameter Declaration]

conn (required): mysql connection

serialno (required): serial number

[Return Value]

Return true for success, or false for fail

[Affected Table]

webster_userinfo

delete_user_one()

[Method]

delete_user_one(conn: MySqlConnection; serialno: string; enrollno: int): Bool

[Feature]

Delete a user from terminal

[Parameter Declaration]

conn (required): mysql connection

serialno (required): serial number

enrollno (required): user id

[Return Value]

Return true for success, or false for fail

Template Management

`upload_fptemplate_one()`

[Method]

`upload_fptemplate_one(conn: MySqlConnection; serialno: string, enrollno: int, template: string, fingerid: int, valid: int): Bool`

[Feature]

Upload a fingerprint template to terminal

[Parameter Declaration]

`conn` (required): mysql connection

`serialno` (required): serial number

`enrollno` (required): user id

`template` (required): fingerprint template

`fingerid`: fingerprint sequence id. Default value is 0. Possible value is 0-9.

`valid`: to indicate whether fingerprint is valid. Default value is 1

[Return Value]

Return true for success, or false for fail

`db_delete_fptemplate()`

[Method]

`db_delete_fptemplate(conn: MySqlConnection; serialno: String): Bool`

[Feature]

Delete all finger print of given serial number from table `webster_usertemplate_fp10` and `webster_usertemplate`

[Parameter Declaration]

`conn` (required): mysql connection

`serialno` (required): serial number

[Return Value]

Return true for success, or false for fail

[Affected Table]

`webster_usertemplate_fp10`

`webster_usertemplate`

`delete_fptemplate_one()`

[Method]

`delete_fptemplate_one(conn: MySqlConnection; serialno: string; enrollno: int; fingerid: int): Bool`

[Feature]

Delete a fingerprint template from terminal

[Parameter Declaration]

`conn` (required): mysql connection

`serialno` (required): serial number

`enrollno` (required): user id

`fingerid` (required): fingerprint sequence id. Possible value is 0-9.

[Return Value]

Return true for success, or false for fail

Transaction Log Management

`clear_terminal_log ()`

[Method]

`clear_terminal_log(conn: MySqlConnection; serialno: String): Bool`

[Feature]

Clear all transaction log in terminal

[Parameter Declaration]

`conn` (required): mysql connection

`serialno` (required): serial number

[Return Value]

Return true for success, or false for fail

`db_delete_log()`

[Method]

`db_delete_log(conn: MySqlConnection; serialno: string): Bool`

[Feature]

Delete all transaction log of given serial number from table `webster_checkout`

[Parameter Declaration]

`conn` (required): mysql connection

`serialno` (required): serial number

[Return Value]

Return true for success, or false for fail

[Affected Table]

`webster_checkout`

