ZPI_Server

Generated by Doxygen 1.8.5

Tue Nov 19 2013 21:54:58

Contents

Chapter 1

Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 2

Class Documentation

2.1 Client Class Reference

Represents handled client-agent.

`#include <client.h>`

Public Member Functions

- Client (int fd, const string &ip, int port)

  Constructor, initialize data with incoming connection returns.
- void readData ()

  Read data from client-agent.
- void process ()

  Process all waiting request.
- void fetchConfig ()

  Fetch client-agent current configuration.

Public Attributes

- int fd

  Client id, mostly socket descriptor used for its communication.
- string ip

  Client ip address.
- int port

  Client port.
- bool toDelete

  If client is not used.
- bool settingsChanged

  If client settings changed.

2.1.1 Detailed Description

Represents handled client-agent.

Definition at line 17 of file client.h.

2.1.2 Constructor & Destructor Documentation

2.1.2.1 Client::Client (int *fd,* const string & *ip,* int *port*)

Constructor, initialize data with incoming connection returns.

Parameters

fd	Socket descriptor.
ip	Client ip address.
port	Client port.

Definition at line 31 of file client.cpp.

2.1.3 Member Function Documentation

2.1.3.1 void Client::fetchConfig ( )

Fetch client-agent current configuration.

Returns

None.

Definition at line 169 of file client.cpp.

2.1.3.2 void Client::process ( )

Process all waiting request.

Returns

None.

Definition at line 104 of file client.cpp.

2.1.3.3 void Client::readData ( )

Read data from client-agent.

Returns

None.

Definition at line 49 of file client.cpp.

The documentation for this class was generated from the following files:

- client.h
- client.cpp

2.2 DB Class Reference

Database handler. Uses SQLite.

```
#include <db.h>
```

Static Public Member Functions

- static bool open (const string &path)

  Open file with database.

- static bool createTables ()

  Creates initial tables.
- static bool generateNewKey (char key[16])

  Generate new key for agent.
- static bool findAgentByKey (const char key[16], TDBAgent &agent)

  Find agent with given key.
- static bool findAgentById (uint16_t id, TDBAgent &agent)

  Find agent with given id.
- static bool updateAgent (const TDBAgent &agent)

  Update agent data based on id inside.
- static bool insertRecord (int agentId, const TSensorsData &data)

  Insert data from sensors connected to agent with given id.
- static bool insertRecord (const TDBAgent &agent, const TSensorsData &data)

  Insert data from sensors connected to given agent.
- static bool insertRecords (int agentId, const vector< TSensorsData > &data)

  Isert many data from sensors connected to given agent id.
- static bool insertRecords (const TDBAgent &agent, const vector< TSensorsData > &data)

  Isert many data from sensors connected to given agent.
- static bool getRecords (int agentId, uint32_t startDate, uint32_t endDate, vector< TSensorsRecord > &records)

  Get sensors data for given agent id based on time interval.
- static bool cleanup ()

  Remove old records from database.
- static void close ()

  Close database.

Static Public Attributes

- static sqlite3 * db

  Current database handle.

2.2.1 Detailed Description

Database handler. Uses SQLite.

Definition at line 85 of file db.h.

2.2.2 Member Function Documentation

2.2.2.1 static bool DB::createTables ( ) [static]

Creates initial tables.

Returns

If succeeded.

Definition at line 26 of file db.cpp.

2.2.2.2 static bool DB::findAgentById (uint16_t *id*, TDBAgent & *agent*) [static]

Find agent with given id.

Parameters

in	*id*	Agent id to check.
out	*agent*	Agent data.

Returns

If succeeded.

Definition at line 105 of file db.cpp.

2.2.2.3 static bool DB::findAgentByKey (const char *key[16],* TDBAgent & *agent*) [static]

Find agent with given key.

Parameters

in	*key*	Key to check.
out	*agent*	Agent data.

Returns

If succeeded.

Definition at line 80 of file db.cpp.

2.2.2.4 static bool DB::generateNewKey (char *key[16]*) [static]

Generate new key for agent.

Parameters

out	*key*	Table with key.

Returns

If succeeded.

Definition at line 65 of file db.cpp.

2.2.2.5 static bool DB::getRecords (int *agentId,* uint32_t *startDate,* uint32_t *endDate,* vector< TSensorsRecord > & *records*) [static]

Get sensors data for given agent id based on time interval.

Parameters

in	*agentId*	Agent-owner of sensors data.
in	*startDate*	Interval start time.
in	*endDate*	Interval end time.
out	*records*	Filtered sensors datas.

Definition at line 232 of file db.cpp.

2.2.2.6 static bool DB::insertRecord (int *agentId,* const TSensorsData & *data*) [static]

Insert data from sensors connected to agent with given id.

Parameters

agentId	Agent-owner of sensors data.
data	Sensors data.

Returns

If succeeded.

Definition at line 170 of file db.cpp.

2.2.2.7 static bool DB::insertRecord (const TDBAgent & *agent,* const TSensorsData & *data*) [inline],[static]

Insert data from sensors connected to given agent.

Parameters

agent	Agent-owner of sensors data.
data	Sensors data.

Returns

If succeeded.

Definition at line 156 of file db.h.

2.2.2.8 static bool DB::insertRecords (int *agentId,* const vector< TSensorsData > & *data*) [static]

Isert many data from sensors connected to given agent id.

Parameters

agentId	Agent-owner of sensors datas.
data	List of sensors datas.

Returns

If succeeded.

Definition at line 211 of file db.cpp.

2.2.2.9 static bool DB::insertRecords (const TDBAgent & *agent,* const vector< TSensorsData > & *data*) [inline], [static]

Isert many data from sensors connected to given agent.

Parameters

agent	Agent-owner of sensors datas.
data	List of sensors datas.

Returns

If succeeded.

Definition at line 174 of file db.h.

2.2.2.10 static bool DB::open (const string & *path*) [static]

Open file with database.

Parameters

path	Path pointing to database file.

Returns

If succeeded.

Definition at line 16 of file db.cpp.

2.2.2.11 static bool DB::updateAgent (const TDBAgent & *agent*) [static]

Update agent data based on id inside.

Parameters

in	*agent*	Agent data to update with existing agent id.

Returns

If succeeded.

Definition at line 129 of file db.cpp.

The documentation for this class was generated from the following files:

- db.h
- db.cpp

2.3 TAgentData Class Reference

Agent data from one moment.

`#include <agents.h>`

Public Attributes

- uint32_t time
  Data time.
- TPacketAgentData packet
  Agent data packet.

2.3.1 Detailed Description

Agent data from one moment.

Definition at line 13 of file agents.h.

The documentation for this class was generated from the following file:

- agents.h

2.4 TDBAgent Class Reference

Client-agent DTO for database use.

`#include <db.h>`

Public Attributes

- int id

  Agent id.
- string name

  Agent name.
- vector< TDBService > services

  List of agent services.
- string tempPath

  Temperature path.
- int tempDivider

  Temperature divider.
- int interval

  Time interval.

2.4.1 Detailed Description

Client-agent DTO for database use.

Definition at line 33 of file db.h.

The documentation for this class was generated from the following file:

- db.h

2.5 TDBService Class Reference

Service DTO for database use.

`#include <db.h>`

Public Attributes

- string name

  Service name.
- bool tcp

  Service tranport type (TCP/UDP).
- int port

  Service port number.

2.5.1 Detailed Description

Service DTO for database use.

Definition at line 18 of file db.h.

The documentation for this class was generated from the following file:

- db.h

2.6 TSensorsRecord::TDisk Struct Reference

Simple disc data.

`#include <db.h>`

Public Attributes

- string name

  Disc name.
- double usage

  Percent disc usage.

2.6.1 Detailed Description

Simple disc data.

Definition at line 60 of file db.h.

The documentation for this struct was generated from the following file:

- db.h

2.7 TSensorsRecord Class Reference

Client-agent sensors DTO for database use.

`#include <db.h>`

Classes

- struct TDisk

  Simple disc data.

Public Attributes

- int id

  Agent id.
- uint32_t timestamp

  Time of sensors read.
- double temp

  Temperature value.
- double cpuUsage

  Cpu percent usage.
- double ramUsage

  RAM percent usage.
- vector< TDisk > disks

  List of discs data.

2.7.1 Detailed Description

Client-agent sensors DTO for database use.

Definition at line 54 of file db.h.

The documentation for this class was generated from the following file:

- db.h

Index