

ZPI_Server

Generated by Doxygen 1.8.5

Tue Nov 19 2013 22:33:16

Contents

1	Class Index	1
1.1	Class List	1
2	Class Documentation	3
2.1	Client Class Reference	3
2.1.1	Detailed Description	3
2.1.2	Constructor & Destructor Documentation	4
2.1.2.1	Client	4
2.1.3	Member Function Documentation	5
2.1.3.1	fetchConfig	5
2.1.3.2	process	5
2.1.3.3	readData	5
2.2	DB Class Reference	5
2.2.1	Detailed Description	6
2.2.2	Member Function Documentation	6
2.2.2.1	cleanup	6
2.2.2.2	close	6
2.2.2.3	createTables	7
2.2.2.4	findAgentById	7
2.2.2.5	findAgentByKey	7
2.2.2.6	generateNewKey	7
2.2.2.7	getRecords	7
2.2.2.8	insertRecord	8
2.2.2.9	insertRecord	8
2.2.2.10	insertRecords	8
2.2.2.11	insertRecords	8
2.2.2.12	open	9
2.2.2.13	updateAgent	9
2.3	TAgentData Class Reference	9
2.3.1	Detailed Description	10
2.4	TDBAgent Class Reference	10

2.4.1 Detailed Description	10
2.5 TDBService Class Reference	10
2.5.1 Detailed Description	11
2.6 TSensorsRecord::TDisk Struct Reference	11
2.6.1 Detailed Description	11
2.7 TSensorsRecord Class Reference	11
2.7.1 Detailed Description	12
 Index	 13

Chapter 1

Class Index

1.1 Class List

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

Client	Represents handled client-agent	3
DB	Database handler. Uses SQLite	5
TAgentData	Agent data from one moment	9
TDBAgent	Client-agent DTO for database use	10
TDBService	Service DTO for database use	10
TSensorsRecord::TDisk	Simple disc data	11
TSensorsRecord	Client-agent sensors DTO for database use	11

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.

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

<i>fd</i>	Socket descriptor.
<i>ip</i>	Client ip address.
<i>port</i>	Client port.

2.1.3 Member Function Documentation

2.1.3.1 void Client::fetchConfig ()

Fetch client-agent current configuration.

Returns

None.

2.1.3.2 void Client::process ()

Process all waiting request.

Returns

None.

2.1.3.3 void Client::readData ()

Read data from client-agent.

Returns

None.

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

2.2.2 Member Function Documentation

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

Remove old records from database.

Returns

If succeeded.

2.2.2.2 static void DB::close () [static]

Close database.

Returns

None.

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

Creates initial tables.

Returns

If succeeded.

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

Find agent with given id.

Parameters

in	<i>id</i>	Agent id to check.
out	<i>agent</i>	Agent data.

Returns

If succeeded.

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

Find agent with given key.

Parameters

in	<i>key</i>	Key to check.
out	<i>agent</i>	Agent data.

Returns

If succeeded.

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

Generate new key for agent.

Parameters

out	<i>key</i>	Table with key.
-----	------------	-----------------

Returns

If succeeded.

2.2.2.7 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	<i>agentId</i>	Agent-owner of sensors data.
in	<i>startDate</i>	Interval start time.
in	<i>endDate</i>	Interval end time.
out	<i>records</i>	Filtered sensors datas.

Returns

If succeeded.

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

Insert data from sensors connected to agent with given id.

Parameters

<i>agentId</i>	Agent-owner of sensors data.
<i>data</i>	Sensors data.

Returns

If succeeded.

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

Insert data from sensors connected to given agent.

Parameters

<i>agent</i>	Agent-owner of sensors data.
<i>data</i>	Sensors data.

Returns

If succeeded.

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

Insert many data from sensors connected to given agent id.

Parameters

<i>agentId</i>	Agent-owner of sensors datas.
<i>data</i>	List of sensors datas.

Returns

If succeeded.

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

Insert many data from sensors connected to given agent.

Parameters

<i>agent</i>	Agent-owner of sensors datas.
<i>data</i>	List of sensors datas.

Returns

If succeeded.

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

Open file with database.

Parameters

<i>path</i>	Path pointing to database file.
-------------	---------------------------------

Returns

If succeeded.

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

Update agent data based on id inside.

Parameters

<i>in</i>	<i>agent</i>	Agent data to update with existing agent id.
-----------	--------------	----------------------------------------------

Returns

If succeeded.

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.

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.

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 transport type (TCP/UDP).
- int [port](#)
Service port number.

2.5.1 Detailed Description

Service DTO for database use.

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.

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.

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

- db.h

Index

- cleanup
 - DB, [6](#)
- Client, [3](#)
 - Client, [4](#)
 - fetchConfig, [5](#)
 - process, [5](#)
 - readData, [5](#)
- close
 - DB, [6](#)
- createTables
 - DB, [6](#)
- DB, [5](#)
 - cleanup, [6](#)
 - close, [6](#)
 - createTables, [6](#)
 - findAgentById, [7](#)
 - findAgentByKey, [7](#)
 - generateNewKey, [7](#)
 - getRecords, [7](#)
 - insertRecord, [8](#)
 - insertRecords, [8](#)
 - open, [9](#)
 - updateAgent, [9](#)
- fetchConfig
 - Client, [5](#)
- findAgentById
 - DB, [7](#)
- findAgentByKey
 - DB, [7](#)
- generateNewKey
 - DB, [7](#)
- getRecords
 - DB, [7](#)
- insertRecord
 - DB, [8](#)
- insertRecords
 - DB, [8](#)
- open
 - DB, [9](#)
- process
 - Client, [5](#)
- readData
 - Client, [5](#)
- TAgentData, [9](#)
- TDBAgent, [10](#)
- TDBService, [10](#)
- TSensorsRecord, [11](#)
- TSensorsRecord::TDisk, [11](#)
- updateAgent
 - DB, [9](#)