

TicTacToe

Generated by Doxygen 1.9.1

1 Namespace Index	1
1.1 Namespace List	1
2 Class Index	3
2.1 Class List	3
3 File Index	5
3.1 File List	5
4 Namespace Documentation	7
4.1 Controllers Namespace Reference	7
4.1.1 Detailed Description	7
4.2 Threads Namespace Reference	7
4.2.1 Detailed Description	7
5 Class Documentation	9
5.1 CryptClass Class Reference	9
5.1.1 Member Function Documentation	10
5.1.1.1 aes256decode()	10
5.1.1.2 aes256encode()	10
5.1.1.3 aes256fromBase64Str()	10
5.1.1.4 aes256fromHexStr()	12
5.1.1.5 aes256toBase64Str()	12
5.1.1.6 aes256toHexStr()	13
5.1.1.7 base64aes256decode()	13
5.1.1.8 base64aes256encode()	13
5.1.1.9 base64decode()	14
5.1.1.10 base64encode()	14
5.1.1.11 bcrypt_checkpw()	15
5.1.1.12 bcrypt_gensalt()	15
5.1.1.13 bcrypt_hashpw()	15
5.1.1.14 rand_read()	16
5.1.1.15 setlv()	16
5.1.1.16 setKey()	17
6 File Documentation	19
6.1 controllers.h File Reference	19
6.1.1 Detailed Description	20
6.2 cryptclass.h File Reference	20
6.2.1 Detailed Description	20
Index	21

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

Controllers	
Contains instances of various controllers used in the application	7
Threads	
Contains instances of QThread used for different managers	7

Chapter 2

Class Index

2.1 Class List

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

CryptClass	9
--------------------------------------	---

Chapter 3

File Index

3.1 File List

Here is a list of all documented files with brief descriptions:

[controllers.h](#)

Contains declarations for database manager and related threads. Provides external linkage for DatabaseManager and QThread instances 19

[cryptclass.h](#)

The [CryptClass](#) class Manges password encryption Manges encrypting user data to database Manges decrypting user data from database Interfaces with dbcontroller 20

Chapter 4

Namespace Documentation

4.1 Controllers Namespace Reference

Contains instances of various controllers used in the application.

Variables

- DatabaseManager **dbManager**

4.1.1 Detailed Description

Contains instances of various controllers used in the application.

4.2 Threads Namespace Reference

Contains instances of QThread used for different managers.

Variables

- QThread **dbManagerThread**

4.2.1 Detailed Description

Contains instances of QThread used for different managers.

Chapter 5

Class Documentation

5.1 CryptClass Class Reference

Static Public Member Functions

- static int [rand_read](#) (int fd, char *out, size_t count)
Random read from a file descriptor.
- static int [bcrypt_gensalt](#) (int salt_rounds, char salt[BCRYPT_HASHSIZE])
Generate a salt for bcrypt.
- static int [bcrypt_hashpw](#) (const char *passwd, const char salt[BCRYPT_HASHSIZE], char hash[BCRYPT_HASHSIZE])
Hash a password using bcrypt.
- static int [bcrypt_checkpw](#) (const char *passwd, const char hash[BCRYPT_HASHSIZE])
Check a password against a bcrypt hash.
- static void [setKey](#) (QString hexKeyString)
Create Key (setter for key256)
- static void [setIv](#) (QString hexIvString)
Create iv128 (setter for iv128)
- static QString [getKey](#) ()
- static QString [getIv](#) ()
- static int [base64encode](#) (const char *textBuf, int textLen, char *encBuf)
Encode a string using Base64.
- static int [base64decode](#) (const char *encBuf, int encLen, char *textBuf)
Decode String using Base64 Encoder.
- static int [aes256encode](#) (const char *textBuf, int textLen, char *encBuf)
AES 256 Encoder.
- static int [aes256decode](#) (const char *encBuf, int encLen, char *textBuf)
AES 256 Decoder.
- static QString [aes256toHexStr](#) (const char *charBuf, int len)
AES 256 to Hex String.
- static QByteArray [aes256fromHexStr](#) (const QString qString)
AES 256 from Hex String.
- static QString [aes256toBase64Str](#) (const char *aes256Buf, int len)
AES 256 to Base64 String.
- static QByteArray [aes256fromBase64Str](#) (const QString base64str)
AES 256 from Base64 String.
- static QString [base64aes256encode](#) (QString textString)
base64 aes256 encode
- static QString [base64aes256decode](#) (QString textString)
Base64 aes256 Decode.

5.1.1 Member Function Documentation

5.1.1.1 aes256decode()

```
int CryptClass::aes256decode (
    const char * encBuf,
    int encLen,
    char * textBuf ) [static]
```

AES 256 Decoder.

Parameters

<i>const</i>	char* <i>encBuf</i>
<i>int</i>	<i>encLen</i>
<i>char*</i>	<i>textBuf</i>

Returns

int

5.1.1.2 aes256encode()

```
int CryptClass::aes256encode (
    const char * textBuf,
    int textLen,
    char * encBuf ) [static]
```

AES 256 Encoder.

Parameters

<i>const</i>	char* <i>textBuf</i>
<i>int</i>	<i>textLen</i>
<i>char*</i>	<i>encBuf</i>

Returns

int

5.1.1.3 aes256fromBase64Str()

```
QByteArray CryptClass::aes256fromBase64Str (
    const QString base64str ) [static]
```

AES 256 from Base64 String.

Parameters

<i>const</i>	char* charBuf
<i>int</i>	len

Returns

int

5.1.1.4 aes256fromHexStr()

```
QByteArray CryptClass::aes256fromHexStr (
    const QString qString ) [static]
```

AES 256 from Hex String.

Parameters

<i>const</i>	char* charBuf
<i>int</i>	len

Returns

int

5.1.1.5 aes256toBase64Str()

```
QString CryptClass::aes256toBase64Str (
    const char * aes2656Buf,
    int len ) [static]
```

AES 256 to Base64 String.

Parameters

<i>const</i>	char* charBuf
<i>int</i>	len

Returns

int

5.1.1.6 aes256toHexStr()

```
QString CryptClass::aes256toHexStr (
    const char * charBuf,
    int len ) [static]
```

AES 256 to Hex String.

Parameters

<i>const</i>	char* <i>charBuf</i>
<i>int</i>	<i>len</i>

Returns

int

5.1.1.7 base64aes256decode()

```
QString CryptClass::base64aes256decode (
    QString textString ) [static]
```

Base64 aes256 Decode.

Parameters

<i>QString</i>	<i>textString</i>
----------------	-------------------

Returns

int

5.1.1.8 base64aes256encode()

```
QString CryptClass::base64aes256encode (
    QString textString ) [static]
```

base64 aes256 encode

Parameters

<i>QString</i>	<i>textString</i>
----------------	-------------------

Returns

int

5.1.1.9 base64decode()

```
int CryptClass::base64decode (
    const char * encBuf,
    int encLen,
    char * textBuf ) [static]
```

Decode String using Base64 Encoder.

Parameters

<i>const</i>	char* <i>encBuf</i>
<i>int</i>	<i>len</i>
<i>char*</i>	<i>textBuf</i>

Returns

int

5.1.1.10 base64encode()

```
int CryptClass::base64encode (
    const char * textBuf,
    int textLen,
    char * encBuf ) [static]
```

Encode a string using Base64.

Parameters

<i>textBuf</i>	The input text buffer.
<i>len</i>	The length of the input text.
<i>encBuf</i>	The output encoded buffer.

Returns

0 on success, non-zero on failure.

5.1.1.11 bcrypt_checkpw()

```
int CryptClass::bcrypt_checkpw (
    const char * passwd,
    const char hash[BCRYPT_HASHSIZE] ) [static]
```

Check a password against a bcrypt hash.

Parameters

<i>passwd</i>	The password to check.
<i>hash</i>	The hash to check against.

Returns

0 if the password matches, non-zero otherwise.

5.1.1.12 bcrypt_gensalt()

```
int CryptClass::bcrypt_gensalt (
    int salt_rounds,
    char salt[BCRYPT_HASHSIZE] ) [static]
```

Generate a salt for bcrypt.

Parameters

<i>salt_rounds</i>	The number of salt rounds.
<i>salt</i>	The generated salt.

Returns

0 on success, non-zero on failure.

5.1.1.13 bcrypt_hashpw()

```
int CryptClass::bcrypt_hashpw (
    const char * passwd,
    const char salt[BCRYPT_HASHSIZE],
    char hash[BCRYPT_HASHSIZE] ) [static]
```

Hash a password using bcrypt.

Parameters

<i>passwd</i>	The password to hash.
<i>salt</i>	The salt to use.
<i>hash</i>	The resulting hash.

Returns

0 on success, non-zero on failure.

5.1.1.14 rand_read()

```
int CryptClass::rand_read (
    int fd,
    char * out,
    size_t count ) [static]
```

Random read from a file descriptor.

Parameters

<i>fd</i>	The file descriptor.
<i>out</i>	The output buffer.
<i>count</i>	The number of bytes to read.

Returns

0 on success, -1 on failure.

5.1.1.15 setIv()

```
void CryptClass::setIv (
    QString hexIvString ) [static]
```

Create iv128 (setter for iv128)

Parameters

<i>QString</i>	<i>hexIvString</i>
----------------	--------------------

Returns

void

5.1.1.16 setKey()

```
void CryptClass::setKey (
    QString hexKeyString )    [static]
```

Create Key (setter for key256)

Parameters

<i>QString</i>	<i>hexKeyString</i>
----------------	---------------------

Returns

void

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

- [cryptclass.h](#)
- [cryptclass.cpp](#)

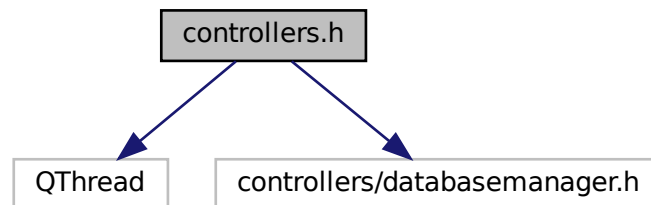
Chapter 6

File Documentation

6.1 controllers.h File Reference

Contains declarations for database manager and related threads. Provides external linkage for DatabaseManager and QThread instances.

```
#include <QThread>
#include "controllers/databasemanager.h"
Include dependency graph for controllers.h:
```



Namespaces

- [Controllers](#)
Contains instances of various controllers used in the application.
- [Threads](#)
Contains instances of QThread used for different managers.

Variables

- DatabaseManager **Controllers::dbManager**

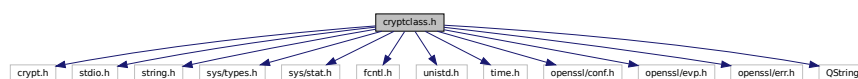
6.1.1 Detailed Description

Contains declarations for database manager and related threads. Provides external linkage for DatabaseManager and QThread instances.

6.2 cryptclass.h File Reference

The [CryptClass](#) class Manges password encryption Manges encrypting user data to database Manges decrypting user data from database Interfaces with dbcontroller.

```
#include <crypt.h>
#include <stdio.h>
#include <string.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <unistd.h>
#include <time.h>
#include <openssl/conf.h>
#include <openssl/evp.h>
#include <openssl/err.h>
#include <QString>
Include dependency graph for cryptclass.h:
```



Classes

- class [CryptClass](#)

Macros

- #define **BCRYPT_HASHSIZE** 512
- #define **RANDBYTES** 16
- #define **MIN_ROUNDS** 4
- #define **MAX_ROUNDS** 31
- #define **DEF_ROUNDS** 12
- #define **BASE64ENCODED_LEN**(textLen) ((4*((textLen+2)/3))+1)
- #define **BASE64DECODED_LEN**(textLen) ((3*textLen/4)+1)
- #define **AES256ENCODED_LEN**(textLen) (((textLen/16)+1)*16)

6.2.1 Detailed Description

The [CryptClass](#) class Manges password encryption Manges encrypting user data to database Manges decrypting user data from database Interfaces with dbcontroller.

Index

- aes256decode
 - [CryptClass, 10](#)
- aes256encode
 - [CryptClass, 10](#)
- aes256fromBase64Str
 - [CryptClass, 10](#)
- aes256fromHexStr
 - [CryptClass, 12](#)
- aes256toBase64Str
 - [CryptClass, 12](#)
- aes256toHexStr
 - [CryptClass, 12](#)

- base64aes256decode
 - [CryptClass, 13](#)
- base64aes256encode
 - [CryptClass, 13](#)
- base64decode
 - [CryptClass, 14](#)
- base64encode
 - [CryptClass, 14](#)
- bcrypt_checkpw
 - [CryptClass, 14](#)
- bcrypt_gensalt
 - [CryptClass, 15](#)
- bcrypt_hashpw
 - [CryptClass, 15](#)

- Controllers, [7](#)
- controllers.h, [19](#)
- CryptClass, [9](#)
 - [aes256decode, 10](#)
 - [aes256encode, 10](#)
 - [aes256fromBase64Str, 10](#)
 - [aes256fromHexStr, 12](#)
 - [aes256toBase64Str, 12](#)
 - [aes256toHexStr, 12](#)
 - [base64aes256decode, 13](#)
 - [base64aes256encode, 13](#)
 - [base64decode, 14](#)
 - [base64encode, 14](#)
 - [bcrypt_checkpw, 14](#)
 - [bcrypt_gensalt, 15](#)
 - [bcrypt_hashpw, 15](#)
 - [rand_read, 16](#)
 - [setlv, 16](#)
 - [setKey, 16](#)
- cryptclass.h, [20](#)

- rand_read
 - [CryptClass, 16](#)
- setlv
 - [CryptClass, 16](#)
- setKey
 - [CryptClass, 16](#)

- Threads, [7](#)