TicTacToe

Generated by Doxygen 1.9.1

Chapter 1

Namespace Index

1.1 Namespace List

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

Chapter 2

Class Index

2.1 Class List

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

Chapter 3

File Index

3.1 File List

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

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 *aes2656Buf, 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

const	char* encBuf
int	encLen
*char**	textBuf

Returns

int

5.1.1.2 aes256encode()

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

AES 256 Encoder.

Parameters

const	char* textBuf
int	textLen
*char**	encBuf

Returns

int

5.1.1.3 aes256fromBase64Str()

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

AES 256 from Base64 String.

Parameters

const	char* charBuf
int	len

Returns

int

5.1.1.4 aes256fromHexStr()

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

AES 256 from Hex String.

Parameters

const	char* charBuf
int	len

Returns

int

5.1.1.5 aes256toBase64Str()

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

AES 256 to Base64 String.

Parameters

const	char* charBuf
int	len

Returns

int

5.1.1.6 aes256toHexStr()

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

AES 256 to Hex String.

Parameters

const	char* charBuf
int	len

Returns

int

5.1.1.7 base64aes256decode()

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

Base64 aes256 Decode.

Parameters

QString	textString

Returns

int

5.1.1.8 base64aes256encode()

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

base64 aes256 encode

Parameters

QString	textString

Returns

int

5.1.1.9 base64decode()

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

Decode String using Base64 Encoder.

Parameters

const	char* encBuf
int	len
*char**	textBuf

Returns

int

5.1.1.10 base64encode()

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

Encode a string using Base64.

Parameters

textBuf	The input text buffer.
len	The length of the input text.
encBuf	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

passwd	The password to check.
hash	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

salt_rounds	The number of salt rounds.
salt	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

passwd	The password to hash.
salt	The salt to use.
hash	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

fd	The file descriptor.
out	The output buffer.
count	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

QString	hexIvString

Returns

void

5.1.1.16 setKey()

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

Create Key (setter for key256)

Parameters

QString	hexKeyString

Returns

void

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

- cryptclass.h
- cryptclass.cpp

Chapter 6

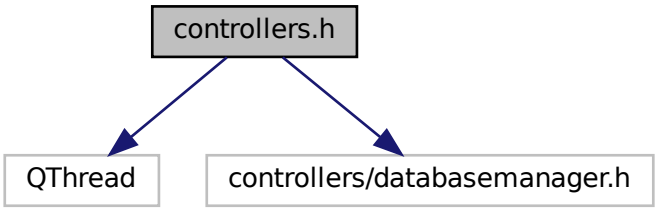
File Documentation

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

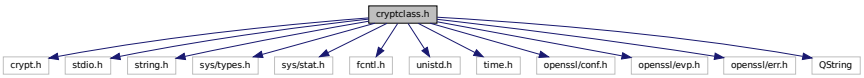
Contains declarations for database manager and related threads. Provides external linkage for DatabaseManager and QThread instances.

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