

QLockButton

Generated by Doxygen 1.8.17

1 QLockButton	1
1.1 TODO	2
1.2 References	2
1.3 License	3
1.4 Greetings	3
2 Hierarchical Index	5
2.1 Class Hierarchy	5
3 Class Index	7
3.1 Class List	7
4 Class Documentation	9
4.1 QLockButton Class Reference	9
4.1.1 Detailed Description	12
4.1.2 Constructor & Destructor Documentation	13
4.1.2.1 QLockButton()	13
4.1.3 Member Function Documentation	13
4.1.3.1 backgroundColor()	13
4.1.3.2 backgroundColorChanged	13
4.1.3.3 borderColor()	13
4.1.3.4 borderColorChanged	14
4.1.3.5 borderWidth()	14
4.1.3.6 borderWidthChanged	14
4.1.3.7 fail	14
4.1.3.8 fillEndColor()	14
4.1.3.9 fillEndColorChanged	15
4.1.3.10 fillStartColor()	15
4.1.3.11 fillStartColorChanged	15
4.1.3.12 innerColor()	15
4.1.3.13 innerColorChanged	16
4.1.3.14 lockTimeout()	16
4.1.3.15 lockTimeoutChanged	16
4.1.3.16 mode()	16
4.1.3.17 modeChanged	16
4.1.3.18 ModeToString()	16
4.1.3.19 mouseMoveEvent()	17
4.1.3.20 mousePressEvent()	17
4.1.3.21 mouseReleaseEvent()	17
4.1.3.22 paintEvent()	18
4.1.3.23 reset()	18
4.1.3.24 resizeEvent()	18
4.1.3.25 setBackgroundColor()	18

4.1.3.26 setBorderColor()	19
4.1.3.27 setBorderWidth()	19
4.1.3.28 setFillEndColor()	19
4.1.3.29 setFillStartColor()	20
4.1.3.30 setInnerColor()	20
4.1.3.31 setLockTimeout()	20
4.1.3.32 setMode()	20
4.1.3.33 setStatus()	21
4.1.3.34 setUnlockTimeout()	21
4.1.3.35 status()	21
4.1.3.36 statusChanged	21
4.1.3.37 StatusToString()	22
4.1.3.38 StringToMode()	22
4.1.3.39 StringToStatus()	22
4.1.3.40 success	23
4.1.3.41 unlockTimeout()	23
4.1.3.42 unlockTimeoutChanged	23
4.2 QLockButtonPlugin Class Reference	24
Index	25

Chapter 1

QLockButton

[QLockButton](#) is a Qt 5 widget that implements an interactive lock button. The widget contains a timer that is activated when the user press inside the button area, if the user keep the button pressed for at given time the button change its state.

This widget can be used to *secure* some action/feature that you don't want to be activated by a random click. The widget provide also a visual indicator of the time needed to fullfill the status change trough a progressive fill effect of the outer rim.

To use this widget you should download the source code of this repository and then you can opt between:

- use the class code directly adding [qlockbutton.h](#), [qlockbutton.cpp](#) files to your code;
- include the class folder adding a `include(qlockbutton/qlockbutton.pri)` directive to your `.pro` file.

Optionally you can compile the main project file to enhance the visual integration experience (loading and building `LockButton.pro` file).

If you have installed succesfully the plugin you can enjoy the full user experience as you can drop [QLockButton](#) into your form straight form the palette in QtCreator (or QtDesigner), there you can manage properties as any other widget. If you are using [QLockButton](#) trough source code then you can add the widget to the UI programatically or *promoting* a widget *placeholder* to [QLockButton](#) class (referencing its header: [qlockbutton.h](#) of course). You don't need to make the *placeholder* square since [QLockButton](#) is beend designed to always *use* a square footprint (inside assigned widget area). To be notified of user interactions connect to the `success` and/or `fail` signals. Every successul attempt to change the state of the button (i.e. keeping the button pressed for the time needed) is notified trough `success` signal along with the state reached by the button (`Locked`cor `Unlocked`). If the user fails then the corresponding signal is fired (it will also report the missing time to a successfull state switch).

You can customize the look and behaviour of the widget through its properties. You can find an example of the code needed to use this widget in the associated Qt application (`Test.pro`).

For the sake of documentation this is a complete list of [QLockButton](#) widget properties and their usage:

- `QColor backgroundColor`: is the main color of the background (default: `Qt::white`).
- `QColor innerColor`: is the main color of the button *actuator* (default: `Qt::blue`).
- `QColor borderColor`: is the color of the external border of both background and *actuator* area (default: `Qt::black`).

- `int borderWidth`: is the thickness of border (default: 1).
- `int lockTimeout`: is the time needed to get to `Locked` status in milliseconds (default: 3000).
- `int UnlockTimeout`: is the time needed to get to `Unlocked` status in milliseconds (default: 3000).
- `Mode mode`: this property tells if the widget can be switched many times (`MultiShot`, default) or just one (`SingleShot`)
- `State state`: this property read/set the internal status. You can choose between none (`Unlocked`, default) or (`Locked`).
- `QColor fillStartColor`: is the color of the start of the fill effect gradient (default `Qt::white`).
- `QColor fillEndColor`: is the color of the end of the fill effect gradient (default `Qt::red`).

There are also a number of static constants made public for reference use:

- `DEF_INNER_RATIO` (actuator size ratio)
- `DEF_GLYPH_RATIO` (lock/unlock glyph size ratio)
- `DEF_TIMEOUT` (default timeout for both Locking and Unlocking)
- `MAX_TIMEOUT` (maximum timeout value)
- `MIN_TIMEOUT` (minimum timeout value)
- `DEF_UPDATE_TIMEOUT` (this is update time in milliseconds of the fill effect)

If you are guessing how this widget may look like:

This is the widget running in its associated Qt application.

1.1 TODO

- Distribute as library/plugin
- Reverse fill direction (CW instead CCW)
- Translation
- Docs (doxygen)
- Add support for stylesheets
- ~~Improve lock/unlock glyph rendering quality~~
- More features...

1.2 References

- [Qt framework](#)
- [Google Material Icons](#) : lock/unlock icon.

1.3 License

Copyright 2023 Michele Santucci. This product is released under MIT License.

1.4 Greetings

Thanks to everyone collaborating. #

Live long and prosper!

That's all folks.

Bye by[t]e[s]...Weirdgyn!

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

QDesignerCustomWidgetInterface	
QLockButtonPlugin	24
QObject	
QLockButtonPlugin	24
QWidget	
QLockButton	9

Chapter 3

Class Index

3.1 Class List

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

QLockButton		
QLockButton class	9
QLockButtonPlugin	24

Chapter 4

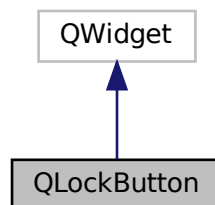
Class Documentation

4.1 QLockButton Class Reference

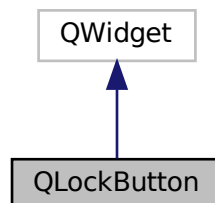
[QLockButton](#) class.

```
#include <qlockbutton.h>
```

Inheritance diagram for QLockButton:



Collaboration diagram for QLockButton:



Public Types

- enum [Mode](#) { **SingleShot** = 0, **MultiShot** }
LockButton function Mode.
- enum [Status](#) { **Unlocked** = 0, **Locked** }
LockButton status.

Signals

- void [success](#) ([QLockButton::Status](#) status)
signal successful change in status.
- void [fail](#) (int remainingTime)
signal unsuccessful attempt.
- void [lockTimeoutChanged](#) ()
signal change in lockTimeout property value.
- void [unlockTimeoutChanged](#) ()
signal change in unlockTimeout property value.
- void [borderWidthChanged](#) ()
signal change in borderWidth property value.
- void [statusChanged](#) ()
signal change in status property value.
- void [modeChanged](#) ()
signal change in mode property value.
- void [innerColorChanged](#) ()
signal innerColor in mode property value.
- void [backgroundColorChanged](#) ()
signal backgroundColor in mode property value.
- void [borderColorChanged](#) ()
signal borderColor in mode property value.
- void [fillStartColorChanged](#) ()
signal fillStartColor in mode property value.
- void [fillEndColorChanged](#) ()
signal fillEndColor in mode property value.

Public Member Functions

- [QLockButton](#) (QWidget *parent=0)
Construct a new [QLockButton](#) object.
- [~QLockButton](#) ()
Destroy the [QLockButton](#) object.
- int [lockTimeout](#) () const
lockTimeout property getter.
- void [setLockTimeout](#) (int lockTimeout)
lockTimeout property setter.
- int [unlockTimeout](#) () const
unlockTimeout property getter.
- void [setUnlockTimeout](#) (int unlockTimeout)
unlockTimeout property setter.
- int [borderWidth](#) () const

- borderWidth property getter.*
- void [setBorderWidth](#) (int borderWidth)
borderWidth property setter.
- [Status](#) [status](#) () const
status property getter.
- void [setStatus](#) ([Status](#) status)
status property setter.
- [Mode](#) [mode](#) () const
mode property getter.
- void [setMode](#) ([Mode](#) mode)
mode property setter.
- void [reset](#) ()
reset internal status.
- QColor [innerColor](#) () const
innerColor property getter.
- void [setInnerColor](#) (const QColor &innerColor)
innerColor property setter.
- QColor [backgroundColor](#) () const
backgroundColor property getter.
- void [setBackgroundColor](#) (const QColor &backgroundColor)
backgroundColor property setter.
- QColor [borderColor](#) () const
borderColor property getter.
- void [setBorderColor](#) (const QColor &borderColor)
borderColor property setter.
- QColor [fillStartColor](#) () const
fillStartColor property getter.
- void [setFillStartColor](#) (const QColor &fillStartColor)
fillStartColor property setter.
- QColor [fillEndColor](#) () const
fillEndColor property getter.
- void [setFillEndColor](#) (const QColor &fillEndColor)
fillEndColor property setter.

Static Public Member Functions

- static QString [ModeToString](#) ([Mode](#) mode)
Convert argument into a readable string.
- static [Mode](#) [StringToMode](#) (QString mode)
Convert argument string in a Mode value.
- static QString [StatusToString](#) ([Status](#) status)
Convert argument into a readable string.
- static [Status](#) [StringToStatus](#) (QString status)
Convert argument in a Status value.

Static Public Attributes

- constexpr static const float [DEF_INNER_RATIO](#) = 0.8f
Size ratio of inner area (button actuator) against the full widget frame.
- constexpr static const float [DEF_GLYPH_RATIO](#) = 0.8f
Size ratio of glyph/symbol ratio relative to inner area.
- constexpr static const int [DEF_TIMEOUT](#) = 3000
Default timeout value for both locking and unlocking (ms).
- constexpr static const int [MAX_TIMEOUT](#) = 3000
Maximum timeout value for locking and unlocking (ms).
- constexpr static const int [MIN_TIMEOUT](#) = 500
Minimum timeout value for locking and unlocking (ms).
- constexpr static const int [DEF_UPDATE_TIMEOUT](#) = 50
Timeout value for paint update (only used for fill effect).

Protected Member Functions

- void [resizeEvent](#) (QResizeEvent *event)
override of base class resizeEvent.
- void [paintEvent](#) (QPaintEvent *event)
override of base class paintEvent.
- void [mousePressEvent](#) (QMouseEvent *event)
override of base class mousePressEvent
- void [mouseReleaseEvent](#) (QMouseEvent *event)
override of base class mouseReleaseEvent
- void [mouseMoveEvent](#) (QMouseEvent *event)
override of base class mouseReleaseEvent

Properties

- int **lockTimeout**
- int **unlockTimeout**
- int **borderWidth**
- [Status](#) **status**
- [Mode](#) **mode**
- QColor **innerColor**
- QColor **backgroundColor**
- QColor **borderColor**
- QColor **fillStartColor**
- QColor **fillEndColor**

4.1.1 Detailed Description

[QLockButton](#) class.

[QLockButton](#) implements a two state, time based lock button.

4.1.2 Constructor & Destructor Documentation

4.1.2.1 QLockButton()

```
QLockButton::QLockButton (
    QWidget * parent = 0 )
```

Construct a new [QLockButton](#) object.

Parameters

<i>parent</i>	parent object
---------------	---------------

4.1.3 Member Function Documentation

4.1.3.1 backgroundColor()

```
QColor QLockButton::backgroundColor ( ) const
```

backgroundColor property getter.

Returns

QColor backgroundColor value

4.1.3.2 backgroundColorChanged

```
void QLockButton::backgroundColorChanged ( ) [signal]
```

signal backgroundColor in mode property value.

4.1.3.3 borderColor()

```
QColor QLockButton::borderColor ( ) const
```

borderColor property getter.

Returns

QColor borderColor value

4.1.3.4 borderColorChanged

```
void QLockButton::borderColorChanged ( ) [signal]
```

signal borderColor in mode property value.

4.1.3.5 borderWidth()

```
int QLockButton::borderWidth ( ) const
```

borderWidth property getter.

Returns

int borderWidth value

4.1.3.6 borderWidthChanged

```
void QLockButton::borderWidthChanged ( ) [signal]
```

signal change in borderWidth property value.

4.1.3.7 fail

```
void QLockButton::fail (
    int remainingTime ) [signal]
```

signal unsuccessful attempt.

Parameters

<i>remainingTime</i>	time missing to a successfull status change
----------------------	---

4.1.3.8 fillEndColor()

```
QColor QLockButton::fillEndColor ( ) const
```

fillEndColor property getter.

Returns

QColor fillEndColor value

4.1.3.9 fillEndColorChanged

```
void QLockButton::fillEndColorChanged ( ) [signal]
```

signal fillEndColor in mode property value.

4.1.3.10 fillStartColor()

```
QColor QLockButton::fillStartColor ( ) const
```

fillStartColor property getter.

Returns

QColor

4.1.3.11 fillStartColorChanged

```
void QLockButton::fillStartColorChanged ( ) [signal]
```

signal fillStartColor in mode property value.

4.1.3.12 innerColor()

```
QColor QLockButton::innerColor ( ) const
```

innerColor property getter.

Returns

QColor innerColor value.

4.1.3.13 innerColorChanged

```
void QLockButton::innerColorChanged ( ) [signal]
```

signal innerColor in mode property value.

4.1.3.14 lockTimeout()

```
int QLockButton::lockTimeout ( ) const
```

lockTimeout property getter.

Returns

int lockTimeout value.

4.1.3.15 lockTimeoutChanged

```
void QLockButton::lockTimeoutChanged ( ) [signal]
```

signal change in lockTimeout property value.

4.1.3.16 mode()

```
Mode QLockButton::mode ( ) const
```

mode property getter.

Returns

Mode mode value

4.1.3.17 modeChanged

```
void QLockButton::modeChanged ( ) [signal]
```

signal change in mode property value.

4.1.3.18 ModeToString()

```
QString QLockButton::ModeToString (
    Mode mode ) [static]
```

Convert argument into a readable string.

Parameters

<i>mode</i>	
-------------	--

Returns

QString

4.1.3.19 mouseMoveEvent()

```
void QLockButton::mouseMoveEvent (
    QMouseEvent * event ) [protected]
```

override of base class mousePressEvent

N/A

Parameters

<i>event</i>	event descriptor
--------------	------------------

4.1.3.20 mousePressEvent()

```
void QLockButton::mousePressEvent (
    QMouseEvent * event ) [protected]
```

override of base class mousePressEvent

Invoked when user press mouse button/touch inside widget area. This action start the status change evaluation process/timeout.

Parameters

<i>event</i>	Event descriptor
--------------	------------------

4.1.3.21 mouseReleaseEvent()

```
void QLockButton::mouseReleaseEvent (
    QMouseEvent * event ) [protected]
```

override of base class mouseReleaseEvent

Invoked when the user release mouse button/touch inside widget area. This action usually reports an unsuccessful status change.

Parameters

<i>event</i>	event descriptor
--------------	------------------

4.1.3.22 paintEvent()

```
void QLockButton::paintEvent (  
    QPaintEvent * event ) [protected]
```

override of base class paintEvent.

Invoked when widget area need to be repainted. Forced by repaint() method calls. Will depict widget current status.

Parameters

<i>event</i>	Event descriptor
--------------	------------------

4.1.3.23 reset()

```
void QLockButton::reset ( )
```

reset internal status.

Reset internal object status to allow further re-triggers (only when used in SingleShot mode)

4.1.3.24 resizeEvent()

```
void QLockButton::resizeEvent (  
    QResizeEvent * event ) [protected]
```

override of base class resizeEvent.

Invoked when widget area is resized. Will update internal sub-areas size accordingly with defined ratios.

Parameters

<i>event</i>	Event descriptor
--------------	------------------

4.1.3.25 setBackgroundColor()

```
void QLockButton::setBackgroundColor (  
    const QColor & backgroundColor )
```

backgroundColor property setter.

Parameters

<i>backgroundColor</i>	new value
------------------------	-----------

4.1.3.26 setBorderColor()

```
void QLockButton::setBorderColor (
    const QColor & borderColor )
```

borderColor property setter.

Parameters

<i>borderColor</i>	new value
--------------------	-----------

4.1.3.27 setBorderWidth()

```
void QLockButton::setBorderWidth (
    int borderWidth )
```

borderWidth property setter.

Parameters

<i>borderWidth</i>	new value
--------------------	-----------

4.1.3.28 setFillEndColor()

```
void QLockButton::setFillEndColor (
    const QColor & fillEndColor )
```

fillEndColor property setter.

Parameters

<i>fillEndColor</i>	new value
---------------------	-----------

4.1.3.29 setFillStartColor()

```
void QLockButton::setFillStartColor (
    const QColor & fillStartColor )
```

fillStartColor property setter.

Parameters

<i>fillStartColor</i>	new value
-----------------------	-----------

4.1.3.30 setInnerColor()

```
void QLockButton::setInnerColor (
    const QColor & innerColor )
```

innerColor property setter.

Parameters

<i>innerColor</i>	new value
-------------------	-----------

4.1.3.31 setLockTimeout()

```
void QLockButton::setLockTimeout (
    int lockTimeout )
```

lockTimeout property setter.

Parameters

<i>lockTimeout</i>	new value
--------------------	-----------

4.1.3.32 setMode()

```
void QLockButton::setMode (
    Mode mode )
```

mode property setter.

Parameters

<i>mode</i>	new value
-------------	-----------

4.1.3.33 setStatus()

```
void QLockButton::setStatus (
    Status status )
```

status property setter.

Parameters

<i>status</i>	new value
---------------	-----------

4.1.3.34 setUnlockTimeout()

```
void QLockButton::setUnlockTimeout (
    int unlockTimeout )
```

unlockTimeout property setter.

Parameters

<i>unlockTimeout</i>	new value
----------------------	-----------

4.1.3.35 status()

```
Status QLockButton::status ( ) const
```

status property getter.

Returns

Status status value

4.1.3.36 statusChanged

```
void QLockButton::statusChanged ( ) [signal]
```

signal change in status property value.

4.1.3.37 StatusToString()

```
QString QLockButton::StatusToString (
    Status status ) [static]
```

Convert argument into a readable string.

Parameters

<i>status</i>	
---------------	--

Returns

QString

4.1.3.38 StringToMode()

```
QLockButton::Mode QLockButton::StringToMode (
    QString mode ) [static]
```

Convert argument string in a Mode value.

Parameters

<i>mode</i>	
-------------	--

Returns

Mode

4.1.3.39 StringToStatus()

```
QLockButton::Status QLockButton::StringToStatus (
    QString status ) [static]
```

Convert argument in a Status value.

Parameters

<i>status</i>	
---------------	--

Returns

Status

4.1.3.40 success

```
void QLockButton::success (
    QLockButton::Status status ) [signal]
```

signal successful change in status.

Parameters

<i>status</i>	reached status
---------------	----------------

4.1.3.41 unlockTimeout()

```
int QLockButton::unlockTimeout ( ) const
```

unlockTimeout property getter.

Returns

int unlockTimeout value

4.1.3.42 unlockTimeoutChanged

```
void QLockButton::unlockTimeoutChanged ( ) [signal]
```

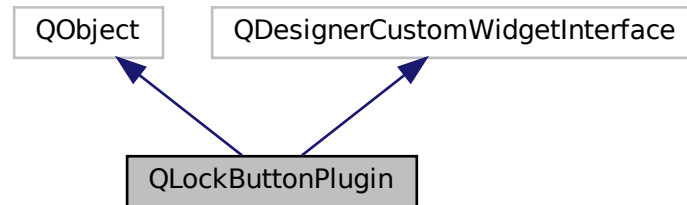
signal change in unlockTimeout property value.

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

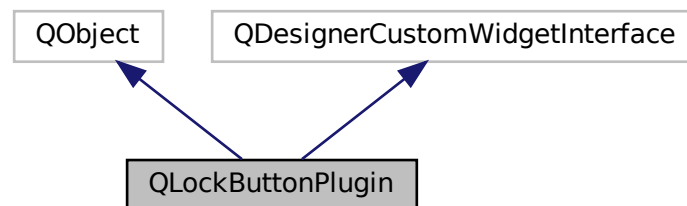
- qlockbutton/qlockbutton.h
- qlockbutton/qlockbutton.cpp

4.2 QLockButtonPlugin Class Reference

Inheritance diagram for QLockButtonPlugin:



Collaboration diagram for QLockButtonPlugin:



Public Member Functions

- **QLockButtonPlugin** (QObject *parent=0)
- bool **isContainer** () const
- bool **isInitialized** () const
- QIcon **icon** () const
- QString **domXml** () const
- QString **group** () const
- QString **includeFile** () const
- QString **name** () const
- QString **toolTip** () const
- QString **whatsThis** () const
- QWidget * **createWidget** (QWidget *parent)
- void **initialize** (QDesignerFormEditorInterface *core)

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

- qlockbuttonplugin.h
- qlockbuttonplugin.cpp

Index

- backgroundColor
 - QLockButton, [13](#)
- backgroundColorChanged
 - QLockButton, [13](#)
- borderColor
 - QLockButton, [13](#)
- borderColorChanged
 - QLockButton, [13](#)
- borderWidth
 - QLockButton, [14](#)
- borderWidthChanged
 - QLockButton, [14](#)
- fail
 - QLockButton, [14](#)
- fillEndColor
 - QLockButton, [14](#)
- fillEndColorChanged
 - QLockButton, [15](#)
- fillStartColor
 - QLockButton, [15](#)
- fillStartColorChanged
 - QLockButton, [15](#)
- innerColor
 - QLockButton, [15](#)
- innerColorChanged
 - QLockButton, [15](#)
- lockTimeout
 - QLockButton, [16](#)
- lockTimeoutChanged
 - QLockButton, [16](#)
- mode
 - QLockButton, [16](#)
- modeChanged
 - QLockButton, [16](#)
- ModeToString
 - QLockButton, [16](#)
- mouseMoveEvent
 - QLockButton, [17](#)
- mousePressEvent
 - QLockButton, [17](#)
- mouseReleaseEvent
 - QLockButton, [17](#)
- paintEvent
 - QLockButton, [18](#)
- QLockButton, [9](#)

- backgroundColor, [13](#)
- backgroundColorChanged, [13](#)
- borderColor, [13](#)
- borderColorChanged, [13](#)
- borderWidth, [14](#)
- borderWidthChanged, [14](#)
- fail, [14](#)
- fillEndColor, [14](#)
- fillEndColorChanged, [15](#)
- fillStartColor, [15](#)
- fillStartColorChanged, [15](#)
- innerColor, [15](#)
- innerColorChanged, [15](#)
- lockTimeout, [16](#)
- lockTimeoutChanged, [16](#)
- mode, [16](#)
- modeChanged, [16](#)
- ModeToString, [16](#)
- mouseMoveEvent, [17](#)
- mousePressEvent, [17](#)
- mouseReleaseEvent, [17](#)
- paintEvent, [18](#)
- QLockButton, [13](#)
- reset, [18](#)
- resizeEvent, [18](#)
- setBackgroundColor, [18](#)
- setBorderColor, [19](#)
- setBorderWidth, [19](#)
- setFillEndColor, [19](#)
- setFillStartColor, [19](#)
- setInnerColor, [20](#)
- setLockTimeout, [20](#)
- setMode, [20](#)
- setStatus, [21](#)
- setUnlockTimeout, [21](#)
- status, [21](#)
- statusChanged, [21](#)
- StatusToString, [21](#)
- StringToMode, [22](#)
- StringToStatus, [22](#)
- success, [23](#)
- unlockTimeout, [23](#)
- unlockTimeoutChanged, [23](#)
- QLockButtonPlugin, [24](#)
- reset
 - QLockButton, [18](#)
- resizeEvent
 - QLockButton, [18](#)

- setBackgroundColor
 - QLockButton, [18](#)
- setBorderColor
 - QLockButton, [19](#)
- setBorderWidth
 - QLockButton, [19](#)
- setFillEndColor
 - QLockButton, [19](#)
- setFillStartColor
 - QLockButton, [19](#)
- setInnerColor
 - QLockButton, [20](#)
- setLockTimeout
 - QLockButton, [20](#)
- setMode
 - QLockButton, [20](#)
- setStatus
 - QLockButton, [21](#)
- setUnlockTimeout
 - QLockButton, [21](#)
- status
 - QLockButton, [21](#)
- statusChanged
 - QLockButton, [21](#)
- StatusToString
 - QLockButton, [21](#)
- StringToMode
 - QLockButton, [22](#)
- StringToStatus
 - QLockButton, [22](#)
- success
 - QLockButton, [23](#)
- unlockTimeout
 - QLockButton, [23](#)
- unlockTimeoutChanged
 - QLockButton, [23](#)