QRegularExpressionValidator Class

QRegularExpressionValidator 类用于根据正则表达式检查字符串。更多的...

Header:	**#include**
CMake:	find_package(Qt6 REQUIRED COMPONENTS Gui) target_link_libraries(mytarget PRIVATE Qt6::Gui)
qmake:	QT += gui
Inherits:	QValidator

- 所有成员的列表，包括继承的成员

特性

- **regularExpression** : QRegularExpression

公共方法

	QRegularExpressionValidator(QObject *parent = nullptr)
	QRegularExpressionValidator(const QRegularExpression &*re*, QObject *parent = nullptr)
virtual	~QRegularExpressionValidator()
QRegularExpression	regularExpression() const

重载的公共方法

virtual QValidator::State	**validate(QString &*input*, int &*pos*) const override**

公共槽

void	**setRegularExpression(const QRegularExpression &*re*)**

void **setRegularExpression**(**const QRegularExpression &*re***)

信号

void **regularExpressionChanged**(**const QRegularExpression &*re***)

详细说明

QRegularExpressionValidator 使用正则表达式（regexp）来判断输入字符串是否为Acceptable,Intermediate， 或者Invalid。正则表达式可以在构造 QRegularExpressionValidator 时提供，也可以在稍后提供。

如果正则表达式与字符串部分匹配，则认为结果Intermediate。例如，“”和“A”是Intermediate对于正则表达式**[AZ][0-9]** （ 而 “*” 将 是 [Invalid](https://doc-qt-io.translate.goog/qt-6/qvalidator.html?x_tr_sl=auto&x_tr_tl=zh-CN&x_tr_hl=zh-CN&_x_tr_pto=wapp#State-enum)）。

QRegularExpressionValidator 自动将正则表达式包装在\\A和\\z锚点中；换句话说，它总是尝试进行精确匹配。

使用示例：

```
// regexp: optional '-' followed by between 1 and 3 digits
QRegularExpression rx("-?\\d{1,3}");
QValidator *validator = new QRegularExpressionValidator(rx, this);

QLineEdit *edit = new QLineEdit(this);
edit->setValidator(validator);
```

下面我们介绍一些验证器的示例。在实践中，它们通常与一个小部件相关联，如上例所示。

```
// integers 1 to 9999
QRegularExpression re("[1-9]\\d{0,3}");
// the validator treats the regexp as "^[1-9]\\d{0,3}$"
QRegularExpressionValidator v(re, 0);
QString s;
int pos = 0;

s = "0";     v.validate(s, pos);    // returns Invalid
s = "12345"; v.validate(s, pos);    // returns Invalid
s = "1";     v.validate(s, pos);    // returns Acceptable

re.setPattern("\\S+");            // one or more non-whitespace characters
v.setRegularExpression(re);
s = "myfile.txt";  v.validate(s, pos); // Returns Acceptable
s = "my file.txt"; v.validate(s, pos); // Returns Invalid

// A, B or C followed by exactly five digits followed by W, X, Y or Z
re.setPattern("[A-C]\\d{5}[W-Z]");
v.setRegularExpression(re);
```

```
s = "a12345Z"; v.validate(s, pos);        // Returns Invalid
s = "A12345Z"; v.validate(s, pos);        // Returns Acceptable
s = "B12";     v.validate(s, pos);        // Returns Intermediate

// match most 'readme' files
re.setPattern("read\\S?me(\\.(txt|asc|1st))?");
re.setPatternOptions(QRegularExpression::CaseInsensitiveOption);
v.setRegularExpression(re);
s = "readme";      v.validate(s, pos); // Returns Acceptable
s = "README.1ST";  v.validate(s, pos); // Returns Acceptable
s = "read me.txt"; v.validate(s, pos); // Returns Invalid
s = "readm";       v.validate(s, pos); // Returns Intermediate
```

可以参阅QRegularExpression,QIntValidator， 和QDoubleValidator。

属性文档

regularExpression : QRegularExpression

该属性保存用于验证的正则表达式

默认情况下，此属性包含具有空模式的正则表达式（因此可以匹配任何字符串）。

访问功能:

QRegularExpression	**regularExpression() const**
void	**setRegularExpression**(const QRegularExpression &*re*)

通知器信号:

void	**regularExpressionChanged(const QRegularExpression &*re*)**

成员函数文档

*[explicit]QRegularExpressionValidator::QRegularExpressionValidator(QObject *parent = nullptr)*

构造一个验证器*parent*接受任何字符串（包括空字符串）作为有效字符串的对象。

*[explicit]QRegularExpressionValidator::QRegularExpressionValidator(const QRegularExpression &re, QObject *parent = nullptr)*

构造一个验证器*parent*接受与正则表达式匹配的所有字符串的对象*re*。

[virtual]QRegularExpressionValidator::~QRegularExpressionValidator()

销毁验证器。

[override virtual]QValidator::State QRegularExpressionValidator::validate(QString &input, int &pos) const

重新实现：QValidator::validate(QString &input, int &pos) const。

退货Acceptable如果*input*与该验证器的正则表达式匹配，Intermediate如果它已部分匹配（即如果添加其他有效字符则可能是有效匹配），并且Invalid如果*input*不匹配。

如果*input*不匹配，则*pos*参数设置为长度*input*范围; 否则不予修改。

例如，如果正则表达式为**\w\d\d**（单词字符、数字、数字），则“A57”为Acceptable，“E5”是Intermediate，“+9”是Invalid。

可以参阅QRegularExpression::match()。