

Editing Markdown with QTextDocument Widgets and Qt Quick

Shawn Rutledge

`shawn.rutledge@qt.io`
ecloud on `#qt-labs`, `#qt-quick` etc.



About me

- Qt user since ~2004
- The Qt Company - Oslo since 2011
- Pointing devices: touch, Wacom tablets
- Linux/X11 and macOS
- QtPDF
- Qt Quick, Controls and Dialogs



4 - 11 September 2020



```
Text {  
    width: 500  
    y: 100  
    textFormat: Text.MarkdownText  
    wrapMode: Text.WordWrap  
    font.pointSize: 16  
    // onLinkHovered: statusText.text = link  
    onLinkActivated: Qt.openUrlExternally(link)  
    text:  
    "Yes *this* is [Markdown](https://spec.commonmark.org/0.29/) not ~~plain text~~\n  
- bullets?\n  + nested bullets?\n- [ ] checkboxes?\n- [x] yes of course"  
}
```

Yes *this* is [Markdown](#) not plain text

- bullets?
 - nested bullets?
- ☐ checkboxes?
- ☒ yes of course

Motivations

- QTD supports little of HTML, but all of Markdown
- Good default text format
- WYSIWYG and text-editing approaches equally good
- Applications:
 - documentation
 - notes, TODO lists
 - email viewing and composition
 - blogging, writing, micro-publishing
 - <your idea here>



4 - 11 September 2020



Markdown Features

- `QTextDocument::setMarkdown()`, `toMarkdown()`
- GitHub dialect by default
 - tables, checkboxes, strikethrough, permissive autolinks
- CommonMark, NoHTML (restricted features)
 - `QTextDocument::setMarkdown(MarkdownDialectCommonMark | MarkdownNoHTML)`
- Thanks to Martin Mitáš for md4c
 - <https://github.com/mity/md4c> v0.4.3 currently



4 - 11 September 2020

Demo: Controls 2 example

```
import QtQuick 2.14
import QtQuick.Controls 2.14
```

```
ApplicationWindow {
    width: 800
    height: 600
    visible: true

    ScrollView {
        anchors.fill: parent

        TextArea {
            id: textEdit
            textFormat: Text.MarkdownText
            wrapMode: Text.WordWrap
            font.pointSize: 12
            onLinkHovered: statusText.text = link
            onLinkActivated: Qt.openUrlExternally(link)

            Component.onCompleted: {
                var request = new XMLHttpRequest()
                request.open('GET', 'example.md')
                request.onreadystatechange = function(event) {
                    if (request.readyState === XMLHttpRequest.DONE)
                        textEdit.text = request.responseText
                }
                request.send()
            }
        }
    }

    Rectangle {
        anchors.bottom: parent.bottom
        width: statusText.width + 12
        height: statusText.height + 4
        Text {

```

Demo: Nettebook



4 - 11 September 2020

Stuff left to work on

- another BlockMarker to decorate block quotes
- CSS (no promises!)
- keep fixing bugs



4 - 11 September 2020



4 - 11 September 2020

Editing Markdown with QTextDocument Widgets and Qt Quick

Shawn Rutledge

`shawn.rutledge@qt.io`
ecloud on `#qt-labs`, `#qt-quick` etc.