Generate Material Theme QSS

from python lib qt-material

彭浩

2024年7月1日

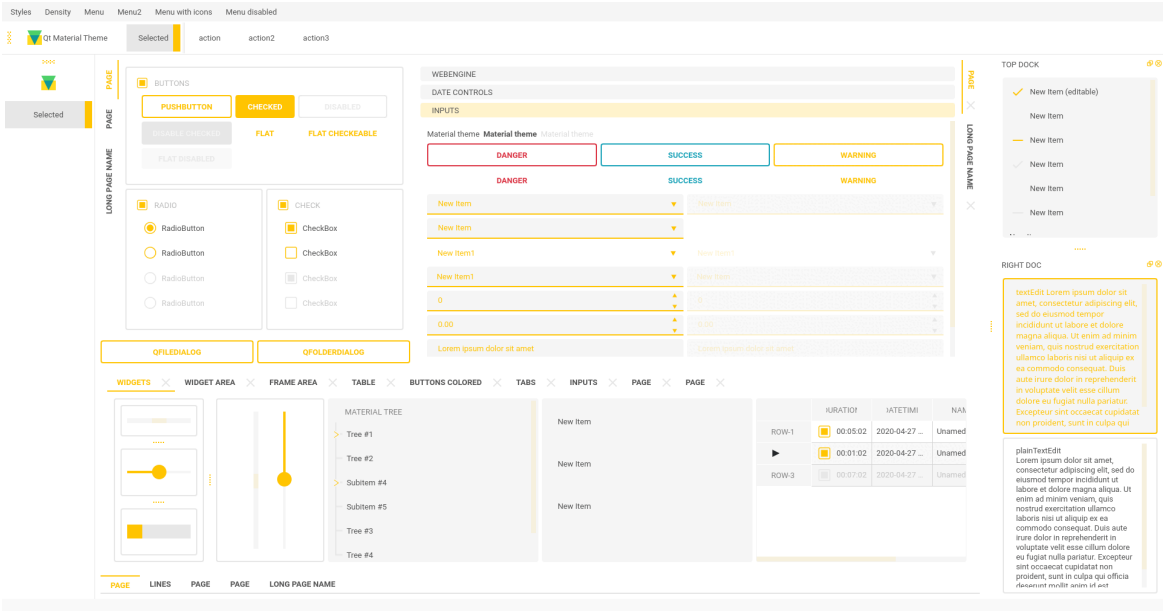
1. Material 风格的实现方式

Material 风格是 Google 推出的一种 ui 设计原则，其界面清爽美观，在网页、app 和GUI设计中得到了广泛应用。在 Qt 中有多种方式可以实现 material 风格的界面，包括 Qt Material Widget 组件库、Qt Quick 中引入 material 风格 以及使用 material 风格的 QSS 样式表。

其中 Qt Material Widgets 组件库重写了 Qt 的基础组件，在通过设计师模式使用前需要先将基础组件提升为 material 组件，使用时较为不便，且该库在 4 年前就已经停止维护，故不作推荐。

此外，Qt Quick模式中以QML编写界面，不能兼容传统设计师模式，失去了一些灵活性。因此，对于传统的 Qt Widget 设计模式，使用 QSS 样式表对基础组件进行渲染是较好的方案，其将界面布局和界面渲染相分离，可以方便地更换渲染主题。

需要注意的是，网络上 Material 风格的 QSS 样式表较少。python 的material 风格的界面库 qt-material 中提供了导出 QSS 样式表的方法，本文在这一方法的基础上，提取了 27 种不同主题的 QSS 样式表。



qt-material说明文档　https://qt-material.readthedocs.io/en/latest/index.html

2. QSS导出程序

qt-material 类库说明文档中给出了导出 QSS 文件的方法，如下所示:

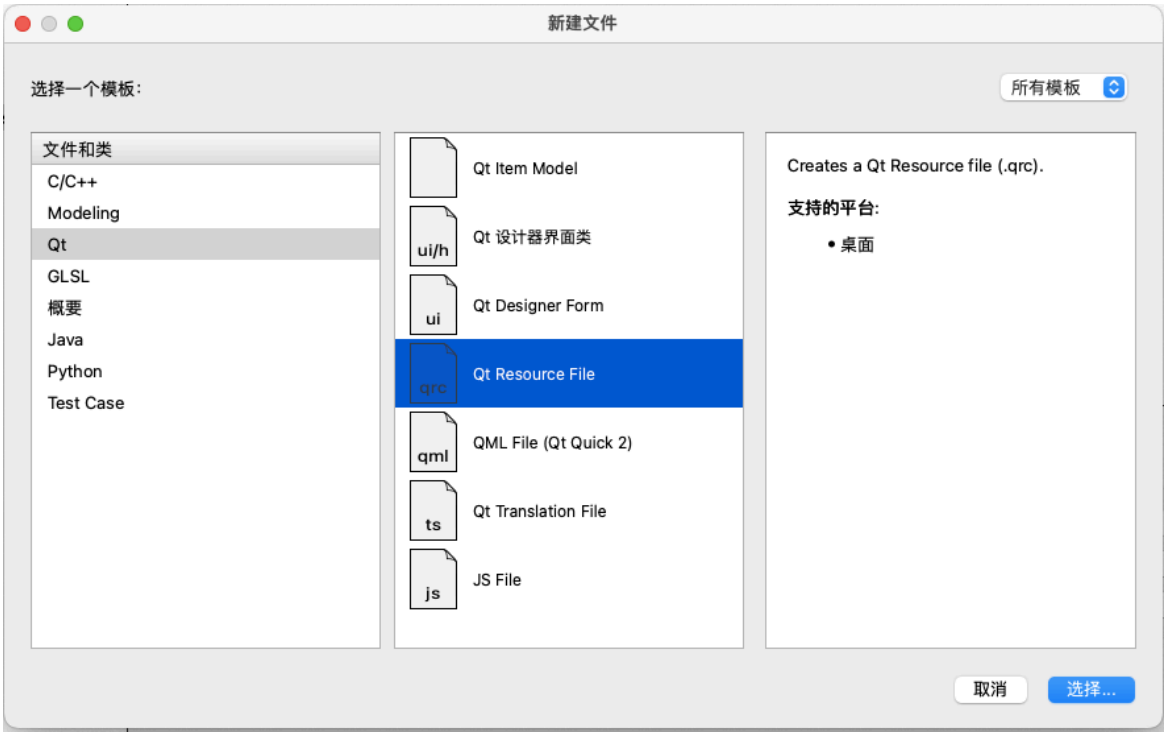
```
from qt_material import export_theme
extra = {

    # Button colors
    'danger': '#dc3545',
    'warning': '#ffc107',
    'success': '#17a2b8',

    # Font
    'font_family': 'monoespace',
    'font_size': '13px',
    'line_height': '13px',

    # Density Scale
    'density_scale': '0',

    # environ
    'pyside6': True,
    'linux': True,
}
export_theme(theme='dark_teal.xml',
             qss='dark_teal.qss',
             rcc='resources.rcc',
             output='theme',
             prefix='icon:/',
             invert_secondary=False,
             extra=extra)
            )
```

注意该示例主要是用于 pyside6 类库的，并不能直接用于 Qt C++ 应用。例如生成的 qss 文件中 icon 图片的 url 为 'icon:/xxx/xxx'，而 Qt C++ 中对资源文件正确的引用应该为 ':icon/xxx/xxx'。此外该方法生成的 .qrc 文件中缺少了 active 模式的图片，因此对代码作了一些改写，得到新的导出方法，代码及注释见 qss.py.

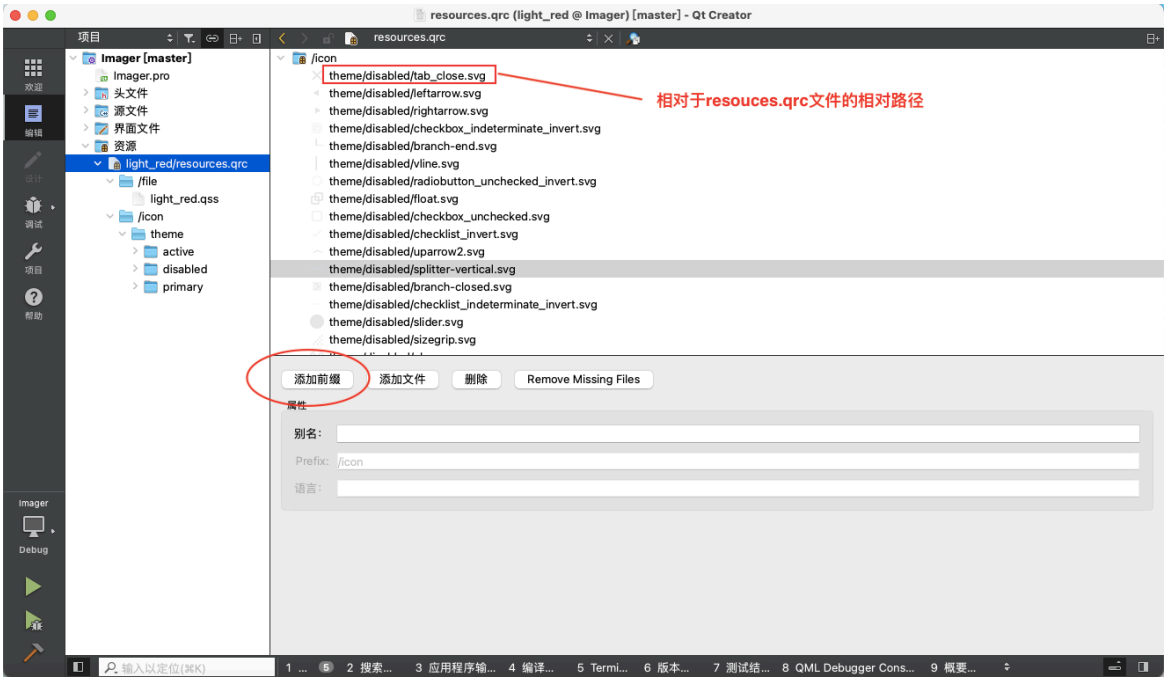
导出的文件包括 .qss 样式表、.rcc 资源文件和一个 theme 文件夹，theme 中包含所需的 icon 图片。其中 invert_secondary 选项用于指定暗亮风格，light风格为 True，dark 风格为 False.

3. 导入资源文件(常规)

首先新建 Qt Resource File，如下图所示:



在创建好的 .qrc 文件中添加前缀，如左边项目栏中的/file、/icon前缀。需要注意的是，这里的前缀只是一个逻辑上的分组，并不代表实际的磁盘目录结构。

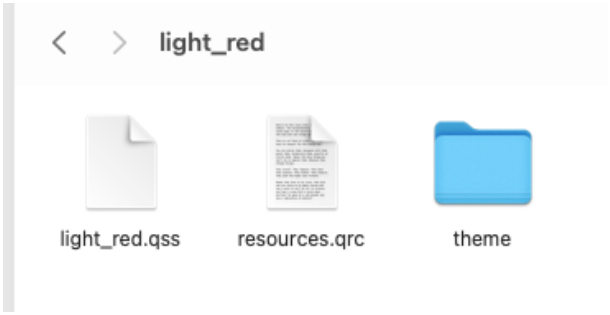


在相应的前缀下添加完资源文件后，就可以在代码中访问这些资源文件了。这些文件会一同编译到可执行文件中，因此可以按照 :+ 前缀名 + 路径 的规则访问，例如通过 QFile qssFile(":file/light_red.qss") 语句读取 QSS 样式表。

虽然在本地调试时，可以直接访问到本地的文件，但是强烈建议将样式表和图片添加到 Qt 项目的资源中，这些资源会编译成 qrc_resources.cpp 文件并打包到可执行文件中，从而在应用发布后也能正常显示资源。

4. 将 Material QSS 导入 Qt 项目

执行 qss.py 程序后，会得到一个相应主题名的文件夹。以 light_red 主题为例，在 light_red 文件夹下包含了样式表 qss、资源文件 qrc 和图片资源包 theme.



以上资源文件都已经配置完毕，直接将 light_red 文件夹复制到相应的 Qt 工程目录下，在工程文件 .pro 中添加资源文件的路径信息，如下图:

```
SOURCES += \
    main.cpp \
    mainwindow.cpp

HEADERS += \
    mainwindow.h

FORMS += \
    mainwindow.ui

# Default rules for deployment.
qnx: target.path = /tmp/$${TARGET}/bin
else: unix:!android: target.path = /opt/$${TARGET}/bin
!isEmpty(target.path): INSTALLS += target

RESOURCES += \
    light_red/resources.qrc
```

重新编译项目即可，如果 Material 风格仍未生效，可以清除项目再重新构建。