Radiooooo Qt

1.0

Создано системой Doxygen 1.9.1

Глава 1

Документация Radiooooo-Qt

1.1 Классы:

- Configurator - класс для работы с настройками приложения
- Radiooooo - класс для работы с API и аудиофайлами

1.2 Заголовочные файлы:

- configurator.h - заголовочный файл класса Configurator
- radiooooo.h - заголовочный файл класса Radiooooo
- defaults.h - заголовочный файл со значениями по умолчанию

1.3 Сигналы и слоты

Ассинхронные взаимодействия (в основном с UI) происходят при помощи сигналов и слотов. Например, у класса Radiooooo есть слот updateConfig для получения сигнала из QML интерфейса.

Объявление слота в C++:

```
public slots:
    void updateConfig(QString param,
                      QString value,
                      bool enable);
```

В QML выполняется такая конструкция:

```
radio.updateConfig(param, value, enable);
```

Объект radio добавляется в контекс QML в main.cpp:

```
Radiooooo radio;
engine.rootContext()->setContextProperty("radio", &radio);
```

Глава 2

Список задач

Класс Configurator

Переписать взаимодействие с QML придерживаясь MVP

Член Radiooooo::forcePause ()

Переписать без вызова QML из C++ (например при помощи проверки флага в QML)

Глава 3

Иерархический список классов

3.1 Иерархия классов

Иерархия классов.

Глава 4

Алфавитный указатель классов

4.1 Классы

Классы с их кратким описанием.

Глава 5

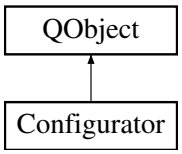
Классы

5.1 Класс Configurator

Используется для загрузки и сохранения настроек приложения.

#include <configurator.h>

Граф наследования:Configurator:



Открытые типы

- typedef QMap< QString, QList< QString > > ConfigStorage
  Тип данных для хранения настроек

Открытые члены

- Configurator (QObject *parent=nullptr)
  Configurator - дефолтный констуктор класса QObject.
- void initDirs ()
  Создаёт папки в домашней дирректории приложения
- void initConfig ()
  Заполняет настройки со значениями по умолчанию
- void saveConfig ()
  Сохраняет настройки в json файл
- void loadConfig ()
  Загружает настройки из json файла
- void updateConfig (QString param, QString value, bool enable)
  Обновляет настройки приложения
- ConfigStorage getConfig ()

Геттер для настроек приложения
- QString getConfigStr ()

  Геттер для настроек приложения
- QJsonDocument configToJson (ConfigStorage c)

  Конвертирует настройки в json документ
- QString getQuickCountries ()

  Возвращает список стран для быстрой настройки в виде строки (для QML)

Открытые атрибуты

- const QList< QString > moods = defMoods

  Все доступные "настрения".
- const QList< QString > allCountries = defCountries

  Все доступные страны
- const QMap< QString, QString > defaultConfig = defConfig

  Настройки по умолчанию
- const int minDecade = 1910

  Минимальный год
- const int maxDecade = 2020

  Максимальный год
- const QMap< QString, QString > quickCountries = quickSetupCountries

  Список стран для быстрой настройки
- const QString appDir = ".radiooooo-qt"

  Имя для домашней дирректории приложения
- const QString appDirPath = QDir::homePath() + "/" + appDir

  Пусть к домашней дирректории приложения
- const QString configFilePath = appDirPath + "/config.json"

  Путь к json файлу для настроек
- const QString audioDirPath = appDirPath + "/" + "audio-files"

  Путь к папке для хранения аудиофайлов

Закрытые данные

- ConfigStorage config

  Переменная для хранения настроек

5.1.1 Подробное описание

Используется для загрузки и сохранения настроек приложения.

Необходимо сделать Переписать взаимодействие с QML придерживаясь MVP

5.1.2 Конструктор(ы)

5.1.2.1 Configurator()

```
Configurator::Configurator (
            QObject * parent = nullptr )   [explicit]
```

Configurator - дефолтный констуктор класса QObject.

Аргументы

parent	- указатель на родительский QObject

5.1.3 Методы

5.1.3.1 configToJson()

QJsonDocument Configurator::configToJson (
Configurator::ConfigStorage c)

Конвертирует настройки в json документ

Аргументы

c	- список настроек

Возвращает

QJsonDocument - json документ с настройками

5.1.3.2 getConfig()

Configurator::ConfigStorage Configurator::getConfig ( )

Геттер для настроек приложения

Возвращает

настройки в виде ConfigStorage - key:value структуры с настройками приложения

5.1.3.3 getConfigStr()

QString Configurator::getConfigStr ( )

Геттер для настроек приложения

Возвращает

настройки в виде json строки

5.1.3.4 getQuickCountries()

QString Configurator::getQuickCountries ( )

Возвращает список стран для быстрой настройки в виде строки (для QML)

Возвращает

QString - json строка со списком стран

5.1.3.5 updateConfig()

void Configurator::updateConfig (
QString param,
QString value,
bool enable)

Обновляет настройки приложения

Аргументы

param	- имя параметра
value	- значение
enable	- флаг, если true, то добавить настройку, иначе удалить

Объявления и описания членов классов находятся в файлах:

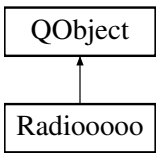
- configurator.h
- configurator.cpp

5.2 Класс Radiooooo

Реализует взаимодействие с https://radiooooo.app/ и логику аудиоплеера

#include <radiooooo.h>

Граф наследования:Radiooooo:



Открытые типы

- enum { IDLE , PLAYING , PAUSED , DOWNLOADING }
  Все состояния аудиоплеера

Открытые слоты

- QString loadConfig ()

  Слот для загрузки настроек
- void updateConfig (QString param, QString value, bool enable)

  Слот для обновления настроек
- void playTrigger (bool play)

  Изменяет состояние аудиоплеера
- void skipTrigger ()

  Пропускает трек
- void setVolume (int volume)

  Устанавливает громкость
- void stateChanged (QMediaPlayer::State playerState)

  Запускает следующий трек когда текущий заканчивается
- void durationChanged (qint64 newDuration)

  Обновляет длительность трека
- void updateProgress (qint64 pos)

  Считает прогресс трека и вызывает updateProgressBar.
- QString getQuickCountries ()

  Получение стран для быстрой настройки
- QList< QString > getMoods ()

  Получает список настроений
- int getMinDecade ()

  геттер для minDecade.
- int getMaxDeacde ()

  геттер для maxDecade.

Сигналы

- void updateProgressBar (double progress)

  Обновлет прогресс бар
- void updateStatusMsg (QString message)

  Обновляет сообщение со статусом
- void forcePause ()

  Обнолвет состояние кнопки "play/pause" в QML.

Открытые члены

- Radiooooo (QObject *parent=nullptr)

  Radiooooo - дефолтный констуктор класса QObject.

Открытые атрибуты

- int state = IDLE

  Текущее состояние аудиоплеера

Закрытые члены

- QJsonObject getSongInfo ()
  Получает данные через API Radiooooo.
- QJsonObject getCountries (QString decade)
  Получает доступные исокоды
- QString downloadFile (QString path, QString url)
  Скачивает файл по URL.
- bool saveFile (QString path, QByteArray data)
  Сохраняет файл
- void playNext ()
  Сохраняет файл и запускает его через QMediaPlayer.

Закрытые данные

- const QString baseAPI = "https://radiooooo.app"
  Корневой эндпоинт для API Radiooooo.
- const QString getCodesUrl = baseAPI + "/country/mood?decade="
  Эндпоинт для получения исокодов
- const QString getSongUrl = baseAPI + "/play"
  Эндпоинт для получения аудиофайла
- QString nowPlaying
  Строка, которая хранит название трека и название группы
- qint64 audioDuration = 0
  Продолжительность трека
- Configurator * cfg
  Указатель на Configurator.
- QMediaPlayer * mediaPlayer
  Указатель на QMediaPlayer (класс Qt для работы с медиафайлами)
- QNetworkAccessManager * netManager
  Указатель на QNetworkAccessManager (класс Qt для работы с сетью)

5.2.1 Подробное описание

Реализует взаимодействие с https://radiooooo.app/ и логику аудиоплеера

Заметки

Функции слоты класса Radiooooo доступны из QML

Взаимодействие с объектом класса Configurator происходит при помощи сигналов

5.2.2 Конструктор(ы)

5.2.2.1 Radiooooo()

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

Radiooooo - дефолтный констуктор класса QObject.

Аргументы

parent	- указатель на родительский QObject

5.2.3 Методы

5.2.3.1 downloadFile()

QString Radiooooo::downloadFile (
QString path,
QString url) [private]

Скачивает файл по URL.

Аргументы

path	- имя файла для сохранения
url	- url откуда скачать файл

Возвращает

QString - полный путь к файлу

5.2.3.2 durationChanged

void Radiooooo::durationChanged (
qint64 newDuration) [slot]

Обновляет длительность трека

Аргументы

newDuration	- новая длительность трека

5.2.3.3 forcePause

void Radiooooo::forcePause ( ) [signal]

Обнолвет состояние кнопки "play/pause" в QML.

Необходимо сделать Переписать без вызова QML из C++ (например при помощи проверки флага в QML)

5.2.3.4 getCountries()

QJsonObject Radiooooo::getCountries (
QString decade) [private]

Получает доступные исокоды

Аргументы

Десятилетие	(например 1970, 1980)

Возвращает

QJsonObject со списком кодов

Заметки

Не проверяется кратность

5.2.3.5 getMaxDeacde

int Radiooooo::getMaxDeacde ( ) [slot]

геттер для maxDecade.

Возвращает

maxDecade из Configurator

5.2.3.6 getMinDecade

int Radiooooo::getMinDecade ( ) [slot]

геттер для minDecade.

Возвращает

minDecade из Configurator

5.2.3.7 getMoods

QList< QString > Radiooooo::getMoods ( ) [slot]

Получает список настроений

Возвращает

QList строк

5.2.3.8 getQuickCountries

QString Radiooooo::getQuickCountries ( ) [slot]

Получение стран для быстрой настройки

Возвращает

QString - строка со списком

5.2.3.9 getSongInfo()

QJsonObject Radiooooo::getSongInfo ( ) [private]

Получает данные через API Radiooooo.

Возвращает

QJsonObject со всей информацией о треке

5.2.3.10 loadConfig

QString Radiooooo::loadConfig ( ) [slot]

Слот для загрузки настроек

Возвращает

QString с настройками

5.2.3.11 playTrigger

void Radiooooo::playTrigger (
bool play) [slot]

Изменяет состояние аудиоплеера

Аргументы

play	- запустить (true) или отсановить (false)

5.2.3.12 saveFile()

bool Radiooooo::saveFile (
QString path,
QByteArray data) [private]

Сохраняет файл

Аргументы

path	- путь куда сохранить
data	- QByteArray содержимое файла

Возвращает

true, если удалось сохранить

5.2.3.13 setVolume

void Radiooooo::setVolume (
int volume) [slot]

Устанавливает громкость

Аргументы

volume	- громкость (0-100)

5.2.3.14 stateChanged

void Radiooooo::stateChanged (
QMediaPlayer::State playerState) [slot]

Запускает следующий трек когда текущий заканчивается

Аргументы

playerState	- состояние плеера

5.2.3.15 updateConfig

void Radiooooo::updateConfig (
QString param,
QString value,
bool enable) [slot]

Слот для обновления настроек

Аргументы

param	- имя параметра
value	- значение
enable	- флаг, если true, то добавить настройку, иначе удалить

5.2.3.16 updateProgress

void Radiooooo::updateProgress (
qint64 pos) [slot]

Считает прогресс трека и вызывает updateProgressBar.

Аргументы

pos	- прошедшее время трека

5.2.3.17 updateProgressBar

void Radiooooo::updateProgressBar (
double progress) [signal]

Обновлет прогресс бар

Аргументы

progress	- процент завершения

5.2.3.18 updateStatusMsg

void Radiooooo::updateStatusMsg (
QString message) [signal]

Обновляет сообщение со статусом

Аргументы

message	- строка с сообщением

Объявления и описания членов классов находятся в файлах:

- radiooooo.h
- radiooooo.cpp

Предметный указатель