jpo

Generated by Doxygen 1.9.4

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

Chapter 2

Class Index

2.1 Class List

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

Chapter 3

File Index

3.1 File List

Here is a list of all documented files with brief descriptions:

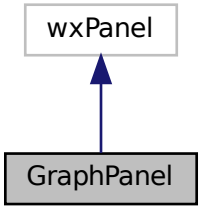
Chapter 4

Class Documentation

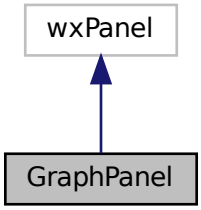
4.1 GraphPanel Class Reference

Custom wxPanel to draw a graph based on sensor data.

Inheritance diagram for GraphPanel:



Collaboration diagram for GraphPanel:



Public Member Functions

- GraphPanel (wxWindow *parent, const vector< nlohmann::json > &data)

  Constructs a GraphPanel.

4.1.1 Detailed Description

Custom wxPanel to draw a graph based on sensor data.

The GraphPanel class takes a vector of JSON objects representing sensor data. It draws a graph with grid lines, axes, data points, and labels for min, max, average, and trend. graph min and max values are based on min and max values that are marked and display with coresponding date

4.1.2 Constructor & Destructor Documentation

4.1.2.1 GraphPanel()

```
GraphPanel::GraphPanel (
            wxWindow * parent,
            const vector< nlohmann::json > & data )  [inline]
```

Constructs a GraphPanel.

Parameters

parent	Pointer to the parent wxWindow.
data	A vector of JSON objects with sensor data.

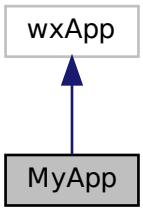
The documentation for this class was generated from the following file:

- main.cpp

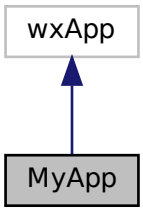
4.2 MyApp Class Reference

wxApp-derived class that initializes the application.

Inheritance diagram for MyApp:



Collaboration diagram for MyApp:



Public Member Functions

- virtual bool OnInit ()

  Application initialization.

4.2.1 Detailed Description

wxApp-derived class that initializes the application.

Sets locale, creates the main frame, and initializes station data.

4.2.2 Member Function Documentation

4.2.2.1 OnInit()

```
virtual bool MyApp::OnInit ( )  [inline], [virtual]
```

Application initialization.

Sets the locale to UTF-8, creates the main window, initializes the sensor database, and displays the main window.

Returns

true if initialization is successful.

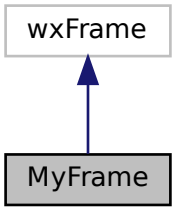
The documentation for this class was generated from the following file:

- main.cpp

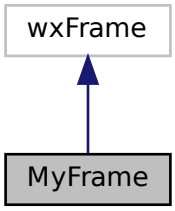
4.3 MyFrame Class Reference

Main window frame for the Professional App.

Inheritance diagram for MyFrame:



Collaboration diagram for MyFrame:



Public Member Functions

- **MyFrame** ()

  Constructs the main application frame.

4.3.1 Detailed Description

Main window frame for the Professional App.

Contains UI elements such as search box, buttons, and result list. Handles searching and displaying sensor station data.

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

- main.cpp

Chapter 5

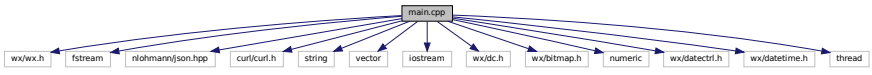
File Documentation

5.1 main.cpp File Reference

Main application file for the JPO project.

```
#include <wx/wx.h>
#include <fstream>
#include <nlohmann/json.hpp>
#include <curl/curl.h>
#include <string>
#include <vector>
#include <iostream>
#include <wx/dc.h>
#include <wx/bitmap.h>
#include <numeric>
#include <wx/datectrl.h>
#include <wx/datetime.h>
#include <thread>
```

Include dependency graph for main.cpp:



Classes

- class GraphPanel

  Custom wxPanel to draw a graph based on sensor data.
- class MyFrame

  Main window frame for the Professional App.
- class MyApp

  wxApp-derived class that initializes the application.

Functions

- size_t WriteCallback (void ∗contents, size_t size, size_t nmemb, string ∗output)

  Callback for libcurl to write received data.
- void fetchAndSaveData (const string &url, const string &filename)

  Fetches JSON data from a URL and saves it to a file.
- void init (wxWindow ∗parent)

  Initializes sensor station data.
- void fetchAndSaveSensorData (int stationID, int sensorID)

  Fetches sensor data for a specific sensor of a station.
- void updateData (int stationID)

  Updates the sensor data for a station.
- **wxIMPLEMENT_APP** (MyApp)

5.1.1 Detailed Description

Main application file for the JPO project.

This application uses wxWidgets to create a GUI for fetching and displaying sensor data. It uses libcurl for HTTP requests and nlohmann::json for JSON parsing.

5.1.2 Function Documentation

5.1.2.1 fetchAndSaveData()

```
void fetchAndSaveData (
            const string & url,
            const string & filename )
```

Fetches JSON data from a URL and saves it to a file.

Uses libcurl to perform an HTTP GET request, parses the response as JSON, and writes the formatted JSON to the specified file.

Parameters

url	The URL to fetch data from.
filename	The file to which the JSON data will be written.

5.1.2.2 fetchAndSaveSensorData()

```
void fetchAndSaveSensorData (
            int stationID,
            int sensorID )
```

Fetches sensor data for a specific sensor of a station.

Retrieves sensor data via HTTP request and updates the local JSON file for the station.

Parameters

stationID	Identifier of the station.
sensorID	Identifier of the sensor.

5.1.2.3 init()

```
void init (
            wxWindow * parent )
```

Initializes sensor station data.

Checks if a local JSON file exists and is non-empty. If not, or if the user requests an update, it downloads the station database.

Parameters

parent	Pointer to the parent wxWindow for dialog messages.

5.1.2.4 updateData()

```
void updateData (
            int stationID )
```

Updates the sensor data for a station.

Constructs the proper URL using the station ID and calls fetchAndSaveData().

Parameters

stationID	Identifier of the station.

5.1.2.5 WriteCallback()

```
size_t WriteCallback (
            void * contents,
            size_t size,
            size_t nmemb,
            string * output )
```

Callback for libcurl to write received data.

This function is called by libcurl as soon as data is received and appends it to a string.

Parameters

contents	Pointer to the data received.
size	Size of a single element.
nmemb	Number of elements.
output	Pointer to the string to which data is appended.

Returns

The total number of bytes processed.

Index