

AIQWA

1.0

Generated by Doxygen 1.9.1



<b>1 Class Index</b>	<b>1</b>
1.1 Class List	1
<b>2 File Index</b>	<b>3</b>
2.1 File List	3
<b>3 Class Documentation</b>	<b>5</b>
3.1 Color Struct Reference	5
3.1.1 Member Data Documentation	5
3.1.1.1 blue	5
3.1.1.2 green	5
3.1.1.3 indice	5
3.1.1.4 nom	6
3.1.1.5 red	6
3.2 CurveChartParams Struct Reference	6
3.2.1 Member Data Documentation	6
3.2.1.1 colorIndex	6
3.2.1.2 dataSets	6
3.2.1.3 description	7
3.2.1.4 globalMax	7
3.2.1.5 globalMin	7
3.2.1.6 legendWidth	7
3.2.1.7 marginBottom	7
3.2.1.8 marginLeft	7
3.2.1.9 nbMeasures	7
3.2.1.10 title	7
3.2.1.11 titleHeight	8
3.3 DataElement Struct Reference	8
3.3.1 Member Data Documentation	8
3.3.1.1 abscisse	8
3.3.1.2 label	8
3.3.1.3 value	8
3.4 dataSet Struct Reference	9
3.4.1 Member Data Documentation	9
3.4.1.1 colorIndex	9
3.4.1.2 dataElements	9
3.4.1.3 legend	9
3.5 memory Struct Reference	9
3.5.1 Member Data Documentation	10
3.5.1.1 response	10
3.5.1.2 size	10
3.6 myCurl Class Reference	10
3.6.1 Detailed Description	11

3.6.2 Constructor & Destructor Documentation	11
3.6.2.1 myCurl()	11
3.6.3 Member Function Documentation	11
3.6.3.1 exec()	11
3.6.3.2 getData()	12
3.6.3.3 mem_cb()	12
3.6.4 Member Data Documentation	12
3.6.4.1 hnd	12
3.7 myGraphics Class Reference	13
3.7.1 Detailed Description	14
3.7.2 Constructor & Destructor Documentation	14
3.7.2.1 myGraphics()	14
3.7.2.2 ~myGraphics()	15
3.7.3 Member Function Documentation	15
3.7.3.1 appendDataVector()	15
3.7.3.2 calcRatioPourcent()	15
3.7.3.3 createColor()	16
3.7.3.4 curveChart()	16
3.7.3.5 curveChartAddCurves()	16
3.7.3.6 curveChartInit()	17
3.7.3.7 curveChartSetLegend()	17
3.7.3.8 getDataVectorMaxValue()	17
3.7.3.9 getDataVectorMinValue()	18
3.7.3.10 initFonts()	18
3.7.3.11 initPalette()	18
3.7.3.12 pieChart()	18
3.7.3.13 pieChartDraw()	19
3.7.3.14 pieChartInit()	19
3.7.3.15 setVector()	19
3.7.4 Member Data Documentation	20
3.7.4.1 ficOut	20
3.7.4.2 fonts	20
3.7.4.3 gdColors	20
3.7.4.4 im	20
3.7.4.5 imageSize	20
3.7.4.6 myPalette	20
3.7.4.7 titre	21
3.8 myOptions Class Reference	21
3.8.1 Detailed Description	22
3.8.2 Constructor & Destructor Documentation	22
3.8.2.1 myOptions()	23
3.8.3 Member Function Documentation	23

3.8.3.1	<a href="#">getApiKeyHeader()</a>	23
3.8.3.2	<a href="#">getDataPathName()</a>	23
3.8.3.3	<a href="#">getFilesOwner()</a>	23
3.8.3.4	<a href="#">getFullCurveChartFileName()</a>	24
3.8.3.5	<a href="#">getFullDataFileName()</a>	24
3.8.3.6	<a href="#">getFullHistoChartFileName()</a>	24
3.8.3.7	<a href="#">getFullLogFileName()</a>	24
3.8.3.8	<a href="#">getFullPieChartFileName()</a>	25
3.8.3.9	<a href="#">getFullUrl()</a>	25
3.8.3.10	<a href="#">isDisplay()</a>	25
3.8.3.11	<a href="#">readFromFile()</a>	25
3.8.4	<a href="#">Member Data Documentation</a>	26
3.8.4.1	<a href="#">options</a>	26
3.9	<a href="#">myParsing Class Reference</a>	26
3.9.1	<a href="#">Detailed Description</a>	27
3.9.2	<a href="#">Constructor &amp; Destructor Documentation</a>	27
3.9.2.1	<a href="#">myParsing()</a>	27
3.9.3	<a href="#">Member Function Documentation</a>	27
3.9.3.1	<a href="#">appendFileToDatas()</a>	27
3.9.3.2	<a href="#">appendToDatas()</a>	27
3.9.3.3	<a href="#">fromChar()</a>	28
3.9.3.4	<a href="#">fromDatasFile()</a>	28
3.9.3.5	<a href="#">fromFile()</a>	28
3.9.3.6	<a href="#">getMyJson()</a>	29
3.9.3.7	<a href="#">setMyJson()</a>	29
3.9.3.8	<a href="#">toFile()</a>	29
3.9.3.9	<a href="#">toString()</a>	30
3.9.4	<a href="#">Member Data Documentation</a>	30
3.9.4.1	<a href="#">myJson</a>	30
3.10	<a href="#">myRegex Class Reference</a>	30
3.10.1	<a href="#">Detailed Description</a>	31
3.10.2	<a href="#">Constructor &amp; Destructor Documentation</a>	31
3.10.2.1	<a href="#">myRegex()</a>	31
3.10.3	<a href="#">Member Function Documentation</a>	31
3.10.3.1	<a href="#">ambeeDataGetDateTime()</a>	31
3.10.3.2	<a href="#">toFormatDDMMYY()</a>	32
3.10.3.3	<a href="#">toFormatDDMMYYHHMM()</a>	32
3.10.3.4	<a href="#">toFormatHHMM()</a>	32
3.11	<a href="#">updatedAt Struct Reference</a>	33
3.11.1	<a href="#">Member Data Documentation</a>	33
3.11.1.1	<a href="#">day</a>	33
3.11.1.2	<a href="#">hour</a>	33

3.11.1.3 minute	33
3.11.1.4 month	33
3.11.1.5 second	34
3.11.1.6 year	34
<b>4 File Documentation</b>	<b>35</b>
4.1 /home/sylvain/Documents/Travail/ajc-akkodis/formation/LinuxEmbarque/Projet/devel/AIrQuality↔ WAtch/main.cpp File Reference	35
4.1.1 Function Documentation	35
4.1.1.1 downloadDatas()	35
4.1.1.2 main()	36
4.2 /home/sylvain/Documents/Travail/ajc-akkodis/formation/LinuxEmbarque/Projet/devel/AIrQuality↔ WAtch/mycurl.cpp File Reference	36
4.3 /home/sylvain/Documents/Travail/ajc-akkodis/formation/LinuxEmbarque/Projet/devel/AIrQuality↔ WAtch/mycurl.h File Reference	36
4.4 /home/sylvain/Documents/Travail/ajc-akkodis/formation/LinuxEmbarque/Projet/devel/AIrQuality↔ WAtch/mygraphics.cpp File Reference	37
4.5 /home/sylvain/Documents/Travail/ajc-akkodis/formation/LinuxEmbarque/Projet/devel/AIrQuality↔ WAtch/mygraphics.h File Reference	38
4.6 /home/sylvain/Documents/Travail/ajc-akkodis/formation/LinuxEmbarque/Projet/devel/AIrQuality↔ WAtch/myoptions.cpp File Reference	39
4.7 /home/sylvain/Documents/Travail/ajc-akkodis/formation/LinuxEmbarque/Projet/devel/AIrQuality↔ WAtch/myoptions.h File Reference	40
4.7.1 Typedef Documentation	41
4.7.1.1 json	41
4.8 /home/sylvain/Documents/Travail/ajc-akkodis/formation/LinuxEmbarque/Projet/devel/AIrQuality↔ WAtch/myparsing.cpp File Reference	41
4.9 /home/sylvain/Documents/Travail/ajc-akkodis/formation/LinuxEmbarque/Projet/devel/AIrQuality↔ WAtch/myparsing.h File Reference	41
4.9.1 Typedef Documentation	42
4.9.1.1 json	43
4.10 /home/sylvain/Documents/Travail/ajc-akkodis/formation/LinuxEmbarque/Projet/devel/AIrQuality↔ WAtch/myregex.cpp File Reference	43
4.11 /home/sylvain/Documents/Travail/ajc-akkodis/formation/LinuxEmbarque/Projet/devel/AIrQuality↔ WAtch/myregex.h File Reference	43
<b>Index</b>	<b>45</b>

# Chapter 1

## Class Index

### 1.1 Class List

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

<a href="#">Color</a>	5
<a href="#">CurveChartParams</a>	6
<a href="#">DataElement</a>	8
<a href="#">dataSet</a>	9
<a href="#">memory</a>	9
<a href="#">myCurl</a>	
The <a href="#">myCurl</a> class	10
<a href="#">myGraphics</a>	
The <a href="#">myGraphics</a> class	13
<a href="#">myOptions</a>	
The <a href="#">myOptions</a> class	21
<a href="#">myParsing</a>	
The <a href="#">myParsing</a> class	26
<a href="#">myRegex</a>	
The <a href="#">myRegex</a> class	30
<a href="#">updatedAt</a>	33





## Chapter 2

# File Index

### 2.1 File List

Here is a list of all files with brief descriptions:

[/home/sylvain/Documents/Travail/ajc-akkodis/formation/LinuxEmbarque/Projet/devel/AlrQualityWAtch/main.cpp](#)  
35

[/home/sylvain/Documents/Travail/ajc-akkodis/formation/LinuxEmbarque/Projet/devel/AlrQualityWAtch/mycurl.cpp](#)  
36

[/home/sylvain/Documents/Travail/ajc-akkodis/formation/LinuxEmbarque/Projet/devel/AlrQualityWAtch/mycurl.h](#)  
36

[/home/sylvain/Documents/Travail/ajc-akkodis/formation/LinuxEmbarque/Projet/devel/AlrQualityWAtch/mygraphics.cpp](#)  
37

[/home/sylvain/Documents/Travail/ajc-akkodis/formation/LinuxEmbarque/Projet/devel/AlrQualityWAtch/mygraphics.h](#)  
38

[/home/sylvain/Documents/Travail/ajc-akkodis/formation/LinuxEmbarque/Projet/devel/AlrQualityWAtch/myoptions.cpp](#)  
39

[/home/sylvain/Documents/Travail/ajc-akkodis/formation/LinuxEmbarque/Projet/devel/AlrQualityWAtch/myoptions.h](#)  
40

[/home/sylvain/Documents/Travail/ajc-akkodis/formation/LinuxEmbarque/Projet/devel/AlrQualityWAtch/myparsing.cpp](#)  
41

[/home/sylvain/Documents/Travail/ajc-akkodis/formation/LinuxEmbarque/Projet/devel/AlrQualityWAtch/myparsing.h](#)  
41

[/home/sylvain/Documents/Travail/ajc-akkodis/formation/LinuxEmbarque/Projet/devel/AlrQualityWAtch/myregex.cpp](#)  
43

[/home/sylvain/Documents/Travail/ajc-akkodis/formation/LinuxEmbarque/Projet/devel/AlrQualityWAtch/myregex.h](#)  
43



## Chapter 3

# Class Documentation

### 3.1 Color Struct Reference

```
#include <mygraphics.h>
```

#### Public Attributes

- string [nom](#)
- int [indice](#)
- int [red](#)
- int [green](#)
- int [blue](#)

#### 3.1.1 Member Data Documentation

##### 3.1.1.1 blue

```
int Color::blue
```

##### 3.1.1.2 green

```
int Color::green
```

##### 3.1.1.3 indice

```
int Color::indice
```

#### 3.1.1.4 nom

```
string Color::nom
```

#### 3.1.1.5 red

```
int Color::red
```

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

- /home/sylvain/Documents/Travail/ajc-akkodis/formation/LinuxEmbarque/Projet/devel/AlrQualityWAtch/[mygraphics.h](#)

## 3.2 CurveChartParams Struct Reference

```
#include <mygraphics.h>
```

### Public Attributes

- string [title](#)
- string [description](#)
- vector< [dataSet](#) > \* [dataSets](#)
- int [nbMeasures](#)
- int [colorIndex](#)
- int [titleHeight](#)
- int [marginLeft](#)
- int [marginBottom](#)
- int [legendWidth](#)
- float [globalMax](#)
- float [globalMin](#)

### 3.2.1 Member Data Documentation

#### 3.2.1.1 colorIndex

```
int CurveChartParams::colorIndex
```

#### 3.2.1.2 dataSets

```
vector<dataSet>* CurveChartParams::dataSets
```

### 3.2.1.3 description

```
string CurveChartParams::description
```

### 3.2.1.4 globalMax

```
float CurveChartParams::globalMax
```

### 3.2.1.5 globalMin

```
float CurveChartParams::globalMin
```

### 3.2.1.6 legendWidth

```
int CurveChartParams::legendWidth
```

### 3.2.1.7 marginBottom

```
int CurveChartParams::marginBottom
```

### 3.2.1.8 marginLeft

```
int CurveChartParams::marginLeft
```

### 3.2.1.9 nbMeasures

```
int CurveChartParams::nbMeasures
```

### 3.2.1.10 title

```
string CurveChartParams::title
```

### 3.2.1.11 titleHeight

```
int CurveChartParams::titleHeight
```

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

- </home/sylvain/Documents/Travail/ajc-akkodis/formation/LinuxEmbarque/Projet/devel/AlrQualityWAtch/mygraphics.h>

## 3.3 DataElement Struct Reference

```
#include <mygraphics.h>
```

### Public Attributes

- string [label](#)
- float [abscisse](#)
- float [value](#)

### 3.3.1 Member Data Documentation

#### 3.3.1.1 abscisse

```
float DataElement::abscisse
```

#### 3.3.1.2 label

```
string DataElement::label
```

#### 3.3.1.3 value

```
float DataElement::value
```

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

- </home/sylvain/Documents/Travail/ajc-akkodis/formation/LinuxEmbarque/Projet/devel/AlrQualityWAtch/mygraphics.h>

## 3.4 dataSet Struct Reference

```
#include <mygraphics.h>
```

### Public Attributes

- vector< [DataElement](#) > \* [dataElements](#)
- int [colorIndex](#)
- string [legend](#)

### 3.4.1 Member Data Documentation

#### 3.4.1.1 colorIndex

```
int dataSet::colorIndex
```

#### 3.4.1.2 dataElements

```
vector<DataElement>* dataSet::dataElements
```

#### 3.4.1.3 legend

```
string dataSet::legend
```

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

- /home/sylvain/Documents/Travail/ajc-akkodis/formation/LinuxEmbarque/Projet/devel/AirQualityWatch/[mygraphics.h](#)

## 3.5 memory Struct Reference

```
#include <mycurl.h>
```

### Public Attributes

- char \* [response](#)
- size\_t [size](#)

### 3.5.1 Member Data Documentation

#### 3.5.1.1 response

```
char* memory::response
```

#### 3.5.1.2 size

```
size_t memory::size
```

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

- /home/sylvain/Documents/Travail/ajc-akkodis/formation/LinuxEmbarque/Projet/devel/AIQualityWatch/[mycurl.h](#)

## 3.6 myCurl Class Reference

The [myCurl](#) class.

```
#include <mycurl.h>
```

### Public Member Functions

- [myCurl](#) ()  
*myCurl*
- void [getData](#) (char \*\*response)  
*getData*
- CURLcode [exec](#) ([myOptions](#) \*options, char \*\*data)  
*exec*

### Static Private Member Functions

- static size\_t [mem\\_cb](#) (void \*contents, size\_t size, size\_t nmemb, void \*userp)  
*mem\_cb*

### Private Attributes

- CURL \* [hnd](#)



### 3.6.1 Detailed Description

The [myCurl](#) class.

Manage **API access**

includes *curl/curl.h* library

includes [myoptions.h](#) library

Requires *-lcurl* flag

### 3.6.2 Constructor & Destructor Documentation

#### 3.6.2.1 myCurl()

```
myCurl::myCurl ( )
```

[myCurl](#)

constructor

### 3.6.3 Member Function Documentation

#### 3.6.3.1 exec()

```
CURLcode myCurl::exec (
    myOptions * options,
    char ** data )
```

exec

Parameters

<i>myOption*</i>	options
------------------	---------

Returns

CURLcode

Prepare and execute curl request using parameters in options

Downloaded data can be retrieved by a call to the method [getData\(\)](#)

### 3.6.3.2 getData()

```
void myCurl::getData (
    char ** response )
```

getData

Allows to recover downloaded data

Parameters

<i>char**</i>	response
---------------	----------

### 3.6.3.3 mem\_cb()

```
size_t myCurl::mem_cb (
    void * contents,
    size_t size,
    size_t nmemb,
    void * userp ) [static], [private]
```

mem\_cb

Call back method required by *curl\_easy\_perform()* method to store read datas in buffer

Parameters

<i>void*</i>	contents
<i>size_t</i>	size
<i>size_t</i>	nmemb
<i>void*</i>	userp

Returns

static size\_t

## 3.6.4 Member Data Documentation

### 3.6.4.1 hnd

```
CURL* myCurl::hnd [private]
```

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

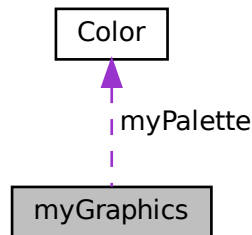
- /home/sylvain/Documents/Travail/ajc-akkodis/formation/LinuxEmbarque/Projet/devel/AlrQualityWatch/[mycurl.h](#)
- /home/sylvain/Documents/Travail/ajc-akkodis/formation/LinuxEmbarque/Projet/devel/AlrQualityWatch/[mycurl.cpp](#)

## 3.7 myGraphics Class Reference

The [myGraphics](#) class.

```
#include <mygraphics.h>
```

Collaboration diagram for myGraphics:



### Public Member Functions

- [myGraphics](#) ()  
*myGraphics*
- [~myGraphics](#) ()
- void [pieChart](#) ([json](#) datas, [myOptions](#) \*options)  
*pieChart*
- void [curveChart](#) ([json](#) datas, [myOptions](#) \*options)  
*curveChart* |param json datas

### Private Member Functions

- void [initPalette](#) ()  
*initPalette*
- void [initFonts](#) ()  
*initFonts*
- [Color](#) \* [createColor](#) (string nom, int indice, int red, int green, int blue)  
*createColor*
- float [getDataVectorMaxValue](#) (vector< [DataElement](#) > \*v)  
*getDataVectorMaxValue*
- float [getDataVectorMinValue](#) (vector< [DataElement](#) > \*v)  
*getDataVectorMinValue*
- vector< [DataElement](#) > \* [appendDataVector](#) (vector< [DataElement](#) > \*v, string label="", float abscisse=0.0, float value=0.0)  
*appendDataVector*
- void [setVector](#) (vector< [DataElement](#) > \*v, [json](#) datas, string dataSet)  
*setVector*
- void [curveChartInit](#) ([CurveChartParams](#) params)

- curveChartInit*
- void [pieChartInit](#) ([CurveChartParams](#) params)
- pieChartInit*
- void [pieChartDraw](#) ([CurveChartParams](#) params)
- pieChartDraw*
- double [calcRatioPourcent](#) ([CurveChartParams](#) params)
- calcRatioPourcent*
- void [curveChartAddCurves](#) ([CurveChartParams](#) params)
- curveChartAddCurves*
- void [curveChartSetLegend](#) ([CurveChartParams](#) params)
- curveChartSetLegend*

## Private Attributes

- string [titre](#)
- char [ficOut](#) [255]
- int [imageSize](#) = 1024
- [Color](#) \* [myPalette](#) [12]
- int [gdColors](#) [12]
- [gdFontPtr](#) [fonts](#) [5]
- [gdImagePtr](#) [im](#)

### 3.7.1 Detailed Description

The [myGraphics](#) class.

```
#include "myoptions.h"
```

```
#include "myparsing.h"
```

```
#include "myregex.h"
```

```
#include "gd.h"
```

requires -lgd flag

Provides methods for generating, displaying and saving pie charts and curve charts

### 3.7.2 Constructor & Destructor Documentation

#### 3.7.2.1 [myGraphics](#)()

```
myGraphics::myGraphics ( )
```

[myGraphics](#)

Constructor

Initialize image, palette and fonts

### 3.7.2.2 ~myGraphics()

```
myGraphics::~myGraphics ( )
```

## 3.7.3 Member Function Documentation

### 3.7.3.1 appendDataVector()

```
vector< DataElement > * myGraphics::appendDataVector (
    vector< DataElement > * v,
    string label = "",
    float abscisse = 0.0,
    float value = 0.0 ) [private]
```

appendDataVector

Parameters

<i>vector&lt;DataElement&gt;</i>	*v
<i>string</i>	label=""
<i>float</i>	abscisse=0.↵ 0
<i>float</i>	value=0.0

Returns

vector<DataElement>\*

Create new [DataElement](#) and appends it to v

### 3.7.3.2 calcRatioPourcent()

```
double myGraphics::calcRatioPourcent (
    CurveChartParams params ) [private]
```

calcRatioPourcent

Parameters

<a href="#">CurveChartParams</a>	params
----------------------------------	--------

Returns

double

Compute S sum of the values

returns  $100/(S+1)$

Required to calculate the angles of the pie chart

### 3.7.3.3 createColor()

```
Color * myGraphics::createColor (
    string nom,
    int indice,
    int red,
    int green,
    int blue ) [private]
```

createColor

#### Parameters

<i>nom</i>	
<i>indice</i>	
<i>red</i>	
<i>green</i>	
<i>blue</i>	

#### Returns

struct [Color](#) object. See [gdImageColorAllocate](#) documentation.

### 3.7.3.4 curveChart()

```
void myGraphics::curveChart (
    json datas,
    myOptions * options )
```

curveChart \param json datas

#### Parameters

<a href="#">myOptions</a>	*options
---------------------------	----------

generate, displays and save curve chart

### 3.7.3.5 curveChartAddCurves()

```
void myGraphics::curveChartAddCurves (
    CurveChartParams params ) [private]
```

curveChartAddCurves

## Parameters

<a href="#">CurveChartParams</a>	params
----------------------------------	--------

Draw lines

**3.7.3.6 curveChartInit()**

```
void myGraphics::curveChartInit (
    CurveChartParams params ) [private]
```

curveChartInit

## Parameters

<a href="#">CurveChartParams</a>	params
----------------------------------	--------

Draws Frame, axes, title and description

**3.7.3.7 curveChartSetLegend()**

```
void myGraphics::curveChartSetLegend (
    CurveChartParams params ) [private]
```

curveChartSetLegend

## Parameters

<a href="#">CurveChartParams</a>	params
----------------------------------	--------

Displays colored labels in legend area

**3.7.3.8 getDataVectorMaxValue()**

```
float myGraphics::getDataVectorMaxValue (
    vector< DataElement > * v ) [private]
```

getDataVectorMaxValue

## Parameters

<code>vector&lt;<a href="#">DataElement</a>&gt;</code>	*v
--	----

## Returns

float

required to set curve chart ordinate scale

### 3.7.3.9 getDataVectorMinValue()

```
float myGraphics::getDataVectorMinValue (
    vector< DataElement > * v ) [private]
```

getDataVectorMinValue

#### Parameters

<i>vector&lt;DataElement&gt;</i>	*v
----------------------------------	----

#### Returns

float

required to set curve chart ordinate scale

### 3.7.3.10 initFonts()

```
void myGraphics::initFonts ( ) [private]
```

initFonts

Initialize fonts

based on \_<gdfontt.h> <gdfonts.h> <gdfontmb.h> <gdfontl.h> <gdfontg.h>

### 3.7.3.11 initPalette()

```
void myGraphics::initPalette ( ) [private]
```

initPalette

creates 12 colors palette

Each color works in pair

eg: 2 -> green and 3 -> darkgreen

0 -> black

1 -> white

### 3.7.3.12 pieChart()

```
void myGraphics::pieChart (
    json datas,
    myOptions * options )
```

pieChart



## Parameters

<i>json</i>	<i>datas</i>
<i>myOptions</i>	<i>*options</i>

generate, displays and save pie chart

**3.7.3.13 pieChartDraw()**

```
void myGraphics::pieChartDraw (
    CurveChartParams params ) [private]
```

pieChartDraw

## Parameters

<i>CurveChartParams</i>	<i>params</i>
-------------------------	---------------

Draws pie chart

**3.7.3.14 pieChartInit()**

```
void myGraphics::pieChartInit (
    CurveChartParams params ) [private]
```

pieChartInit

## Parameters

<i>CurveChartParams</i>	<i>params</i>
-------------------------	---------------

Draws Frame, axes, title and description

**3.7.3.15 setVector()**

```
void myGraphics::setVector (
    vector< DataElement > * v,
    json datas,
    string dataSet ) [private]
```

setVector

## Parameters

<i>vector&lt;DataElement&gt;*</i>	<i>v</i>
<i>json</i>	<i>datas</i>
<i>string</i>	<i>dataSet</i>

Creates a [DataElement](#) for each measure stored in json datas

parsing datas using dataset : `datas[i][\"stations\"][0][dataSet];`

### 3.7.4 Member Data Documentation

#### 3.7.4.1 ficOut

```
char myGraphics::ficOut[255] [private]
```

#### 3.7.4.2 fonts

```
gdFontPtr myGraphics::fonts[5] [private]
```

#### 3.7.4.3 gdColors

```
int myGraphics::gdColors[12] [private]
```

#### 3.7.4.4 im

```
gdImagePtr myGraphics::im [private]
```

#### 3.7.4.5 imageSize

```
int myGraphics::imageSize = 1024 [private]
```

#### 3.7.4.6 myPalette

```
Color* myGraphics::myPalette[12] [private]
```

### 3.7.4.7 titre

```
string myGraphics::titre [private]
```

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

- /home/sylvain/Documents/Travail/ajc-akkodis/formation/LinuxEmbarque/Projet/devel/AlrQualityWatch/[mygraphics.h](#)
- /home/sylvain/Documents/Travail/ajc-akkodis/formation/LinuxEmbarque/Projet/devel/AlrQualityWatch/[mygraphics.cpp](#)

## 3.8 myOptions Class Reference

The [myOptions](#) class.

```
#include <myoptions.h>
```

### Public Member Functions

- [myOptions](#) ()  
*myOptions*
- void [readFromFile](#) (string fileName)  
*readFromFile*
- string [getFullUrl](#) ()  
*getFullUrl*
- string [getApiKeyHeader](#) ()  
*getApiKeyHeader*
- string [getFilesOwner](#) ()  
*getFilesOwner*
- string [getFullLogFileName](#) ()  
*getFullLogFileName*
- string [getFullDataFileName](#) ()  
*getFullDataFileName*
- string [getDataPathName](#) ()  
*getDataPathName();*
- string [getFullCurveChartFileName](#) ()  
*getFullCurveChartFileName*
- string [getFullHistoChartFileName](#) ()  
*getFullHistoChartFileName*
- string [getFullPieChartFileName](#) ()  
*getFullPieChartFileName*
- bool [isDisplay](#) ()  
*isDisplay*

### Private Attributes

- [json options](#)  
*options*

### 3.8.1 Detailed Description

The [myOptions](#) class.

Manage acces to AIQWA.conf file

Allows to modify API parameters and local files names and paths

without the need to recompile

File Format :

```
{  
  "curlopts": {  
    "url": "https://api.ambeedata.com/latest/by-lat-lng",  
    "urlParams": {  
      "lat": "43.560537",  
      "lng": "1.404690"  
    },  
    "api-key": "x-api-key:b83fcfd7137ff81d96b92a34d3488506b7d3976bda58077cab133e94efd0a240"  
  },  
  "display":true,  
  "filesOwner": "owner",  
  "logPath": "logs",  
  "logFileName": "AIQWA.log",  
  "dataPath": "datas",  
  "dataFileName": "AirQualityWatch",  
  "chartPath": "charts",  
  "curveChartFileName": "curve.png",  
  "histoChartFileName": "histo.png",  
  "pieChartFileName": "pie.png"  
}
```

### 3.8.2 Constructor & Destructor Documentation

### 3.8.2.1 myOptions()

```
myOptions::myOptions ( )
```

[myOptions](#)

Constructor

## 3.8.3 Member Function Documentation

### 3.8.3.1 getApiKeyHeader()

```
string myOptions::getApiKeyHeader ( )
```

getApiKeyHeader

Returns

string

### 3.8.3.2 getDataPathName()

```
string myOptions::getDataPathName ( )
```

[getDataPathName\(\)](#);

Returns

string

return datas path

### 3.8.3.3 getFilesOwner()

```
string myOptions::getFilesOwner ( )
```

getFilesOwner

Returns

string

#### 3.8.3.4 getFullCurveChartFileName()

```
string myOptions::getFullCurveChartFileName ( )
```

getFullCurveChartFileName

##### Returns

string

return path/fileName

#### 3.8.3.5 getFullDataFileName()

```
string myOptions::getFullDataFileName ( )
```

getFullDataFileName

##### Returns

string

return datas path/fileName

#### 3.8.3.6 getFullHistoChartFileName()

```
string myOptions::getFullHistoChartFileName ( )
```

getFullHistoChartFileName

##### Returns

string

return path/fileName

#### 3.8.3.7 getFullLogFileName()

```
string myOptions::getFullLogFileName ( )
```

getFullLogFileName

##### Returns

string

### 3.8.3.8 getFullPieChartFileName()

```
string myOptions::getFullPieChartFileName ( )
```

getFullPieChartFileName

#### Returns

string

return path/fileName

### 3.8.3.9 getFullUrl()

```
string myOptions::getFullUrl ( )
```

getFullUrl

#### Returns

string

Returns full API Url, including GET params

### 3.8.3.10 isDisplay()

```
bool myOptions::isDisplay ( )
```

isDisplay

#### Returns

bool

Set false on embedded system

Set true on local system to allow system call to *display*

### 3.8.3.11 readFromFile()

```
void myOptions::readFromFile (
    string fileName )
```

readFromFile

#### Parameters

<i>string</i>	fileName
---------------	----------

Read options from file in json format

### 3.8.4 Member Data Documentation

#### 3.8.4.1 options

```
json myOptions::options [private]
```

options

Stores json options

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

- /home/sylvain/Documents/Travail/ajc-akkodis/formation/LinuxEmbarque/Projet/devel/AlrQualityWAtch/[myoptions.h](#)
- /home/sylvain/Documents/Travail/ajc-akkodis/formation/LinuxEmbarque/Projet/devel/AlrQualityWAtch/[myoptions.cpp](#)

## 3.9 myParsing Class Reference

The [myParsing](#) class.

```
#include <myparsing.h>
```

### Public Member Functions

- [myParsing](#) ()  
*myParsing*
- const [json](#) & [getMyJson](#) () const  
*getMyJson*
- void [setMyJson](#) (const [json](#) &newMyJson)  
*setMyJson*
- [myParsing](#) \* [fromChar](#) (string str)  
*fromChar*
- string [toString](#) ()  
*toString*
- [myParsing](#) \* [fromFile](#) (string fileName)  
*fromFile*
- int [toFile](#) ([myOptions](#) \*options)  
*toFile*
- void [appendToDats](#) ([myOptions](#) \*options)  
*appendToDats*
- void [appendFileToDats](#) (string fileName)  
*appendFileToDats*
- const [json](#) [fromDatsFile](#) ([myOptions](#) \*options)  
*fromDatsFile*



## Private Attributes

- [json myJson](#)

### 3.9.1 Detailed Description

The [myParsing](#) class.

shortcut methods based on <nlohmann/json.hpp> library

### 3.9.2 Constructor & Destructor Documentation

#### 3.9.2.1 myParsing()

```
myParsing::myParsing ( )
```

[myParsing](#)

Constructor

### 3.9.3 Member Function Documentation

#### 3.9.3.1 appendFileToDatas()

```
void myParsing::appendFileToDatas (
    string fileName )
```

appendFileToDatas

Parameters

<i>string</i>	<i>fileName</i>
---------------	-----------------

Append current latest downloaded data to file *fileName*

#### 3.9.3.2 appendToDatas()

```
void myParsing::appendToDatas (
    myOptions * options )
```

appendToDatas

## Parameters

<i>myOptions*</i>	options
-------------------	---------

Append current latest downloaded data to file *options->getFullDataFileName().json*

**3.9.3.3 fromChar()**

```
myParsing * myParsing::fromChar (
    string str )
```

fromChar

## Parameters

<i>char*</i>	str
--------------	-----

## Returns

myParsing\*

Generate json from char \*data read from API returned by curl

**3.9.3.4 fromDatasFile()**

```
const json myParsing::fromDatasFile (
    myOptions * options )
```

fromDatasFile

## Parameters

<i>myOptions</i>	*options
------------------	----------

## Returns

const json

returns json content of *\_options->getFullDataFileName().json*"

the file is json array containing previously stored datasets

**3.9.3.5 fromFile()**

```
myParsing * myParsing::fromFile (
    string fileName )
```

fromFile

**Parameters**

<i>string</i>	fileName
---------------	----------

**Returns**

myParsing\*

Read file in json format, parse it and stores data in myJson

**3.9.3.6 getMyJson()**

```
const json & myParsing::getMyJson ( ) const
```

getMyJson

**Returns**

const json &

**3.9.3.7 setMyJson()**

```
void myParsing::setMyJson (
    const json & newMyJson )
```

setMyJson

**Parameters**

<i>const</i>	json &newMyJson
--------------	-----------------

**3.9.3.8 toFile()**

```
int myParsing::toFile (
    myOptions * options )
```

toFile

**Parameters**

<i>myOptions*</i>	options
-------------------	---------

**Returns**

int

stores myJson to fileName returned by options->getFullDataFileName

an [updatedAt](#) tag is appended to fileName

**3.9.3.9 toString()**

```
string myParsing::toString ( )
```

toString

**Returns**

string

return json dump, for debug purpose

**3.9.4 Member Data Documentation****3.9.4.1 myJson**

```
json myParsing::myJson [private]
```

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

- /home/sylvain/Documents/Travail/ajc-akkodis/formation/LinuxEmbarque/Projet/devel/AirQualityWatch/[myparsing.h](#)
- /home/sylvain/Documents/Travail/ajc-akkodis/formation/LinuxEmbarque/Projet/devel/AirQualityWatch/[myparsing.cpp](#)

**3.10 myRegex Class Reference**

The [myRegex](#) class.

```
#include <myregex.h>
```

**Public Member Functions**

- [myRegex](#) ()  
*myRegex*

## Static Public Member Functions

- static [updatedAt ambeeDataGetDateTime](#) (string updatedAtField)  
*ambeeDataGetDateTime*
- static string [toFormatDDMMYY](#) (string updatedAtField)  
*toFormatDDMMYY*
- static string [toFormatHHMM](#) (string updatedAtField)  
*toFormatHHMM*
- static string [toFormatDDMMYYHHMM](#) (string updatedAtField)  
*toFormatDDMMYYHHMM*

### 3.10.1 Detailed Description

The [myRegex](#) class.

Regex helpers to display dates and times

### 3.10.2 Constructor & Destructor Documentation

#### 3.10.2.1 myRegex()

```
myRegex::myRegex ( )
```

[myRegex](#)

### 3.10.3 Member Function Documentation

#### 3.10.3.1 ambeeDataGetDateTime()

```
updatedAt myRegex::ambeeDataGetDateTime (
    string updatedAtField ) [static]
```

ambeeDataGetDateTime

Parameters

<i>updatedAtField</i>	
-----------------------	--

Returns

### 3.10.3.2 toFormatDDMMYY()

```
string myRegex::toFormatDDMMYY (
    string updatedAtField ) [static]
```

toFormatDDMMYY

#### Parameters

<i>updatedAtField</i>	
-----------------------	--

#### Returns

### 3.10.3.3 toFormatDDMMYYHHMM()

```
string myRegex::toFormatDDMMYYHHMM (
    string updatedAtField ) [static]
```

toFormatDDMMYYHHMM

#### Parameters

<i>updatedAtField</i>	
-----------------------	--

#### Returns

### 3.10.3.4 toFormatHHMM()

```
string myRegex::toFormatHHMM (
    string updatedAtField ) [static]
```

toFormatHHMM

#### Parameters

<i>updatedAtField</i>	
-----------------------	--

#### Returns

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

- [/home/sylvain/Documents/Travail/ajc-akkodis/formation/LinuxEmbarque/Projet/devel/AlrQualityWAtch/myregex.h](#)
- [/home/sylvain/Documents/Travail/ajc-akkodis/formation/LinuxEmbarque/Projet/devel/AlrQualityWAtch/myregex.cpp](#)

## 3.11 updatedAt Struct Reference

```
#include <myregex.h>
```

### Public Attributes

- int [year](#)
- int [month](#)
- int [day](#)
- int [hour](#)
- int [minute](#)
- int [second](#)

### 3.11.1 Member Data Documentation

#### 3.11.1.1 day

```
int updatedAt::day
```

#### 3.11.1.2 hour

```
int updatedAt::hour
```

#### 3.11.1.3 minute

```
int updatedAt::minute
```

#### 3.11.1.4 month

```
int updatedAt::month
```

### 3.11.1.5 second

```
int updatedAt::second
```

### 3.11.1.6 year

```
int updatedAt::year
```

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

- </home/sylvain/Documents/Travail/ajc-akkodis/formation/LinuxEmbarque/Projet/devel/AirQualityWatch/myregex.h>

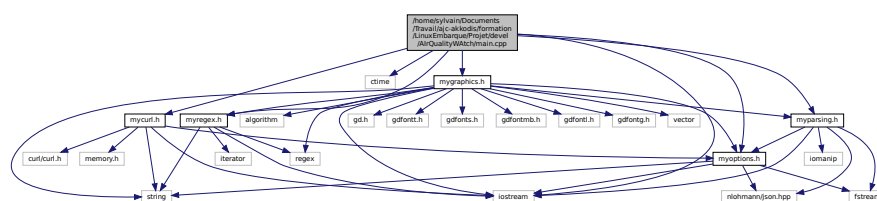


## File Documentation

#### 4.1 /home/sylvain/Documents/Travail/ajc-akkodis/formation/Linux↵ Embarque/Projet/devel/AlrQualityWatch/main.cpp File Reference

```
#include <iostream>
#include <ctime>
#include "mycurl.h"
#include "myparsing.h"
#include "myoptions.h"
#include "mygraphics.h"
#include "myregex.h"
```

Include dependency graph for main.cpp:



## Functions

- int **downloadDatas** (char \*\*data, myOptions \*options)
- int **main** ()

### 4.1.1 Function Documentation

#### 4.1.1.1 downloadDatas()

```
int downloadDatas (
    char ** data,
    myOptions * options )
```

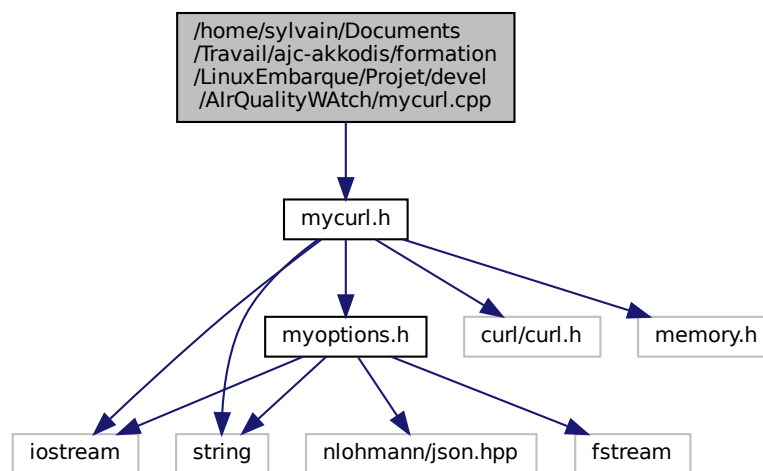
#### 4.1.1.2 main()

```
int main ( )
```

## 4.2 /home/sylvain/Documents/Travail/ajc-akkodis/formation/LinuxEmbarque/Projet/devel/AlrQualityWatch/mycurl.cpp File Reference

```
#include "mycurl.h"
```

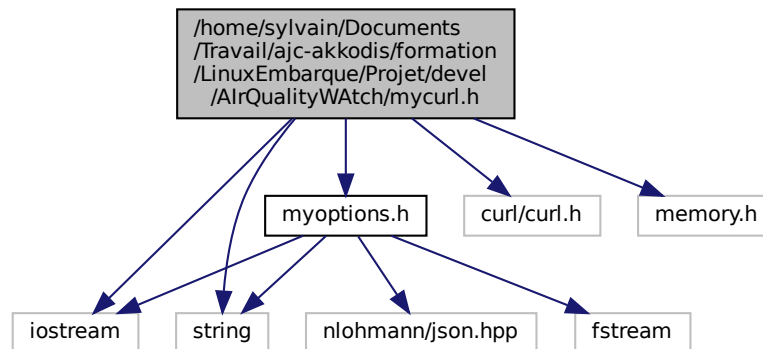
Include dependency graph for mycurl.cpp:



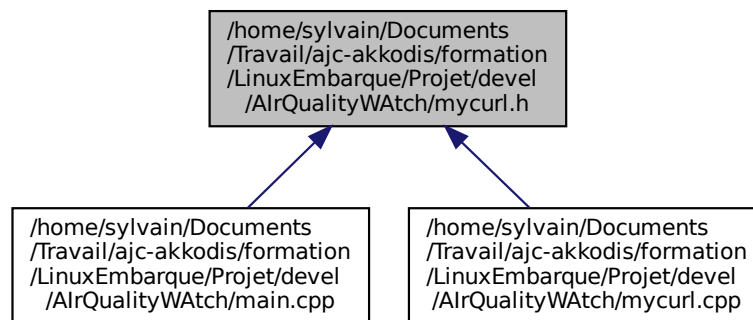
## 4.3 /home/sylvain/Documents/Travail/ajc-akkodis/formation/LinuxEmbarque/Projet/devel/AlrQualityWatch/mycurl.h File Reference

```
#include <iostream>
#include <curl/curl.h>
#include <memory.h>
#include <string>
#include "myoptions.h"
```

Include dependency graph for mycurl.h:



This graph shows which files directly or indirectly include this file:



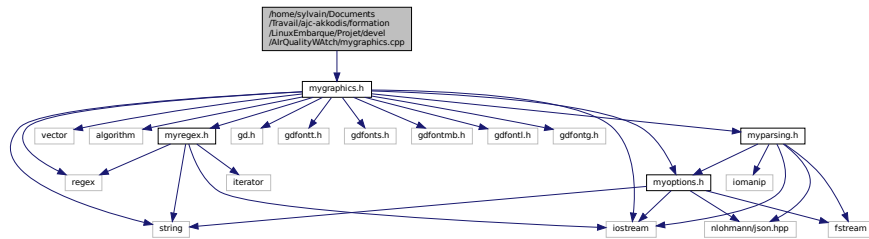
## Classes

- struct [memory](#)
  - class [myCurl](#)
- The [myCurl](#) class.

## 4.4 /home/sylvain/Documents/Travail/ajc-akkodis/formation/Linux↔ Embarque/Projet/devel/AlrQualityWATCH/mygraphics.cpp File Reference

```
#include "mygraphics.h"
```

Include dependency graph for mygraphics.cpp:



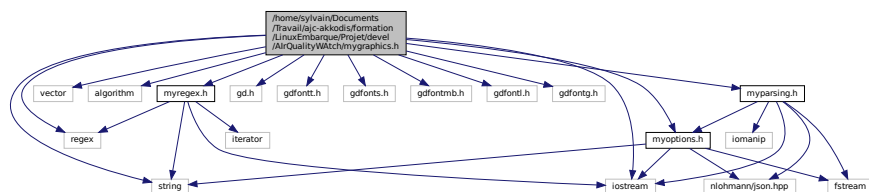
## 4.5 /home/sylvain/Documents/Travail/ajc-akkodis/formation/LinuxEmbarque/Projet/devel/AlrQualityWAtch/mygraphics.h File Reference

```

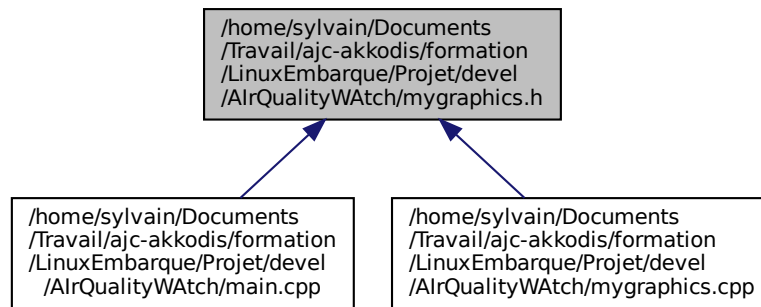
#include <iostream>
#include <string>
#include <vector>
#include <algorithm>
#include <regex>
#include <gd.h>
#include <gdfontt.h>
#include <gdfonts.h>
#include <gdfontmb.h>
#include <gdfontl.h>
#include <gdfontg.h>
#include "myoptions.h"
#include "myparsing.h"
#include "myregex.h"

```

Include dependency graph for mygraphics.h:



This graph shows which files directly or indirectly include this file:



## Classes

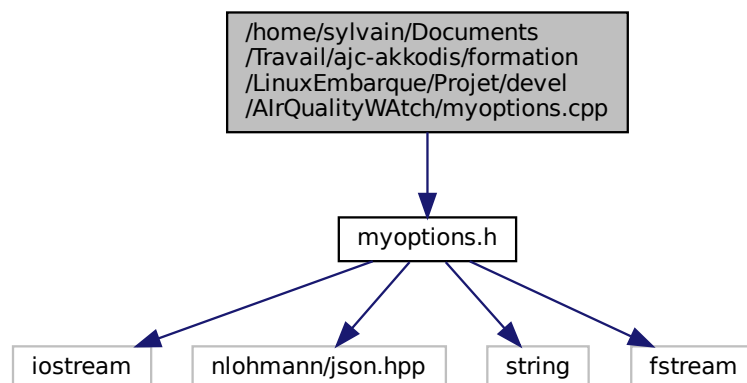
- struct [Color](#)
- struct [DataElement](#)
- struct [dataSet](#)
- struct [CurveChartParams](#)
- class [myGraphics](#)

*The [myGraphics](#) class.*

## 4.6 /home/sylvain/Documents/Travail/ajc-akkodis/formation/LinuxEmbarque/Projet/devel/AlrQualityWatch/myoptions.cpp File Reference

```
#include "myoptions.h"
```

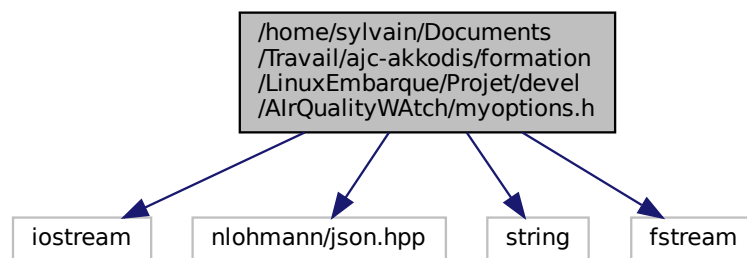
Include dependency graph for `myoptions.cpp`:



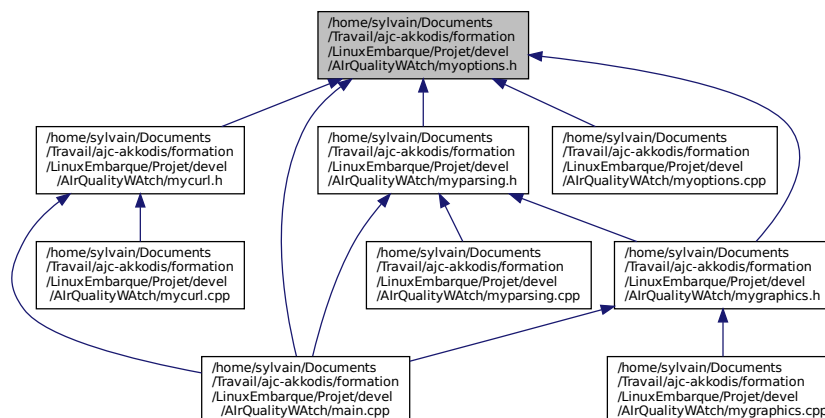
## 4.7 /home/sylvain/Documents/Travail/ajc-akkodis/formation/LinuxEmbarque/Projet/devel/AlrQualityWAtch/myoptions.h File Reference

```
#include <iostream>
#include <nlohmann/json.hpp>
#include <string>
#include <fstream>
```

Include dependency graph for myoptions.h:



This graph shows which files directly or indirectly include this file:



### Classes

- class [myOptions](#)  
The *myOptions* class.

### Typedefs

- using [json](#) = nlohmann::json

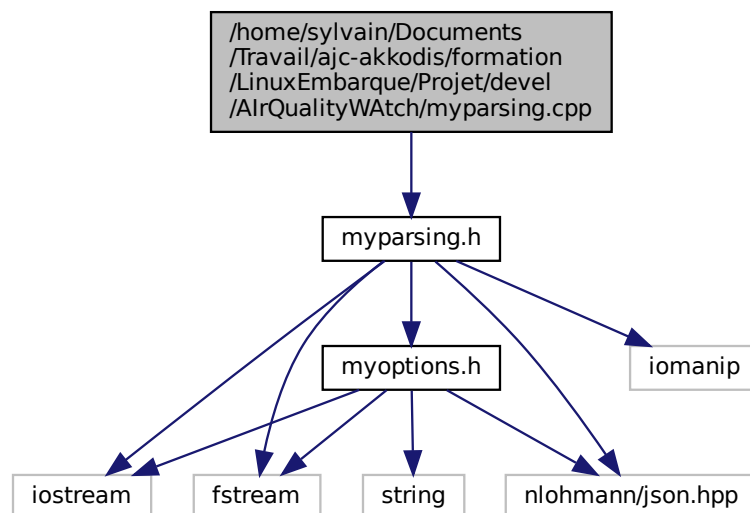
### 4.7.1 Typedef Documentation

#### 4.7.1.1 json

```
using json = nlohmann::json
```

## 4.8 /home/sylvain/Documents/Travail/ajc-akkodis/formation/Linux↔ Embarque/Projet/devel/AlrQualityWAtch/myparsing.cpp File Reference

```
#include "myparsing.h"  
Include dependency graph for myparsing.cpp:
```

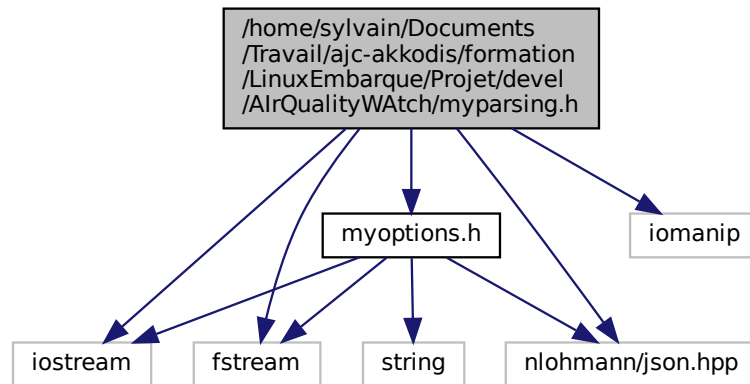


## 4.9 /home/sylvain/Documents/Travail/ajc-akkodis/formation/Linux↔ Embarque/Projet/devel/AlrQualityWAtch/myparsing.h File Reference

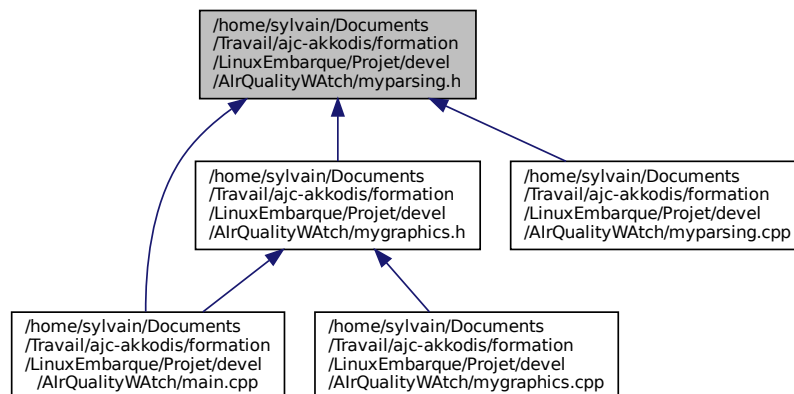
```
#include <iostream>  
#include <fstream>  
#include <iomanip>  
#include <nlohmann/json.hpp>
```

```
#include "myoptions.h"
```

Include dependency graph for `myparsing.h`:



This graph shows which files directly or indirectly include this file:



## Classes

- class `myParsing`  
The `myParsing` class.

## Typedefs

- using `json` = `nlohmann::json`

### 4.9.1 Typedef Documentation

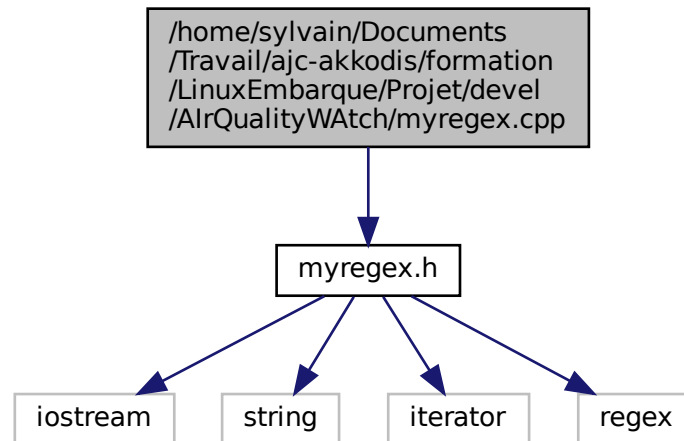


#### 4.9.1.1 json

```
using json = nlohmann::json
```

### 4.10 /home/sylvain/Documents/Travail/ajc-akkodis/formation/Linux↵ Embarque/Projet/devel/AlrQualityWAtch/myregex.cpp File Reference

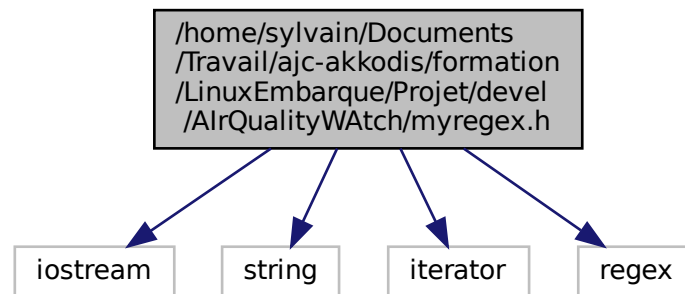
```
#include "myregex.h"  
Include dependency graph for myregex.cpp:
```



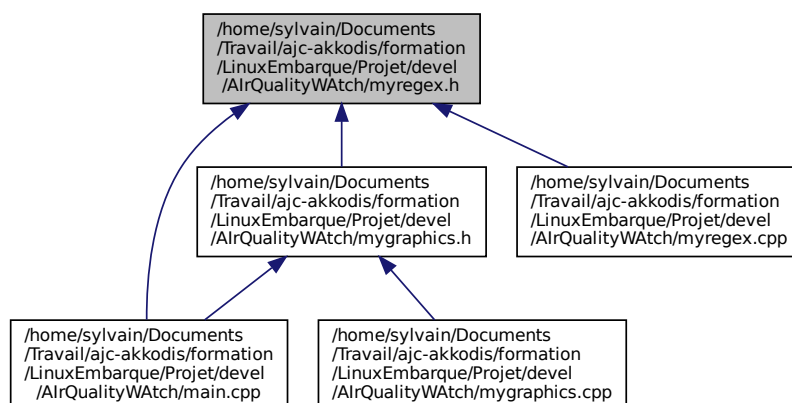
### 4.11 /home/sylvain/Documents/Travail/ajc-akkodis/formation/Linux↵ Embarque/Projet/devel/AlrQualityWAtch/myregex.h File Reference

```
#include <iostream>  
#include <string>  
#include <iterator>  
#include <regex>
```

Include dependency graph for `myregex.h`:



This graph shows which files directly or indirectly include this file:



## Classes

- struct `updatedAt`
- class `myRegex`

The `myRegex` class.

# Index

/home/sylvain/Documents/Travail/ajc-akkodis/formation/LinuxEmbarque/Projet/devel/AirQualityWatch/main.cpp, 35

/home/sylvain/Documents/Travail/ajc-akkodis/formation/LinuxEmbarque/Projet/devel/AirQualityWatch/mycurl.cpp, 36

/home/sylvain/Documents/Travail/ajc-akkodis/formation/LinuxEmbarque/Projet/devel/AirQualityWatch/mycurl.h, 36

/home/sylvain/Documents/Travail/ajc-akkodis/formation/LinuxEmbarque/Projet/devel/AirQualityWatch/mygraphics.cpp, 37

/home/sylvain/Documents/Travail/ajc-akkodis/formation/LinuxEmbarque/Projet/devel/AirQualityWatch/mygraphics.h, 38

/home/sylvain/Documents/Travail/ajc-akkodis/formation/LinuxEmbarque/Projet/devel/AirQualityWatch/myoptions.cpp, 39

/home/sylvain/Documents/Travail/ajc-akkodis/formation/LinuxEmbarque/Projet/devel/AirQualityWatch/myoptions.h, 40

/home/sylvain/Documents/Travail/ajc-akkodis/formation/LinuxEmbarque/Projet/devel/AirQualityWatch/myparsing.cpp, 41

/home/sylvain/Documents/Travail/ajc-akkodis/formation/LinuxEmbarque/Projet/devel/AirQualityWatch/myparsing.h, 41

/home/sylvain/Documents/Travail/ajc-akkodis/formation/LinuxEmbarque/Projet/devel/AirQualityWatch/myregex.cpp, 43

/home/sylvain/Documents/Travail/ajc-akkodis/formation/LinuxEmbarque/Projet/devel/AirQualityWatch/myregex.h, 43

~myGraphics

    myGraphics, 14

abscisse

    DataElement, 8

ambeeDataGetDateTime

    myRegex, 31

appendDataVector

    myGraphics, 15

appendFileToDatas

    myParsing, 27

appendToDatas

    myParsing, 27

blue

    Color, 5

calcRatioPourcent

    myGraphics, 15

Color, 5

    blue, 5

    green, 5

    indice, 5

    nom, 5

    red, 6

colorIndex

    CurveChartParams, 6

    dataSet, 9

CurveChartParams, 6

dataElement, 8

    abscisse, 8

    label, 8

    value, 8

dataElements

    dataSet, 9

dataSet, 9

    colorIndex, 9

    dataElements, 9

    legend, 9

dataSets

    CurveChartParams, 6

day

    updatedAt, 33

description

    CurveChartParams, 6

downloadDatas

    main.cpp, 35

exec

    myCurl, 11

ficOut

    myGraphics, 20

fonts

    myGraphics, 20

fromChar

- myParsing, 28
- fromDatasFile
  - myParsing, 28
- fromFile
  - myParsing, 28
- gdColors
  - myGraphics, 20
- getApiKeyHeader
  - myOptions, 23
- getData
  - myCurl, 11
- getDataPathName
  - myOptions, 23
- getDataVectorMaxValue
  - myGraphics, 17
- getDataVectorMinValue
  - myGraphics, 17
- getFilesOwner
  - myOptions, 23
- getFullCurveChartFileName
  - myOptions, 23
- getFullDataFileName
  - myOptions, 24
- getFullHistoChartFileName
  - myOptions, 24
- getFullLogFileName
  - myOptions, 24
- getFullPieChartFileName
  - myOptions, 24
- getFullUrl
  - myOptions, 25
- getMyJson
  - myParsing, 29
- globalMax
  - CurveChartParams, 7
- globalMin
  - CurveChartParams, 7
- green
  - Color, 5
- hnd
  - myCurl, 12
- hour
  - updatedAt, 33
- im
  - myGraphics, 20
- imageSize
  - myGraphics, 20
- indice
  - Color, 5
- initFonts
  - myGraphics, 18
- initPalette
  - myGraphics, 18
- isDisplay
  - myOptions, 25
- json
  - myoptions.h, 41
  - myparsing.h, 42
- label
  - DataElement, 8
- legend
  - dataSet, 9
- legendWidth
  - CurveChartParams, 7
- main
  - main.cpp, 35
- main.cpp
  - downloadDatas, 35
  - main, 35
- marginBottom
  - CurveChartParams, 7
- marginLeft
  - CurveChartParams, 7
- mem\_cb
  - myCurl, 12
- memory, 9
  - response, 10
  - size, 10
- minute
  - updatedAt, 33
- month
  - updatedAt, 33
- myCurl, 10
  - exec, 11
  - getData, 11
  - hnd, 12
  - mem\_cb, 12
  - myCurl, 11
- myGraphics, 13
  - ~myGraphics, 14
  - appendDataVector, 15
  - calcRatioPourcent, 15
  - createColor, 16
  - curveChart, 16
  - curveChartAddCurves, 16
  - curveChartInit, 17
  - curveChartSetLegend, 17
  - ficOut, 20
  - fonts, 20
  - gdColors, 20
  - getDataVectorMaxValue, 17
  - getDataVectorMinValue, 17
  - im, 20
  - imageSize, 20
  - initFonts, 18
  - initPalette, 18
  - myGraphics, 14
  - myPalette, 20
  - pieChart, 18
  - pieChartDraw, 19
  - pieChartInit, 19
  - setVector, 19

- titre, [20](#)
- myJson
  - myParsing, [30](#)
- myOptions, [21](#)
  - getApiKeyHeader, [23](#)
  - getDataPathName, [23](#)
  - getFilesOwner, [23](#)
  - getFullCurveChartFileName, [23](#)
  - getFullDataFileName, [24](#)
  - getFullHistoChartFileName, [24](#)
  - getFullLogFileName, [24](#)
  - getFullPieChartFileName, [24](#)
  - getFullUrl, [25](#)
  - isDisplay, [25](#)
  - myOptions, [22](#)
  - options, [26](#)
  - readFromFile, [25](#)
- myoptions.h
  - json, [41](#)
- myPalette
  - myGraphics, [20](#)
- myParsing, [26](#)
  - appendFileToDatas, [27](#)
  - appendToDatas, [27](#)
  - fromChar, [28](#)
  - fromDatasFile, [28](#)
  - fromFile, [28](#)
  - getMyJson, [29](#)
  - myJson, [30](#)
  - myParsing, [27](#)
  - setMyJson, [29](#)
  - toFile, [29](#)
  - toString, [30](#)
- myparsing.h
  - json, [42](#)
- myRegex, [30](#)
  - ambeeDataGetDateTime, [31](#)
  - myRegex, [31](#)
  - toFormatDDMMYY, [31](#)
  - toFormatDDMMYYHHMM, [32](#)
  - toFormatHHMM, [32](#)
- nbMeasures
  - CurveChartParams, [7](#)
- nom
  - Color, [5](#)
- options
  - myOptions, [26](#)
- pieChart
  - myGraphics, [18](#)
- pieChartDraw
  - myGraphics, [19](#)
- pieChartInit
  - myGraphics, [19](#)
- readFromFile
  - myOptions, [25](#)
- red
  - Color, [6](#)
- response
  - memory, [10](#)
- second
  - updatedAt, [33](#)
- setMyJson
  - myParsing, [29](#)
- setVector
  - myGraphics, [19](#)
- size
  - memory, [10](#)
- title
  - CurveChartParams, [7](#)
- titleHeight
  - CurveChartParams, [7](#)
- titre
  - myGraphics, [20](#)
- toFile
  - myParsing, [29](#)
- toFormatDDMMYY
  - myRegex, [31](#)
- toFormatDDMMYYHHMM
  - myRegex, [32](#)
- toFormatHHMM
  - myRegex, [32](#)
- toString
  - myParsing, [30](#)
- updatedAt, [33](#)
  - day, [33](#)
  - hour, [33](#)
  - minute, [33](#)
  - month, [33](#)
  - second, [33](#)
  - year, [34](#)
- value
  - DataElement, [8](#)
- year
  - updatedAt, [34](#)