

## jsonToBatProject

Generated on Mon Feb 26 2024 16:15:29 for jsonToBatProject by Doxygen 1.9.8

Mon Feb 26 2024 16:15:29



---

<b>1 README</b>	<b>1</b>
1.1 README	1
1.1.1 Current workflows:	1
1.1.2 Regarding coding style (?):	1
1.1.3 Git (?):	1
<b>2 Todo List</b>	<b>3</b>
<b>3 Namespace Index</b>	<b>5</b>
3.1 Namespace List	5
<b>4 File Index</b>	<b>7</b>
4.1 File List	7
<b>5 Namespace Documentation</b>	<b>9</b>
5.1 WIP Namespace Reference	9
5.1.1 Detailed Description	9
5.1.2 Function Documentation	9
5.1.2.1 exampleEasyLogging()	9
<b>6 File Documentation</b>	<b>11</b>
6.1 README.md File Reference	11
6.2 src/main.cpp File Reference	11
6.2.1 Function Documentation	11
6.2.1.1 main()	11
6.3 main.cpp	12
<b>Index</b>	<b>13</b>



# Chapter 1

## README

Doxygen Documentation

Sonar Cloud

### 1.1 README

#### 1.1.1 Current workflows:

- build
  - build and test the application on:
    - \* windows with cl
    - \* ubuntu with g++
    - \* ubuntu with clang++
- CodeQL
  - Code security
- Doxygen Action
  - Generate Doxygen documentation
  - Deploys generated documentation to gh-pages
- Microsoft C++ Code Analysis
- pages-build-deployment
- SonarCloud
  - Static code analysis *For Scanning Alerts -> Security*

#### 1.1.2 Regarding coding style (?):

- no classes in global namespace
- no "using NAMESPACE"
- 4 space indenting
- ? *setup astyle options?*

#### 1.1.3 Git (?):

- no direct commits onto main (only via pull-requests)



## Chapter 2

# Todo List

Global `main` (int argc, char \*argv[])

Github

- "Dev-Ops"
- Doxygen settings
- Template-Comment
- Template-Header-Comment

Global `WIP::exampleEasyLogging` ()

Configure easylogging properly

- outsource easylogging config
  - e.g. startup class?





## Chapter 3

# Namespace Index

### 3.1 Namespace List

Here is a list of all namespaces with brief descriptions:

<a href="#">WIP</a>	Namespace for work in progress . . . . .	<a href="#">9</a>
---------------------	--	-------------------



# Chapter 4

## File Index

### 4.1 File List

Here is a list of all files with brief descriptions:

src/ <a href="#">main.cpp</a> . . . . .	11
---	----



## Chapter 5

# Namespace Documentation

### 5.1 WIP Namespace Reference

Namespace for work in progress.

#### Functions

- void [exampleEasyLogging](#) ()  
*Example of how to use easylogging with a configuration file.*

#### 5.1.1 Detailed Description

Namespace for work in progress.

- This namespace is for code that is not finished yet
- It is used to keep the main namespace clean

#### 5.1.2 Function Documentation

##### 5.1.2.1 [exampleEasyLogging\(\)](#)

```
void WIP::exampleEasyLogging ( )
```

Example of how to use easylogging with a configuration file.

- This function is an example of how to use easylogging
- The configuration file is located in ../conf
- Before proper integration, config has to be done properly

**Todo**

Definition at line [52](#) of file [main.cpp](#).



# Chapter 6

## File Documentation

### 6.1 README.md File Reference

### 6.2 src/main.cpp File Reference

```
#include <easylogging++.h>
#include <iostream>
```

#### Namespaces

- namespace [WIP](#)  
*Namespace for work in progress.*

#### Functions

- void [WIP::exampleEasyLogging](#) ()  
*Example of how to use easylogging with a configuration file.*
- int [main](#) (int argc, char \*argv[ ])  
*Main function.*

#### 6.2.1 Function Documentation

##### 6.2.1.1 main()

```
int main (
    int argc,
    char * argv[ ] )
```

Main function.

Codeconvention:

- Formatter: `astyle`

#### [Todo](#)

Definition at line [25](#) of file [main.cpp](#).

References [WIP::exampleEasyLogging\(\)](#).

## 6.3 main.cpp

[Go to the documentation of this file.](#)

```
00001 #include <easylogging++.h>
00002 #include <iostream>
00003
00004
00005 INITIALIZE_EASYLOGGINGPP
00006
00007 namespace WIP {
00008 void exampleEasyLogging();
00009 }
00010
00025 int main(int argc, char *argv[]) {
00026     WIP::exampleEasyLogging();
00027     std::cout << "Hello, World!" << std::endl;
00028     return 0;
00029 }
00030
00038 namespace WIP {
00052 void exampleEasyLogging() {
00053     el::Configurations conf("conf/easylogging.conf");
00054     el::Loggers::reconfigureLogger("default", conf);
00055     el::Loggers::reconfigureAllLoggers(conf);
00056
00057     LOG(INFO) << "My first info log using default logger";
00058 }
00059 } // namespace WIP
```



# Index

exampleEasyLogging  
WIP, [9](#)

main  
main.cpp, [11](#)  
main.cpp  
main, [11](#)

README, [1](#)  
README.md, [11](#)

src/main.cpp, [11](#), [12](#)

Todo List, [3](#)

WIP, [9](#)  
exampleEasyLogging, [9](#)