

jsonToBatProject

Generated on Mon Feb 26 2024 17:05:02 for jsonToBatProject by Doxygen 1.10.0

Mon Feb 26 2024 17:05:02

1 README	1
1.1 README	1
1.1.1 Current workflows:	1
1.1.2 Regarding coding style (?):	1
1.1.3 Git (?):	1
2 Todo List	3
3 Namespace Index	5
3.1 Namespace List	5
4 File Index	7
4.1 File List	7
5 Namespace Documentation	9
5.1 WIP Namespace Reference	9
5.1.1 Detailed Description	9
5.1.2 Function Documentation	9
5.1.2.1 exampleEasyLogging()	9
6 File Documentation	11
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
Index	13

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

Namespace **WIP**

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:

WIP	Namespace for work in progress	9
---------------------	--	-------------------

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

src/ main.cpp	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.

- Todo**
- Github
 - "Dev-Ops"
 - Doxygen settings
 - Template-Comment
 - Template-Header-Comment

- 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 [53](#) 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`

Definition at line [26](#) 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 /** \todo
00005  * - Github
00006  * - "Dev-Ops"
00007  * - Doxygen settings
00008  * - Template-Comment
00009  * - Template-Header-Comment
00010  *
00011  */
00012
00013 INITIALIZE_EASYLOGGINGPP
00014
00015 namespace WIP {
00016 void exampleEasyLogging();
00017 }
00018
00019 /**
00020  * \brief Main function
00021  *
00022  * \details
00023  * Codeconvention:
00024  * - Formatter: astyle
00025  */
00026 int main(int argc, char *argv[]) {
00027     WIP::exampleEasyLogging();
00028     std::cout << "Hello, World!" << std::endl;
00029     return 0;
00030 }
00031
00032 /**
00033  * \brief Namespace for work in progress
00034  *
00035  * \details
00036  * - This namespace is for code that is not finished yet
00037  * - It is used to keep the main namespace clean
00038  */
00039 namespace WIP {
00040 /**
00041  * \brief Example of how to use easylogging with a configuration file
00042  *
00043  * \details
00044  * - This function is an example of how to use easylogging
00045  * - The configuration file is located in ../conf
00046  * - Before proper integration, config has to be done properly
00047  *
00048  * \todo
00049  * - Configure easylogging properly
00050  * - outsource easylogging config
00051  * - e.g. startup class?
00052  */
00053 void exampleEasyLogging() {
00054     el::Configurations conf("conf/easylogging.conf");
00055     el::Loggers::reconfigureLogger("default", conf);
00056     el::Loggers::reconfigureAllLoggers(conf);
00057
00058     LOG(INFO) << "My first info log using default logger";
00059 }
00060 } // 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](#)