

Writing Regexprs 2021-22 / Regex Explainer

Generated by Doxygen 1.9.3

1 Namespace Index	1
1.1 Namespace List	1
2 Class Index	3
2.1 Class List	3
3 File Index	5
3.1 File List	5
4 Namespace Documentation	7
4.1 wr22 Namespace Reference	7
4.2 wr22::regex_explainer Namespace Reference	7
4.3 wr22::regex_explainer::explanation Namespace Reference	7
4.3.1 Typedef Documentation	8
4.3.1.1 type_of_sentence	8
4.3.2 Function Documentation	8
4.3.2.1 get_full_explanation()	8
4.3.2.2 get_sample() [1/10]	9
4.3.2.3 get_sample() [2/10]	9
4.3.2.4 get_sample() [3/10]	9
4.3.2.5 get_sample() [4/10]	10
4.3.2.6 get_sample() [5/10]	10
4.3.2.7 get_sample() [6/10]	10
4.3.2.8 get_sample() [7/10]	10
4.3.2.9 get_sample() [8/10]	10
4.3.2.10 get_sample() [9/10]	10
4.3.2.11 get_sample() [10/10]	11
4.3.2.12 to_json() [1/2]	11
4.3.2.13 to_json() [2/2]	11
4.3.2.14 upgrade_sample()	11
4.4 wr22::regex_explainer::hints Namespace Reference	11
4.4.1 Function Documentation	12
4.4.1.1 get_hint() [1/5]	12
4.4.1.2 get_hint() [2/5]	12
4.4.1.3 get_hint() [3/5]	12
4.4.1.4 get_hint() [4/5]	12
4.4.1.5 get_hint() [5/5]	12
5 Class Documentation	13
5.1 wr22::regex_explainer::explanation::Explanation Class Reference	13
5.1.1 Detailed Description	13
5.1.2 Constructor & Destructor Documentation	13
5.1.2.1 Explanation() [1/2]	13
5.1.2.2 Explanation() [2/2]	13

5.1.3 Member Function Documentation	14
5.1.3.1 get_depth()	14
5.1.3.2 get_explanation()	14
5.1.3.3 is_bold()	14
5.2 wr22::regex_explainer::explanation::Hint Class Reference	15
5.2.1 Detailed Description	15
5.2.2 Constructor & Destructor Documentation	15
5.2.2.1 Hint() [1/2]	15
5.2.2.2 Hint() [2/2]	15
5.2.3 Member Function Documentation	16
5.2.3.1 get_additional_information()	16
5.2.3.2 get_hint()	16
5.2.4 Member Data Documentation	16
5.2.4.1 main_sentence	16
6 File Documentation	17
6.1 include/wr22/regex_explainer/explanation/explanation.hpp File Reference	17
6.2 explanation.hpp	17
6.3 include/wr22/regex_explainer/explanation/explanation_folder.hpp File Reference	18
6.4 explanation_folder.hpp	19
6.5 include/wr22/regex_explainer/explanation/sample_folder.hpp File Reference	19
6.6 sample_folder.hpp	20
6.7 include/wr22/regex_explainer/explanation/upgrade_sample.hpp File Reference	21
6.8 upgrade_sample.hpp	21
6.9 include/wr22/regex_explainer/hints/hint.hpp File Reference	21
6.10 hint.hpp	22
6.11 include/wr22/regex_explainer/hints/hint_folder.hpp File Reference	22
6.12 hint_folder.hpp	23
6.13 src/explanation/explanation.cpp File Reference	23
6.14 src/explanation/explanation_folder.cpp File Reference	24
6.15 src/explanation/sample_folder.cpp File Reference	24
6.16 src/explanation/upgrade_sample.cpp File Reference	25
6.17 src/hints/hint.cpp File Reference	25
6.18 src/hints/hint_folder.cpp File Reference	25
Index	27

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

wr22	7
wr22::regex_explainer	7
wr22::regex_explainer::explanation	7
wr22::regex_explainer::hints	11

Chapter 2

Class Index

2.1 Class List

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

wr22::regex_explainer::explanation::Explanation	
A class for holding an explanation tree element	13
wr22::regex_explainer::explanation::Hint	
A class for holding a hint	15

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

include/wr22/regex_explainer/explanation/ explanation.hpp	17
include/wr22/regex_explainer/explanation/ explanation_folder.hpp	18
include/wr22/regex_explainer/explanation/ sample_folder.hpp	19
include/wr22/regex_explainer/explanation/ upgrade_sample.hpp	21
include/wr22/regex_explainer/hints/ hint.hpp	21
include/wr22/regex_explainer/hints/ hint_folder.hpp	22
src/explanation/ explanation.cpp	23
src/explanation/ explanation_folder.cpp	24
src/explanation/ sample_folder.cpp	24
src/explanation/ upgrade_sample.cpp	25
src/hints/ hint.cpp	25
src/hints/ hint_folder.cpp	25

Chapter 4

Namespace Documentation

4.1 wr22 Namespace Reference

Namespaces

- namespace [regex_explainer](#)

4.2 wr22::regex_explainer Namespace Reference

Namespaces

- namespace [explanation](#)
- namespace [hints](#)

4.3 wr22::regex_explainer::explanation Namespace Reference

Classes

- class [Explanation](#)
A class for holding an explanation tree element.
- class [Hint](#)
A class for holding a hint.

Typedefs

- typedef wr22::utils::Adt< std::string, std::u32string, uint32_t > [type_of_sentence](#)

Functions

- `std::vector< Explanation > get_full_explanation (const SpannedPart &spanned_part, size_t depth=0)`
The function returns the vector processed information about each node of the syntax tree.
- `void to_json (nlohmann::json &j, const Explanation &explanation)`
- `std::string get_sample (const Empty &vertex)`
The function returns a sample for the AST node depending on its type.
- `std::vector< std::string > get_sample (const Literal &vertex)`
The function describes which literal matched and what index it has in the ASCII table.
- `std::string get_sample (const Alternatives &vertex)`
The function lists all alternatives that needed to be mached and describes each one in detail.
- `std::string get_sample (const Sequence &vertex)`
The function lists all items of the sequence list and match there explanation one after another.
- `std::vector< std::string > get_sample (const Group &vertex)`
he function says that it is group and describes it in detail.
- `std::string get_sample (const Optional &vertex)`
Next three functions return the explanation of some quantifier.
- `std::string get_sample (const Plus &vertex)`
- `std::string get_sample (const Star &vertex)`
- `std::string get_sample (const Wildcard &vertex)`
The function returns the explanation for some regex basic that is always the same and does not require any modifications depending on the situation.
- `std::vector< std::string > get_sample (const CharacterClass &vertex)`
The function lists the symbols from the certain set that need to be matched.
- `std::string upgrade_sample (const std::string &symbol, const std::string &sample)`
Upgrades the default sample and makes it more specific.
- `void to_json (nlohmann::json &j, const Hint &hint)`

4.3.1 Typedef Documentation

4.3.1.1 type_of_sentence

```
typedef wr22::utils::Adt<std::string, std::u32string, uint32_t> wr22::regex\_explainer::explanation::type\_of\_s
```

4.3.2 Function Documentation

4.3.2.1 get_full_explanation()

```
std::vector< Explanation > wr22::regex_explainer::explanation::get_full_explanation (
    const SpannedPart & spanned_part,
    size_t depth = 0 )
```

The function returns the vector processed information about each node of the syntax tree.

Thus, a complete description of the work of the written regular expression is created.

Parameters

<i>spanned_part</i>	is the current node of syntax tree to be explained.
<i>depth</i>	is the depth of the current explanation in the resulting tree of explanations.

Returns

a vector of elements that have type of class [Explanation](#).

4.3.2.2 get_sample() [1/10]

```
std::string wr22::regex_explainer::explanation::get_sample (
    const Alternatives & vertex )
```

The function lists all alternatives that needed to be mached and describes each one in detail.

4.3.2.3 get_sample() [2/10]

```
std::vector< std::string > wr22::regex_explainer::explanation::get_sample (
    const CharacterClass & vertex )
```

The function lists the symbols from the certain set that need to be matched.

The function is case sensitive.

4.3.2.4 get_sample() [3/10]

```
std::string wr22::regex_explainer::explanation::get_sample (
    const Empty & vertex )
```

The function returns a sample for the AST node depending on its type.

This is functions that returns the sample for the node from AST as the std::string_view.

If the sample consists of one part, then simply returns std::string. If the sample consists of several parts, then a vector of patterns with type std::string is returned, which should then make up a whole sentence / paragraph.

The function is overloaded and returns a different pattern of type "std::string_view" depending on the type of the argument it takes.

The function for type "Empty" is always default because it does not have any meaningful information.

4.3.2.5 `get_sample()` [4/10]

```
std::vector< std::string > wr22::regex_explainer::explanation::get_sample (
    const Group & vertex )
```

he function says that it is group and describes it in detail.

4.3.2.6 `get_sample()` [5/10]

```
std::vector< std::string > wr22::regex_explainer::explanation::get_sample (
    const Literal & vertex )
```

The function describes which literal matched and what index it has in the ASCII table.

The function is case sensitive.

4.3.2.7 `get_sample()` [6/10]

```
std::string wr22::regex_explainer::explanation::get_sample (
    const Optional & vertex )
```

Next three functions return the explanation of some quantifier.

They are always the same and do not require any modifications depending on the situation.

4.3.2.8 `get_sample()` [7/10]

```
std::string wr22::regex_explainer::explanation::get_sample (
    const Plus & vertex )
```

4.3.2.9 `get_sample()` [8/10]

```
std::string wr22::regex_explainer::explanation::get_sample (
    const Sequence & vertex )
```

The function lists all items of the sequence list and match there explanation one after another.

There is no information about the type of this vertex.

4.3.2.10 `get_sample()` [9/10]

```
std::string wr22::regex_explainer::explanation::get_sample (
    const Star & vertex )
```

4.3.2.11 get_sample() [10/10]

```
std::string wr22::regex_explainer::explanation::get_sample (
    const Wildcard & vertex )
```

The function returns the explanation for some regex basic that is always the same and does not require any modifications depending on the situation.

4.3.2.12 to_json() [1/2]

```
void wr22::regex_explainer::explanation::to_json (
    nlohmann::json & j,
    const Explanation & explanation )
```

4.3.2.13 to_json() [2/2]

```
void wr22::regex_explainer::explanation::to_json (
    nlohmann::json & j,
    const Hint & hint )
```

4.3.2.14 upgrade_sample()

```
std::string wr22::regex_explainer::explanation::upgrade_sample (
    const std::string & symbol,
    const std::string & sample )
```

Upgrades the default sample and makes it more specific.

Parameters

<i>symbol</i>	the part to be added to the default sample.
<i>sample</i>	initial default sample to be upgraded.

Returns

the upgraded sample in string type

4.4 wr22::regex_explainer::hints Namespace Reference

Functions

- [Hint get_hint](#) (const ParseError &error)

- [Hint get_hint](#) (const UnexpectedEnd &error)
If any exception was thrown, then there is an error in the regular expression.
- [Hint get_hint](#) (const ExpectedEnd &error)
- [Hint get_hint](#) (const UnexpectedChar &error)
- [Hint get_hint](#) (const InvalidRange &error)

4.4.1 Function Documentation

4.4.1.1 `get_hint()` [1/5]

```
Hint wr22::regex_explainer::hints::get_hint (
    const ExpectedEnd & error )
```

4.4.1.2 `get_hint()` [2/5]

```
Hint wr22::regex_explainer::hints::get_hint (
    const InvalidRange & error )
```

4.4.1.3 `get_hint()` [3/5]

```
Hint wr22::regex_explainer::hints::get_hint (
    const ParseError & error )
```

4.4.1.4 `get_hint()` [4/5]

```
Hint wr22::regex_explainer::hints::get_hint (
    const UnexpectedChar & error )
```

4.4.1.5 `get_hint()` [5/5]

```
Hint wr22::regex_explainer::hints::get_hint (
    const UnexpectedEnd & error )
```

If any exception was thrown, then there is an error in the regular expression.

If the error is related to syntax, then this function returns a hint on how to fix it. In case of other errors, the function returns information about where the failure occurred and some additional information in some cases.

Chapter 5

Class Documentation

5.1 wr22::regex_explainer::explanation::Explanation Class Reference

A class for holding an explanation tree element.

```
#include <explanation_folder.hpp>
```

Public Member Functions

- [Explanation](#) ()
- [Explanation](#) ([type_of_sentence](#) explanation_, [size_t](#) depth_, [bool](#) bold_=false)
Constructor.
- [type_of_sentence](#) [get_explanation](#) () const
- [size_t](#) [get_depth](#) () const
- [bool](#) [is_bold](#) () const

5.1.1 Detailed Description

A class for holding an explanation tree element.

5.1.2 Constructor & Destructor Documentation

5.1.2.1 Explanation() [1/2]

```
wr22::regex_explainer::explanation::Explanation::Explanation ( )
```

5.1.2.2 Explanation() [2/2]

```
wr22::regex_explainer::explanation::Explanation::Explanation (
    type\_of\_sentence explanation_,
    size\_t depth_,
    bool bold_ = false )
```

Constructor.

Parameters

<i>explanation</i> ↔ —	is the variant type that represents some part of explanation.
<i>depth_</i>	is the depth of the current explanation in the resulting tree of explanations.
<i>bold</i>	is boolean variable indicating the need to highlight the color in bold.

5.1.3 Member Function Documentation

5.1.3.1 get_depth()

```
size_t wr22::regex_explainer::explanation::Explanation::get_depth ( ) const
```

Returns

the depth of current spanned_part

5.1.3.2 get_explanation()

```
type_of_sentence wr22::regex_explainer::explanation::Explanation::get_explanation ( ) const
```

Returns

the explanation for current spanned_part

5.1.3.3 is_bold()

```
bool wr22::regex_explainer::explanation::Explanation::is_bold ( ) const
```

Returns

the bold type indicator

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

- include/wr22/regex_explainer/explanation/[explanation_folder.hpp](#)
- src/explanation/[explanation_folder.cpp](#)

5.2 wr22::regex_explainer::explanation::Hint Class Reference

A class for holding a hint.

```
#include <hint_folder.hpp>
```

Public Member Functions

- [Hint](#) ()
- [Hint](#) (std::string hint_, std::string additional_info_="")
Constructor.
- std::string [get_hint](#) () const
- std::string [get_additional_information](#) () const

Static Public Attributes

- static constexpr const char * [main_sentence](#)
This is a sentence for any error indicating that there is an error in the regular expression.

5.2.1 Detailed Description

A class for holding a hint.

5.2.2 Constructor & Destructor Documentation

5.2.2.1 Hint() [1/2]

```
wr22::regex_explainer::explanation::Hint::Hint ( )
```

5.2.2.2 Hint() [2/2]

```
wr22::regex_explainer::explanation::Hint::Hint (
    std::string hint_,
    std::string additional_info_ = "" ) [explicit]
```

Constructor.

Parameters

<i>hint_</i>	is a string containing information about the location where the error occurred and the expected character.
<i>additional_info_</i> ↩	if an optional variable containing additional information about the error.

5.2.3 Member Function Documentation

5.2.3.1 `get_additional_information()`

```
std::string wr22::regex_explainer::explanation::Hint::get_additional_information ( ) const
```

Returns

the additional information for current error

5.2.3.2 `get_hint()`

```
std::string wr22::regex_explainer::explanation::Hint::get_hint ( ) const
```

Returns

the hint for current error

5.2.4 Member Data Documentation

5.2.4.1 `main_sentence`

```
constexpr const char* wr22::regex_explainer::explanation::Hint::main_sentence [static], [constexpr]
```

Initial value:

```
=  
    "Your pattern contains one or more errors, please see the explanation section above"
```

This is a sentence for any error indicating that there is an error in the regular expression.

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

- [include/wr22/regex_explainer/hints/hint_folder.hpp](#)
- [src/hints/hint_folder.cpp](#)

Chapter 6

File Documentation

6.1 `include/wr22/regex_explainer/explanation/explanation.hpp` File Reference

```
#include "wr22/regex_parser/regex/part.hpp"
#include "wr22/regex_parser/span/span.hpp"
#include "wr22/regex_explainer/explanation/explanation_folder.hpp"
#include "wr22/regex_explainer/explanation/sample_folder.hpp"
#include "wr22/regex_explainer/explanation/upgrade_sample.hpp"
#include "wr22/unicode/conversion.hpp"
#include <string_view>
```

Namespaces

- namespace [wr22](#)
- namespace [wr22::regex_explainer](#)
- namespace [wr22::regex_explainer::explanation](#)

Functions

- `std::vector< Explanation > wr22::regex_explainer::explanation::get_full_explanation (const SpannedPart &spanned_part, size_t depth=0)`

The function returns the vector processed information about each node of the syntax tree.

6.2 `explanation.hpp`

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

```
1 #pragma once
2
3 // wr22
4 #include "wr22/regex_parser/regex/part.hpp"
5 #include "wr22/regex_parser/span/span.hpp"
6 #include "wr22/regex_explainer/explanation/explanation_folder.hpp"
7 #include "wr22/regex_explainer/explanation/sample_folder.hpp"
8 #include "wr22/regex_explainer/explanation/upgrade_sample.hpp"
9 #include "wr22/unicode/conversion.hpp"
```

```

10
11 // STL
12 #include <string_view>
13
14 namespace wr22::regex_explainer::explanation {
15
16     using regex_parser::regex::SpannedPart;
17
18     std::vector<Explanation>
19     get_full_explanation(const SpannedPart &spanned_part, size_t depth = 0);
20
21 } // namespace wr22::regex_explainer::explanation

```

6.3 include/wr22/regex_explainer/explanation/explanation_folder.hpp

File Reference

```

#include "wr22/regex_explainer/explanation/explanation_folder.hpp"
#include "wr22/utils/adt.hpp"
#include <nlohmann/json_fwd.hpp>
#include <string>
#include <variant>
#include <nlohmann/json.hpp>

```

Classes

- class [wr22::regex_explainer::explanation::Explanation](#)
A class for holding an explanation tree element.

Namespaces

- namespace [wr22](#)
- namespace [wr22::regex_explainer](#)
- namespace [wr22::regex_explainer::explanation](#)

Typedefs

- typedef [wr22::utils::Adt< std::string, std::u32string, uint32_t > wr22::regex_explainer::explanation::type_of_sentence](#)

Functions

- void [wr22::regex_explainer::explanation::to_json](#) (nlohmann::json &j, const Explanation &explanation)

6.4 explanation_folder.hpp

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

```

1 #pragma once
2
3 // wr22
4 #include "wr22/regex_explainer/explanation/explanation_folder.hpp"
5 #include "wr22/utils/adt.hpp"
6
7 // STL
8 #include <nlohmann/json_fwd.hpp>
9 #include <string>
10
11 // variant
12 #include <variant>
13
14 // nlohmann
15 #include <nlohmann/json.hpp>
16
17 namespace wr22::regex_explainer::explanation {
18
19 typedef wr22::utils::Adt<std::string, std::u32string, uint32_t> type_of_sentence;
20
21
22 class Explanation {
23 public:
24     Explanation();
25
26     Explanation(type_of_sentence explanation_, size_t depth_, bool bold_ = false);
27
28     [[nodiscard]] type_of_sentence get_explanation() const;
29     [[nodiscard]] size_t get_depth() const;
30     [[nodiscard]] bool is_bold() const;
31
32 private:
33     type_of_sentence explanation;
34     size_t depth;
35     bool bold;
36 };
37
38 void to_json(nlohmann::json& j, const Explanation& explanation);
39
40 } // namespace wr22::regex_explainer::explanation

```

6.5 include/wr22/regex_explainer/explanation/sample_folder.hpp File Reference

```

#include "wr22/regex_explainer/explanation/explanation_folder.hpp"
#include "wr22/regex_parser/regex/part.hpp"
#include "wr22/regex_parser/span/span.hpp"
#include <string_view>

```

Namespaces

- namespace [wr22](#)
- namespace [wr22::regex_explainer](#)
- namespace [wr22::regex_explainer::explanation](#)

Functions

- std::string [wr22::regex_explainer::explanation::get_sample](#) (const Empty &vertex)
The function returns a sample for the AST node depending on its type.
- std::vector< std::string > [wr22::regex_explainer::explanation::get_sample](#) (const Literal &vertex)
The function describes which literal matched and what index it has in the ASCII table.

- `std::string wr22::regex_explainer::explanation::get_sample` (const Alternatives &vertex)
The function lists all alternatives that needed to be mached and describes each one in detail.
- `std::string wr22::regex_explainer::explanation::get_sample` (const Sequence &vertex)
The function lists all items of the sequence list and match there explanation one after another.
- `std::vector< std::string > wr22::regex_explainer::explanation::get_sample` (const Group &vertex)
he function says that it is group and describes it in detail.
- `std::string wr22::regex_explainer::explanation::get_sample` (const Optional &vertex)
Next three functions return the explanation of some quantifier.
- `std::string wr22::regex_explainer::explanation::get_sample` (const Plus &vertex)
- `std::string wr22::regex_explainer::explanation::get_sample` (const Star &vertex)
- `std::string wr22::regex_explainer::explanation::get_sample` (const Wildcard &vertex)
The function returns the explanation for some regex basic that is always the same and does not require any modifications depending on the situation.
- `std::vector< std::string > wr22::regex_explainer::explanation::get_sample` (const CharacterClass &vertex)
The function lists the symbols from the certain set that need to be matched.

6.6 sample_folder.hpp

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

```

1 #pragma once
2
3 // wr22
4 #include "wr22/regex_explainer/explanation/explanation_folder.hpp"
5 #include "wr22/regex_parser/regex/part.hpp"
6 #include "wr22/regex_parser/span/span.hpp"
7
8 // STL
9 #include <string_view>
10
11 namespace wr22::regex_explainer::explanation {
12
13     using regex_parser::regex::part::Empty;
14     using regex_parser::regex::part::Literal;
15     using regex_parser::regex::part::Alternatives;
16     using regex_parser::regex::part::Sequence;
17     using regex_parser::regex::part::Group;
18     using regex_parser::regex::part::Optional;
19     using regex_parser::regex::part::Plus;
20     using regex_parser::regex::part::Star;
21     using regex_parser::regex::part::Wildcard;
22     using regex_parser::regex::part::CharacterClass;
23
24     std::string get_sample(const Empty &vertex);
25
26     std::vector<std::string> get_sample(const Literal &vertex);
27
28     std::string get_sample(const Alternatives &vertex);
29
30     std::string get_sample(const Sequence &vertex);
31
32     std::vector<std::string> get_sample(const Group &vertex);
33
34     std::string get_sample(const Optional &vertex);
35
36     std::string get_sample(const Plus &vertex);
37
38     std::string get_sample(const Star &vertex);
39
40     std::string get_sample(const Wildcard &vertex);
41
42     std::vector<std::string> get_sample(const CharacterClass &vertex);
43
44 } // namespace wr22::regex_explainer::explanation

```


6.7 include/wr22/regex_explainer/explanation/upgrade_sample.hpp File Reference

```
#include "wr22/regex_explainer/explanation/sample_folder.hpp"
#include <string_view>
```

Namespaces

- namespace [wr22](#)
- namespace [wr22::regex_explainer](#)
- namespace [wr22::regex_explainer::explanation](#)

Functions

- `std::string wr22::regex_explainer::explanation::upgrade_sample` (const std::string &symbol, const std::string &sample)

Upgrades the default sample and makes it more specific.

6.8 upgrade_sample.hpp

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

```
1 #pragma once
2
3 #include "wr22/regex_explainer/explanation/sample_folder.hpp"
4
5 // STL
6 #include <string_view>
7
8 namespace wr22::regex_explainer::explanation {
9
10 std::string upgrade_sample(const std::string& symbol, const std::string& sample);
11
12 } // namespace wr22::regex_explainer::explanation
```

6.9 include/wr22/regex_explainer/hints/hint.hpp File Reference

```
#include "wr22/regex_explainer/hints/hint_folder.hpp"
#include "wr22/regex_parser/parser/errors.hpp"
#include "wr22/unicode/conversion.hpp"
#include <vector>
```

Namespaces

- namespace [wr22](#)
- namespace [wr22::regex_explainer](#)
- namespace [wr22::regex_explainer::hints](#)

Functions

- Hint [wr22::regex_explainer::hints::get_hint](#) (const UnexpectedEnd &error)
If any exception was thrown, then there is an error in the regular expression.
- Hint [wr22::regex_explainer::hints::get_hint](#) (const ExpectedEnd &error)
- Hint [wr22::regex_explainer::hints::get_hint](#) (const UnexpectedChar &error)
- Hint [wr22::regex_explainer::hints::get_hint](#) (const InvalidRange &error)

6.10 hint.hpp

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

```
1 #pragma once
2
3 // wr22
4 #include "wr22/regex_explainer/hints/hint_folder.hpp"
5 #include "wr22/regex_parser/parser/errors.hpp"
6 #include "wr22/unicode/conversion.hpp"
7
8 // vector
9 #include <vector>
10
11 namespace wr22::regex_explainer::hints {
12
13     using explanation::Hint;
14     using regex_parser::parser::errors::ParseError;
15     using regex_parser::parser::errors::UnexpectedEnd;
16     using regex_parser::parser::errors::ExpectedEnd;
17     using regex_parser::parser::errors::UnexpectedChar;
18     using regex_parser::parser::errors::InvalidRange;
19
20
21
22
23
24
25     Hint get_hint(const UnexpectedEnd &error);
26
27     Hint get_hint(const ExpectedEnd &error);
28
29     Hint get_hint(const UnexpectedChar &error);
30
31     Hint get_hint(const InvalidRange &error);
32
33 } // namespace wr22::regex_explainer::hints
34
35
36
```

6.11 include/wr22/regex_explainer/hints/hint_folder.hpp File Reference

```
#include <string>
#include <string_view>
#include <utility>
#include <nlohmann/json.hpp>
```

Classes

- class [wr22::regex_explainer::explanation::Hint](#)
A class for holding a hint.

Namespaces

- namespace [wr22](#)
- namespace [wr22::regex_explainer](#)
- namespace [wr22::regex_explainer::explanation](#)

Functions

- void [wr22::regex_explainer::explanation::to_json](#) (nlohmann::json &j, const Hint &hint)

6.12 hint_folder.hpp

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

```
1 #pragma once
2
3 // STL
4 #include <string>
5 #include <string_view>
6 #include <utility>
7
8 // nlohmann
9 #include <nlohmann/json.hpp>
10
11 namespace wr22::regex_explainer::explanation {
12
13     class Hint {
14     public:
15         Hint();
16
17         explicit Hint(std::string hint_, std::string additional_info_ = "");
18
19         [[nodiscard]] std::string get_hint() const;
20         [[nodiscard]] std::string get_additional_information() const;
21
22         static constexpr const char* main_sentence =
23             "Your pattern contains one or more errors, please see the explanation section above";
24
25     private:
26         std::string hint;
27         std::string additional_info;
28     };
29
30 void to_json(nlohmann::json& j, const Hint& hint);
31
32 } // namespace wr22::regex_explainer::explanation
```

6.13 src/explanation/explanation.cpp File Reference

```
#include "wr22/regex_explainer/explanation/explanation.hpp"
#include "wr22/unicode/conversion.hpp"
#include <optional>
#include <string_view>
#include <variant>
#include <vector>
#include <fmt/format.h>
```

Namespaces

- namespace [wr22](#)
- namespace [wr22::regex_explainer](#)
- namespace [wr22::regex_explainer::explanation](#)

Functions

- std::vector< Explanation > [wr22::regex_explainer::explanation::get_full_explanation](#) (const SpannedPart &spanned_part, size_t depth=0)

The function returns the vector processed information about each node of the syntax tree.

6.14 src/explanation/explanation_folder.cpp File Reference

```
#include <wr22/regex_explainer/explanation/explanation_folder.hpp>
#include <wr22/unicode/conversion.hpp>
#include <utility>
```

Namespaces

- namespace [wr22](#)
- namespace [wr22::regex_explainer](#)
- namespace [wr22::regex_explainer::explanation](#)

Functions

- void [wr22::regex_explainer::explanation::to_json](#) (nlohmann::json &j, const Explanation &explanation)

6.15 src/explanation/sample_folder.cpp File Reference

```
#include "wr22/regex_explainer/explanation/sample_folder.hpp"
#include <string_view>
```

Namespaces

- namespace [wr22](#)
- namespace [wr22::regex_explainer](#)
- namespace [wr22::regex_explainer::explanation](#)

Functions

- std::string [wr22::regex_explainer::explanation::get_sample](#) (const Empty &vertex)
The function returns a sample for the AST node depending on its type.
- std::vector< std::string > [wr22::regex_explainer::explanation::get_sample](#) (const Literal &vertex)
The function describes which literal matched and what index it has in the ASCII table.
- std::string [wr22::regex_explainer::explanation::get_sample](#) (const Alternatives &vertex)
The function lists all alternatives that needed to be mached and describes each one in detail.
- std::string [wr22::regex_explainer::explanation::get_sample](#) (const Sequence &vertex)
The function lists all items of the sequence list and match there explanation one after another.
- std::vector< std::string > [wr22::regex_explainer::explanation::get_sample](#) (const Group &vertex)
he function says that it is group and describes it in detail.
- std::string [wr22::regex_explainer::explanation::get_sample](#) (const Optional &vertex)
Next three functions return the explanation of some quantifier.
- std::string [wr22::regex_explainer::explanation::get_sample](#) (const Plus &vertex)
- std::string [wr22::regex_explainer::explanation::get_sample](#) (const Star &vertex)
- std::string [wr22::regex_explainer::explanation::get_sample](#) (const Wildcard &vertex)
The function returns the explanation for some regex basic that is always the same and does not require any modifications depending on the situation.
- std::vector< std::string > [wr22::regex_explainer::explanation::get_sample](#) (const CharacterClass &vertex)
The function lists the symbols from the certain set that need to be matched.

6.16 src/explanation/upgrade_sample.cpp File Reference

```
#include "wr22/regex_explainer/explanation/upgrade_sample.hpp"
```

Namespaces

- namespace [wr22](#)
- namespace [wr22::regex_explainer](#)
- namespace [wr22::regex_explainer::explanation](#)

Functions

- `std::string wr22::regex_explainer::explanation::upgrade_sample` (const std::string &symbol, const std::string &sample)

Upgrades the default sample and makes it more specific.

6.17 src/hints/hint.cpp File Reference

```
#include "wr22/regex_explainer/hints/hint.hpp"  
#include <string>
```

Namespaces

- namespace [wr22](#)
- namespace [wr22::regex_explainer](#)
- namespace [wr22::regex_explainer::hints](#)

Functions

- Hint [wr22::regex_explainer::hints::get_hint](#) (const ParseError &error)
- Hint [wr22::regex_explainer::hints::get_hint](#) (const UnexpectedEnd &error)
If any exception was thrown, then there is an error in the regular expression.
- Hint [wr22::regex_explainer::hints::get_hint](#) (const ExpectedEnd &error)
- Hint [wr22::regex_explainer::hints::get_hint](#) (const UnexpectedChar &error)
- Hint [wr22::regex_explainer::hints::get_hint](#) (const InvalidRange &error)

6.18 src/hints/hint_folder.cpp File Reference

```
#include <wr22/regex_explainer/hints/hint_folder.hpp>
```

Namespaces

- namespace [wr22](#)
- namespace [wr22::regex_explainer](#)
- namespace [wr22::regex_explainer::explanation](#)

Functions

- void [wr22::regex_explainer::explanation::to_json](#) (nlohmann::json &j, const Hint &hint)

Index

Explanation
 wr22::regex_explainer::explanation::Explanation, 13

get_additional_information
 wr22::regex_explainer::explanation::Hint, 16

get_depth
 wr22::regex_explainer::explanation::Explanation, 14

get_explanation
 wr22::regex_explainer::explanation::Explanation, 14

get_full_explanation
 wr22::regex_explainer::explanation, 8

get_hint
 wr22::regex_explainer::explanation::Hint, 16
 wr22::regex_explainer::hints, 12

get_sample
 wr22::regex_explainer::explanation, 9, 10

Hint
 wr22::regex_explainer::explanation::Hint, 15

include/wr22/regex_explainer/explanation/explanation.hpp, 17

include/wr22/regex_explainer/explanation/explanation_folder.hpp, 18, 19

include/wr22/regex_explainer/explanation/sample_folder.hpp, 19, 20

include/wr22/regex_explainer/explanation/upgrade_sample.hpp, 21

include/wr22/regex_explainer/hints/hint.hpp, 21, 22

include/wr22/regex_explainer/hints/hint_folder.hpp, 22, 23

is_bold
 wr22::regex_explainer::explanation::Explanation, 14

main_sentence
 wr22::regex_explainer::explanation::Hint, 16

src/explanation/explanation.cpp, 23

src/explanation/explanation_folder.cpp, 24

src/explanation/sample_folder.cpp, 24

src/explanation/upgrade_sample.cpp, 25

src/hints/hint.cpp, 25

src/hints/hint_folder.cpp, 25

to_json
 wr22::regex_explainer::explanation, 11

type_of_sentence
 wr22::regex_explainer::explanation, 8

upgrade_sample
 wr22::regex_explainer::explanation, 11

wr22, 7

wr22::regex_explainer, 7

wr22::regex_explainer::explanation, 7

 get_full_explanation, 8

 get_sample, 9, 10

 to_json, 11

 type_of_sentence, 8

 upgrade_sample, 11

wr22::regex_explainer::explanation::Explanation, 13

 Explanation, 13

 get_depth, 14

 get_explanation, 14

 is_bold, 14

wr22::regex_explainer::explanation::Hint, 15

 get_additional_information, 16

 get_hint, 16

 Hint, 15

 main_sentence, 16

wr22::regex_explainer::hints, 11

 get_hint, 12