

`string_view`

0.1

Generated by Doxygen 1.9.1

| | |
|--|-----------|
| 1 Main Page | 1 |
| 2 Class Index | 3 |
| 2.1 Class List | 3 |
| 3 Class Documentation | 5 |
| 3.1 string_view Class Reference | 5 |
| 3.1.1 Detailed Description | 6 |
| 3.1.2 Constructor & Destructor Documentation | 6 |
| 3.1.2.1 string_view() [1/4] | 6 |
| 3.1.2.2 string_view() [2/4] | 6 |
| 3.1.2.3 string_view() [3/4] | 6 |
| 3.1.2.4 string_view() [4/4] | 7 |
| 3.1.3 Member Function Documentation | 7 |
| 3.1.3.1 empty() | 7 |
| 3.1.3.2 ends_with() [1/3] | 7 |
| 3.1.3.3 ends_with() [2/3] | 8 |
| 3.1.3.4 ends_with() [3/3] | 8 |
| 3.1.3.5 remove_prefix() | 8 |
| 3.1.3.6 remove_suffix() | 9 |
| 3.1.3.7 size() | 9 |
| 3.1.3.8 starts_with() [1/3] | 9 |
| 3.1.3.9 starts_with() [2/3] | 10 |
| 3.1.3.10 starts_with() [3/3] | 10 |
| 3.1.3.11 substr() | 10 |
| 3.1.4 Friends And Related Function Documentation | 11 |
| 3.1.4.1 operator<< | 11 |
| 3.1.4.2 operator== | 11 |
| Index | 13 |

Chapter 1

Main Page

This is simple C++ `std::string_view` implementation as part of practical task "Enterprise C++" in Sphere.

Chapter 2

Class Index

2.1 Class List

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

| | |
|---------------------------------------|---|
| string_view | 5 |
|---------------------------------------|---|

Chapter 3

Class Documentation

3.1 string_view Class Reference

Public Member Functions

- [string_view](#) ()
Default constructor.
- [string_view](#) (const [string_view](#) &other)
Copy constructor.
- [string_view](#) (const char *raw)
Constructor from raw pointer.
- [string_view](#) (const char *raw, size_t count)
Constructor from raw pointer with count.
- size_t [size](#) () const
Get size.
- bool [empty](#) () const
Check if view is empty.
- bool [starts_with](#) (const [string_view](#) &view) const
Check if view starts with another view.
- bool [starts_with](#) (char c) const
Check if view starts with given char.
- bool [starts_with](#) (const char *c) const
Check if view starts with raw string.
- bool [ends_with](#) (const [string_view](#) &view) const
Check if view ends with another view.
- bool [ends_with](#) (char c) const
Check if view ends with given char.
- bool [ends_with](#) (const char *c) const
Check if view ends with raw string.
- [string_view substr](#) (size_t pos, size_t count) const
Get a substring of view.
- const char & [operator\[\]](#) (size_t pos) const
- void [remove_suffix](#) (size_t n)
Remove suffix.
- void [remove_prefix](#) (size_t n)
Remove prefix.

Friends

- `std::ostream & operator<< (std::ostream &os, const string_view &s)`
Output overloading.
- `bool operator== (const string_view &view, const string_view &other)`
operator== overloading

3.1.1 Detailed Description

implementation

3.1.2 Constructor & Destructor Documentation

3.1.2.1 `string_view()` [1/4]

```
string_view::string_view ( )
```

Default constructor.

Constructs empty [string_view](#) with data = nullptr and length = 0.

3.1.2.2 `string_view()` [2/4]

```
string_view::string_view (
    const string\_view & other )
```

Copy constructor.

Parameters

| | |
|--------------|---|
| <i>other</i> | Other string_view instance. |
|--------------|---|

Constructs [string_view](#) of same content as other with data = other.data and length = other.length.

3.1.2.3 `string_view()` [3/4]

```
string_view::string_view (
    const char * raw )
```

Constructor from raw pointer.

Parameters

| | |
|------------|--------------------------------|
| <i>raw</i> | Raw pointer string first item. |
|------------|--------------------------------|

Constructs a [string_view](#) instance of null-terminated char array pointed to by raw.

3.1.2.4 string_view() [4/4]

```
string_view::string_view (
    const char * raw,
    size_t count )
```

Constructor from raw pointer with count.

Parameters

| | |
|--------------|--|
| <i>raw</i> | Raw pointer string first item. |
| <i>count</i> | Number of elements to include in view. |

Constructs view of count elements.

3.1.3 Member Function Documentation

3.1.3.1 empty()

```
bool string_view::empty ( ) const
```

Check if view is empty.

Returns

False if not empty, true otherwise.

3.1.3.2 ends_with() [1/3]

```
bool string_view::ends_with (
    char c ) const
```

Check if view ends with given char.

Parameters

| | |
|----------|----------------|
| <i>c</i> | Char to check. |
|----------|----------------|

Checks if this [string_view](#) instance ends with given char.

Returns

True if this view ends with given char, false otherwise.

3.1.3.3 ends_with() [2/3]

```
bool string_view::ends_with (
    const char * c ) const
```

Check if view ends with raw string.

Parameters

| | |
|----------|-------------------------|
| <i>c</i> | Raw char array pointer. |
|----------|-------------------------|

Checks if this [string_view](#) instance ends with raw string.

Returns

True if this view ends with given raw string, false otherwise.

3.1.3.4 ends_with() [3/3]

```
bool string_view::ends_with (
    const string\_view & view ) const
```

Check if view ends with another view.

Parameters

| | |
|-------------|----------------|
| <i>view</i> | View to check. |
|-------------|----------------|

Checks if this [string_view](#) instance ends with another view.

Returns

True if this view ends with another, false otherwise.

3.1.3.5 remove_prefix()

```
void string_view::remove_prefix (
    size_t n )
```

Remove prefix.

Parameters

| | |
|----------|---|
| <i>n</i> | — size of prefix to remove. @detailed Move the beginnig of view <i>n</i> chars forward. |
|----------|---|

3.1.3.6 remove_suffix()

```
void string_view::remove_suffix (
    size_t n )
```

Remove suffix.

Parameters

| | |
|----------|---|
| <i>n</i> | — size of suffix to remove. @detailed Move the end of view <i>n</i> chars back. |
|----------|---|

3.1.3.7 size()

```
size_t string_view::size ( ) const
```

Get size.

Returns

Size of view in chars.

3.1.3.8 starts_with() [1/3]

```
bool string_view::starts_with (
    char c ) const
```

Check if view starts with given char.

Parameters

| | |
|----------|----------------|
| <i>c</i> | Char to check. |
|----------|----------------|

Checks if this [string_view](#) instance starts with given char .

Returns

True if this view starts with given char, false otherwise.

3.1.3.9 starts_with() [2/3]

```
bool string_view::starts_with (
    const char * c ) const
```

Check if view starts with raw string.

Parameters

| | |
|----------|-------------------------|
| <i>c</i> | Raw char array pointer. |
|----------|-------------------------|

Checks if this [string_view](#) instance starts with raw string.

Returns

True if this view starts with given raw string, false otherwise.

3.1.3.10 starts_with() [3/3]

```
bool string_view::starts_with (
    const string\_view & view ) const
```

Check if view starts with another view.

Parameters

| | |
|-------------|----------------|
| <i>view</i> | View to check. |
|-------------|----------------|

Checks if this [string_view](#) instance starts with another view.

Returns

True if this view starts with another, false otherwise.

3.1.3.11 substr()

```
string\_view string_view::substr (
    size_t pos,
    size_t count ) const
```

Get a substring of view.

Parameters

| | |
|--------------|---------------------------|
| <i>pos</i> | Position to start with. |
| <i>count</i> | Number of chars required. |

Returns

Sub-view.

Returns as substringview beginning from postion pos of count elements.

3.1.4 Friends And Related Function Documentation

3.1.4.1 operator<<

```
std::ostream& operator<< (  
    std::ostream & os,  
    const string_view & s ) [friend]
```

Output overloading.

Parameters

| | |
|-----------|-----------------------|
| <i>os</i> | Output stream. |
| <i>s</i> | string_view instance. |

Returns

Output stream.

3.1.4.2 operator==

```
bool operator== (  
    const string_view & view,  
    const string_view & other ) [friend]
```

operator== overloading

Lexicographically compares two string_view instances.

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

- string_view.h
- string_view.cpp

Index

- empty
 - string_view, [7](#)
- ends_with
 - string_view, [7](#), [8](#)
- operator<<
 - string_view, [11](#)
- operator==
 - string_view, [11](#)
- remove_prefix
 - string_view, [8](#)
- remove_suffix
 - string_view, [9](#)
- size
 - string_view, [9](#)
- starts_with
 - string_view, [9](#), [10](#)
- string_view, [5](#)
 - empty, [7](#)
 - ends_with, [7](#), [8](#)
 - operator<<, [11](#)
 - operator==, [11](#)
 - remove_prefix, [8](#)
 - remove_suffix, [9](#)
 - size, [9](#)
 - starts_with, [9](#), [10](#)
 - string_view, [6](#), [7](#)
 - substr, [10](#)
- substr
 - string_view, [10](#)