

Introduction

Let's take a look at what we'll cover in this section.

Since C++11 and move semantics, passing strings has become much faster. You can end up with many temporary string copies. In C++17 you get a new type called `string_view`. It allows you to create a constant, non-owning view of a contiguous character sequence. You can manipulate that view and pass it around without the need to copy the referenced data.

Nevertheless, the feature comes at some cost: you need to be careful not to end up with “dangling” views, and usually such views might not be null-terminated.

In this section you'll learn:

- What is `string_view`?
 - Why might it speed up your code?
 - What are the risks involved with using `string_view` objects?
 - What is the reference lifetime extension and what does it mean for `string_view`?
 - How can you use `string_view` to make your API more generic?
-

Let's get started with the basics.