

How can I build a URL query string in the Windows Runtime?

 devblogs.microsoft.com/oldnewthing/20220630-00

June 30, 2022



Raymond Chen

Last time, we looked at a Windows Runtime helper object for parsing URL query strings. It also has a helper object for building URL query strings, but for some reason the two aren't in the same place.

In C#, you can use a `FormUrlEncodedContent`. In the Windows Runtime, a similar function is provided by the `HttpFormUrlEncodedContent` class. The oddity is that the query parser is in the `Windows.Foundation` namespace, but the query builder is in the `Windows.Web.Http` namespace. My guess is that query parsing is something protocol handlers need to do, whereas query building is something you typically do only when dealing with HTTP servers.

Anyway, here's the `HttpFormUrlEncodedContent` class:

```
std::map<hstring, hstring> content;
content.emplace(L"v", L"dQw4w9WgXcQ");
content.emplace(L"t", L"43s");
auto encoder = HttpFormUrlEncodedContent(content);
auto encoded = encoder.ToString();
```

You can optimize this by using the `initializer_list` constructor for `std::map`:

```
std::map<hstring, hstring> content{
    { L"v", L"dQw4w9WgXcQ" },
    { L"t", L"43s" },
};
auto encoder = HttpFormUrlEncodedContent(content);
auto encoded = encoder.ToString();
```

But wait, we can do better: We don't need a `std::map` at all. The `HttpFormUrlEncodedContent` constructor takes an `IIterator<KeyValuePair<hstring, hstring>>`, and C++/WinRT allows you to pass, among other things, a `std::initializer_list<std::pair<hstring, hstring>>`. Those are easy to make on the fly:

```
auto encoder = HttpFormUrlEncodedContent({
    { L"v", L"dQw4w9WgXcQ" },
    { L"t", L"43s" },
});
auto encoded = encoder.ToString();
```

And now you can make it a one-liner:

```
auto encoded = HttpFormUrlEncodedContent({
    { L"v", L"dQw4w9WgXcQ" },
    { L"t", L"43s" },
}).ToString();
```

Raymond Chen

Follow

