How can I parse URL query string in the Windows Runtime?

devblogs.microsoft.com/oldnewthing/20220629-00

June 29, 2022

Raymond Chen

In C#, you can use the `HttpUtility.ParseQueryString` method to parse URL query strings. For C++/WinRT, you can use the Windows Runtime `Windows.Foundation.WwwFormUrlDecoder` class.

```
Uri uri{ L"https://www.youtube.com/watch?v=%64Qw4w9WgXcQ&t=43s" };
auto decoder = uri.QueryParsed();

// or you can parse a query string directly
WwwFormUrlDecoder decoder(L"v=%64Qw4w9WgXcQ&t=43s");
```

Once it's parsed, you can pull out the pieces.

```
auto id = decoder.GetFirstValueByName(L"v");
```

The `WwwFormUrlDecoder` is a bit difficult to work with, however, because the `GetFirstValueByName` method throws an exception if the key is not present at all. Since you are usually parsing query strings from potentially-untrusted sources, you can't be certain that the value you want is present.

What you can do is convert the parsed query into some other format that is easier to work with, say, a `std::multimap`:

```
auto to_multimap(
    winrt::Windows::Foundation::WwwFormUrlDecoder const& decoder)
{
    std::multimap<winrt::hstring, winrt::hstring> parsed;
    for (auto&& entry : decoder) {
        parsed.emplace(entry.Name(), entry.Value());
    }
    return parsed;
}
```

Now you can pull out the values from the query in a way you are more comfortable with. And you can even use the usual multimap methods to extract multiple-valued keys, if that's something you care about.

```
auto parsed = to_multimap(uri.QueryParsed());

// Does a "v=" key exist?
if (!parsed.contains(L"v"))
{
    error();
}

// Process all the "v=" keys.
auto values = parsed.equal_range(L"v");
for (auto it = values.first; it != values.second; ++it)
{
    process_value(it->second);
}

// Process the first "v=" key, if any.
auto it = parsed.lower_bound(L"v");
if (it != parsed.end() && it->first == L"v") {
    process_value(it->second);
}
```

Raymond Chen

Follow