記事
Toshihiko Minamoto · 2021年7月12日 3m read

REST の高度な URL マッピング

現在では、基本的な REST API を Caché
に実装する方法は一般的であり、これに関する優れたドキュメントが「REST in Caché」に掲載されています。

時々次のような疑問が投げ掛けられることがあります。

REST url のパラメータをオプションにするにはどうすればよいですか？

簡単に言えば、Caché では次のような URL をマッピングする URL マップを作成することができます。

```
http://server/API/object///old
```

この書き方は奇妙に見えるかもしれませんが、実際に有効な URL です。 詳細は、RFC3986 セクション 3.3
をご覧ください。

%CSP.Rest が提供しているフレームワークでは、実際に上記の URL
に一致するマップを作成することができます。 上記の参照先のドキュメントには、URL
マップは正規表現に変換されているという説明のみが補足として示されています。

この力を利用するには、正規表現を理解する必要があります。 Caché
のコンテキストで正規表現の基礎を説明したドキュメントを Michael が執筆しています。こちらをご覧ください。
正規表現に慣れていない場合は、時間を取って一読することを強くお勧めします。

%CSP.Rest では、マップを実際に正規表現にコンパイルするのは UrlMap ブロックです。
たとえば、Url="/class/:namespace/:classname" のような URL マップ（Docserver
の例から使用）は次のようにコンパイルされます。

```
/class/([^/]+)/([^/]+)
```

Michael の記事を思い出すと、この RegEx は /class/ で始まる URL に一致させてから、その次の / で区切られた
URL の 2 つの部分を 2 つの一位グループに投入していることがわかります。 これらは後でパラメータとして
GetClass に渡されます。

クラスの UrlMap の Url パラメータに正規表現を直接渡すことができます。
このようにすると、単純化の制限を回避することができます。 また、高度な URL
マップを作成することもできます。

```
XData UrlMap [ XMLNamespace = "http://www.intersystems.com/urlmap" ]
{
<Routes>
<Route Url="//API/object/([^/]*)/([^/]*)/old" Method="GET" Call="GetClass"/>
</Routes>
}
```

ここでは、単に *1 つ以上*の修飾子 + を *0 以上*の * に置き換えました。

正規表現の導入とともに、手元にあるこの力の意味が理解できたと思います。
ただし、次のことを忘れてはいけません。

??????????????????
--??????????????????

正規表現は非常に強力なツールであるため、ほかのソリューションでうまく解決できることにでも実装したくなりますが、 次に正規表現を作成するときは、そのことを忘れないようにしてください！

それでは、コーディングをお楽しみください！

#マッピング #REST API #Caché

ソースURL:
https://jp.community.intersystems.com/post/rest-%E3%81%AE%E9%AB%98%E5%BA%A6%E3%81%AA-url-%E3%83%9E%E3%83%83%E3%83%94%E3%83%B3%E3%82%B0