Reflection - ASP.Net MVC Core

In ASP.Net Core, server calls can be given as follows:

- localhost:12345/**Home**/**Index**
- localhost:12345/**Northwind**/**Get**/***666***
- localhost:12345/**Northwind**/**GetOrders**?***employeeId=1&customerId=ALFKI***

With this „Route“ a method in a class is called like this:

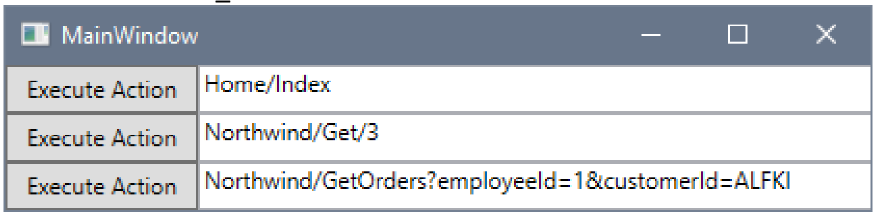
- Example 1: Method **Index**() of class **Home**Controller
- Example 2: Method **Get**(***666***) of class **Northwind**Controller
- Example 3: Method **GetOrders**(***1,"ALFKI"***) of class **Northwind**Controller

The rule therefore is:
Controllername*/*Method*/*Parameter or ***Controllername*/*Method*?*nameA=valueA&nameB=valueB***

Program a single method that calls the appropriate method from such a string:

```
private void btnExecuteAction_Click(string url)
{
  string response = ""; //split url here and call appropriate method
  MessageBox.Show(response);
}
```

For easier usage you can create three button clicks, but all of them should call the method btnExecuteAction_Click from above.



Home/Index
Northwind/Get/3
Northwind/GetOrders?employeeId=1&customerId=ALFKI

For simplicity, use the following classes and methods – they are just there to be called using reflection:

```
public class HomeController
{
  public string Index() =>
                $"{GetType().FullName}::{MethodBase.GetCurrentMethod().Name}";
}

public class NorthwindController
{
  public string Index() =>
                $"{GetType().FullName}::{MethodBase.GetCurrentMethod().Name}";
  public string Get(int id) =>
                $"{GetType().FullName}::{MethodBase.GetCurrentMethod().Name}({id})";
  public string GetOrders(int employeeId, string customerId) =>
   $"{GetType().FullName}::{MethodBase.GetCurrentMethod().Name}({employeeId},{customerId})";
}
```

First only program the first two variants and then try to program the third variant as an extension (also possible with ASP.Net MVC Core).
As a relief, you can simply assume that there are only int and string parameters.
Procedure:

- Find method in controller
- Read the parameter list of the method, important are name and type
- Split the string after **?** into name-value pairs (i.e. split with **&** and **=**)
- Loop parameter list of the method and check, if there are parameters in the query list with the same name
- If yes - depending on the type in method parameter list value cast / parse to int or string