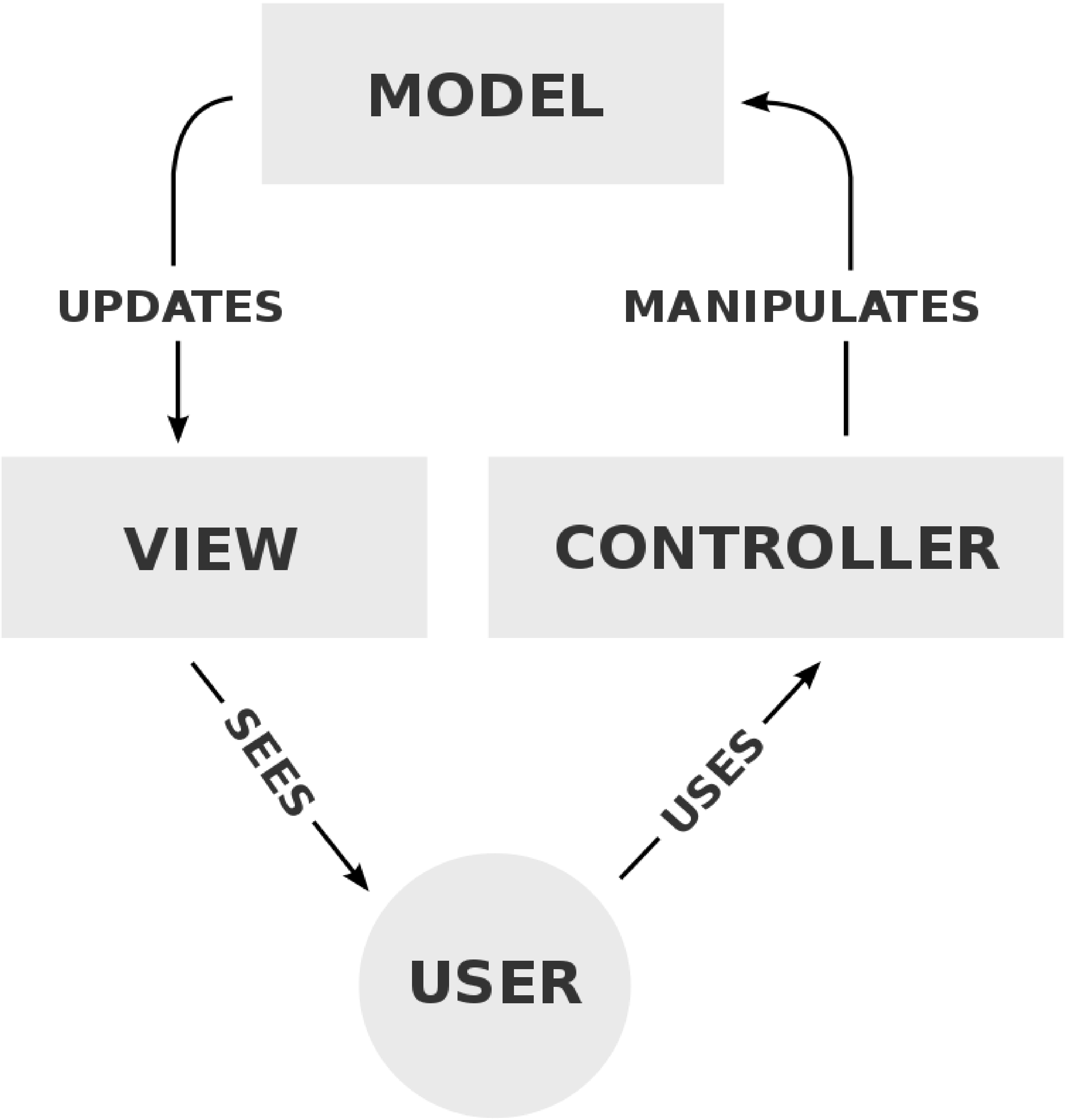
Controller

MVC pattern

- ជាsoftware design pattern ដែលវាធ្វើការបំបែក application ជា ៣ផ្នែកគឺ model, view និង controllerដើម្បីបង្កើតភាពងាយស្រួលដល់developerដើម្បីmaintenance application។


MODEL
UPDATES
MANIPULATES
VIEW
CONTROLLER
SEES
USES
USER

MVC pattern

- Model(is a data and business logic) ជាការបង្ហាញទំរង់នៃទិន្ន័យនិង business logic.វាជាអ្នកគ្រប់គ្រងទិន្ន័យរបស់កម្មវិធីដែលមានតួនាទីជាអ្នកទទួលនិងរក្សារទុកទិន្ន័យក្នុងដាតាបេស។
- View(is a User Interface) ជាអ្វីៗដែលអ្នកទស្សនាមើលឃើញដែលវាបង្ហាញទិន្ន័យ ដោយប្រើប្រាស់នូវ modelទៅកាន់អ្នកទស្សននិងអោយគេធ្វើការកែប្រែទិន្ន័យបាន។
- Controller (request handler)ជាអ្នកដោះស្រាយនូវសំណើររបស់អ្នកប្រើប្រាស់(user)។ជាទូទៅអ្នកប្រើប្រាស់ធ្វើអន្តកម្មជាមួយ Viewតាមរយះការលើកជាសំណើរ URL ត្រឹមត្រូវ ដែលសំណើរទាំងត្រូវបានដោះស្រាយដោយ Controller។បន្ទាប់មក controllerនិងបង្ហាញ នូវចំលើយតមតាមរ័យះ data model ទៅជាview សមរម្យមួយ។

Controller

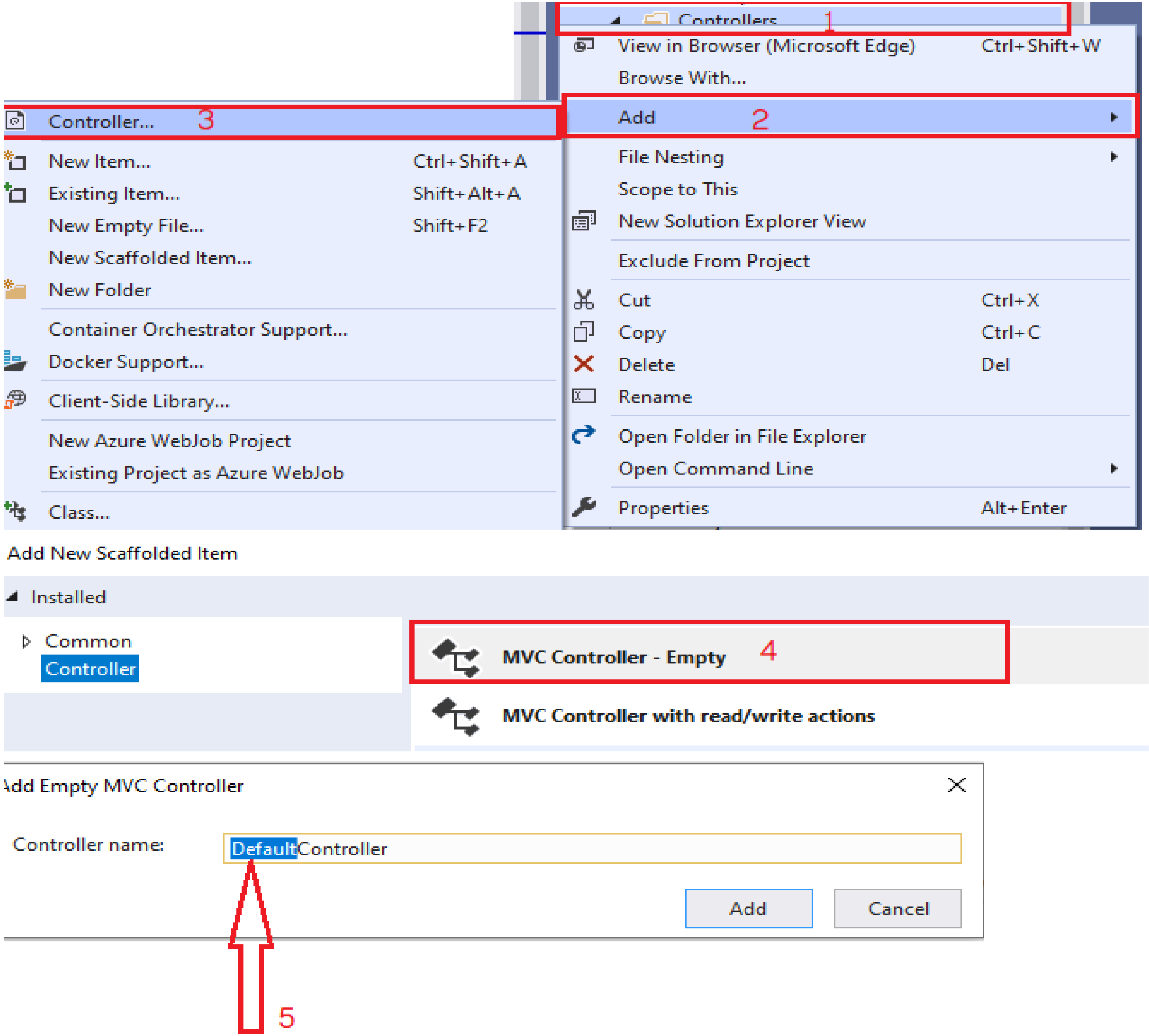
- Controller ជាបណ្តុំ classes ជាច្រើនដែលចាំទទួលដោះស្រាយនូវសំណើរពី userនិង និងឆ្លើយនូវតមនូវសំណើររបស់user។នៅពេលដែលuser វាយនូវសំណើរ(request) នៅលើ address bar របស់ browser ពេលនោះ route engine របស់ asp.net mvc និងធ្វើការផ្គុរផ្គងនូវសំណើរជាមួយ controller´s actions methodជាក់លាក់ណាមួយ។

តួនាទីរបស់ controller

- គ្រប់គ្រង user’s request(handle the user’s request)
- បង្កើត model (create model)
- និងបញ្ជូនចំលើយតបទៅ user វិញ(send the response)

ការបង្កើត controller

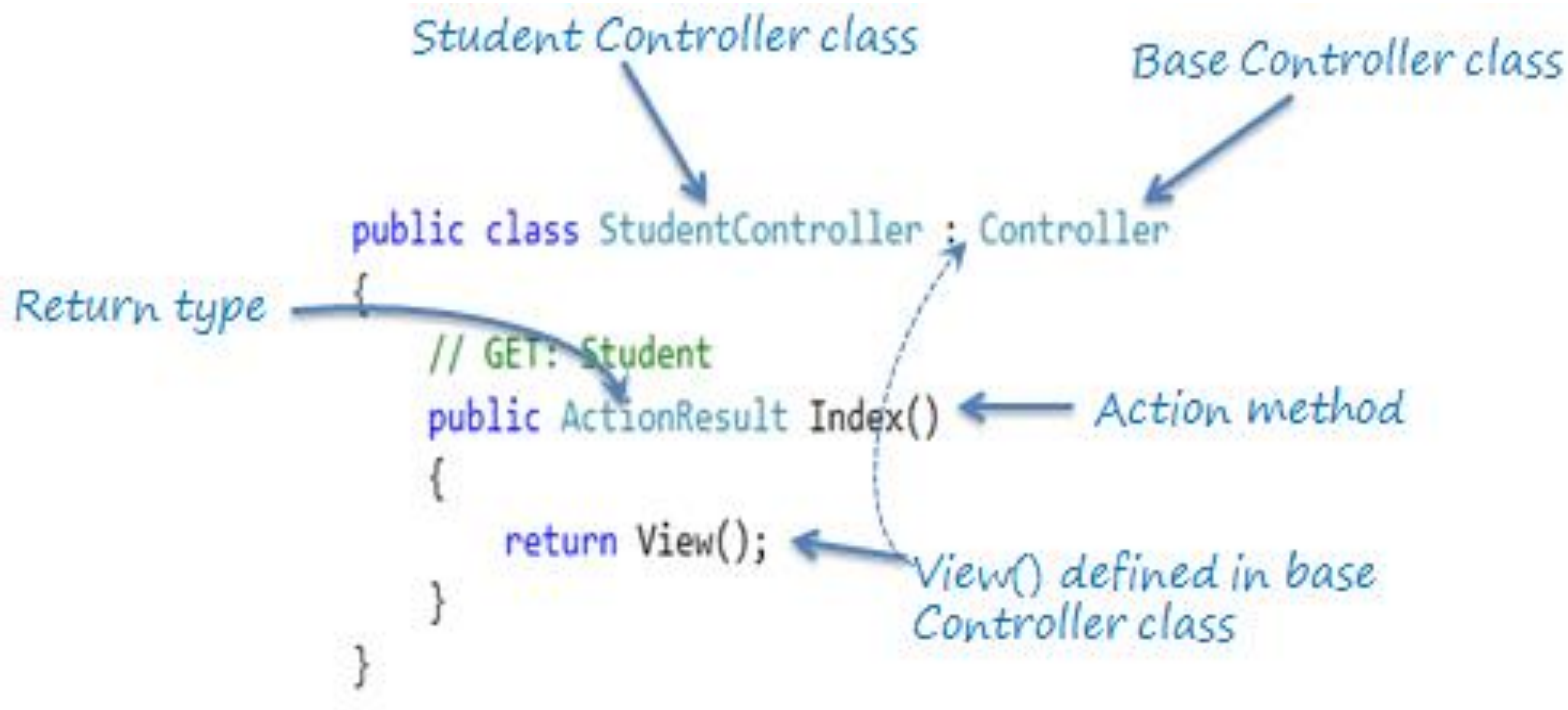
- ចុចម៉ៅស្ដាំលើថត controllers
- ចុចពាក្យថា add → controller
- ជ្រើសរើស controller template → MVC empty → Add
- ដាក់ឈ្មោះcontroller(ឈ្មោះ controller ត្រូវបញ្ចប់ដោយ controller)
- រួចហើយចុចពាក្យ→ Add



Controller name: DefaultController

Action method

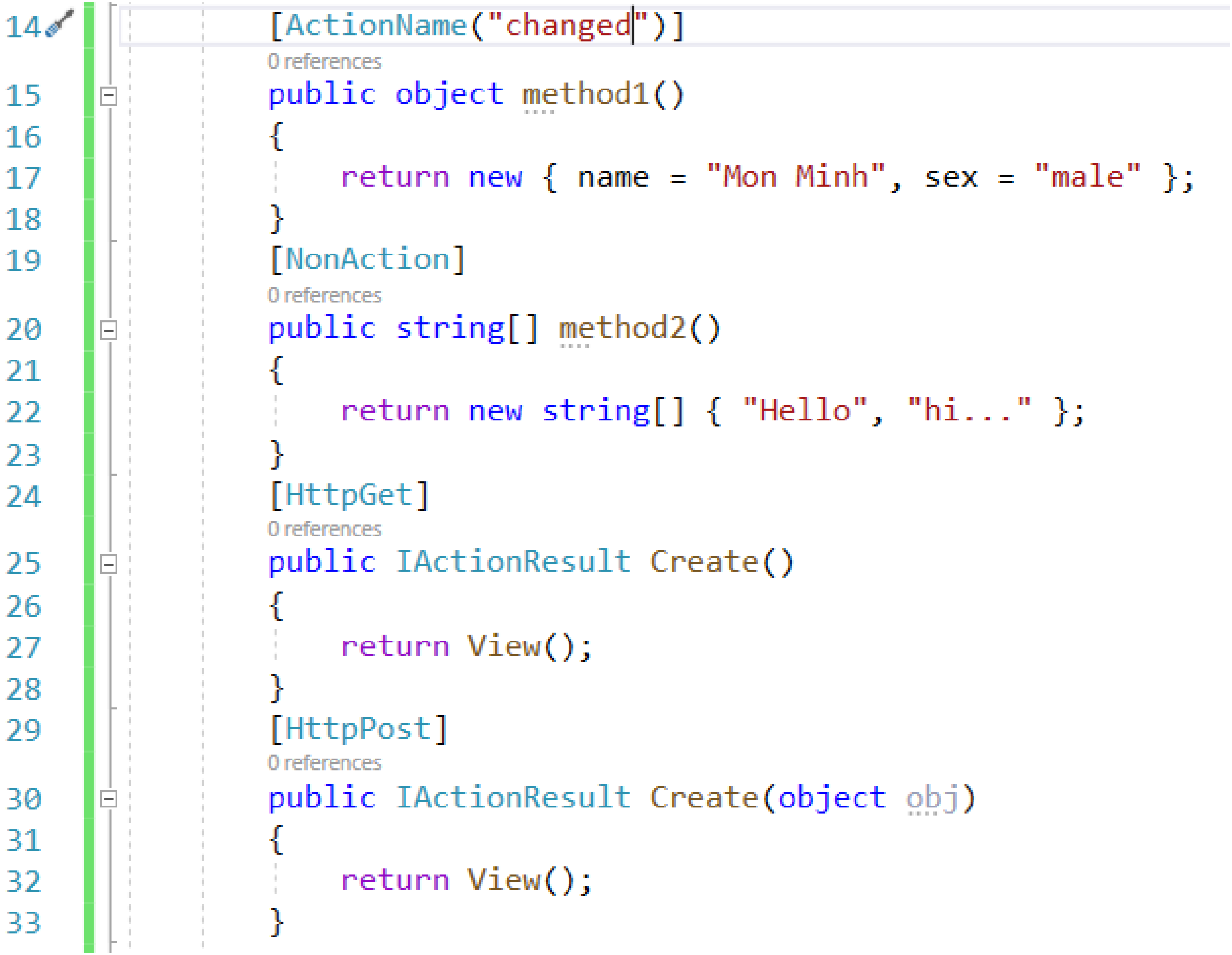
- គ្រប់ public methodទាំងអស់ដែលគេសរសេរនៅក្នុង controller class គេហៅថា Action Method។នៅពេលដែលយើងបង្កើត action method ត្រូវតែអនុវត្តន៍តាមរបៀបដូខចាងក្រោម៖
  - គ្រប់ actions method ទាំងអស់ត្រូវតែ public
  - វាមិនអាច overloadedទេ
  - វាមិនអាចជា static method
  - វាមិនអាចមានref ឬ out parameters
  - ត្រូវតែមាន return type



```
public class StudentController : Controller
{
    // GET: Student
    public ActionResult Index()
    {
        return View();
    }
}
```

Action Selector

- ជា attribute មួយដែលគេប្រើប្រាស់សម្រាប់កំណត់លក្ខណះរបស់action method របស់ controller។
- គេចែកជាបីគឺ៖
  - ActionName
    - ប្រើដម្បើប្តូរឈ្មោះនូវ action method
  - NoAction
    - ប្រើសម្រាប់កំណត់ action method ជា private ឬ method ធម្មតា
  - ActionVerb
    - ប្រើសប្រាប់កំណត់ប្រភេទនៃ http request(GET ឬ POST request)



```
        [ActionName("changed")]
        0 references
        public object method1()
        {
            return new { name = "Mon Minh", sex = "male" };
        }
        [NonAction]
        0 references
        public string[] method2()
        {
            return new string[] { "Hello", "hi..." };
        }
        [HttpGet]
        0 references
        public IActionResult Create()
        {
            return View();
        }
        [HttpPost]
        0 references
        public IActionResult Create(object obj)
        {
            return View();
        }
```

ActionResult

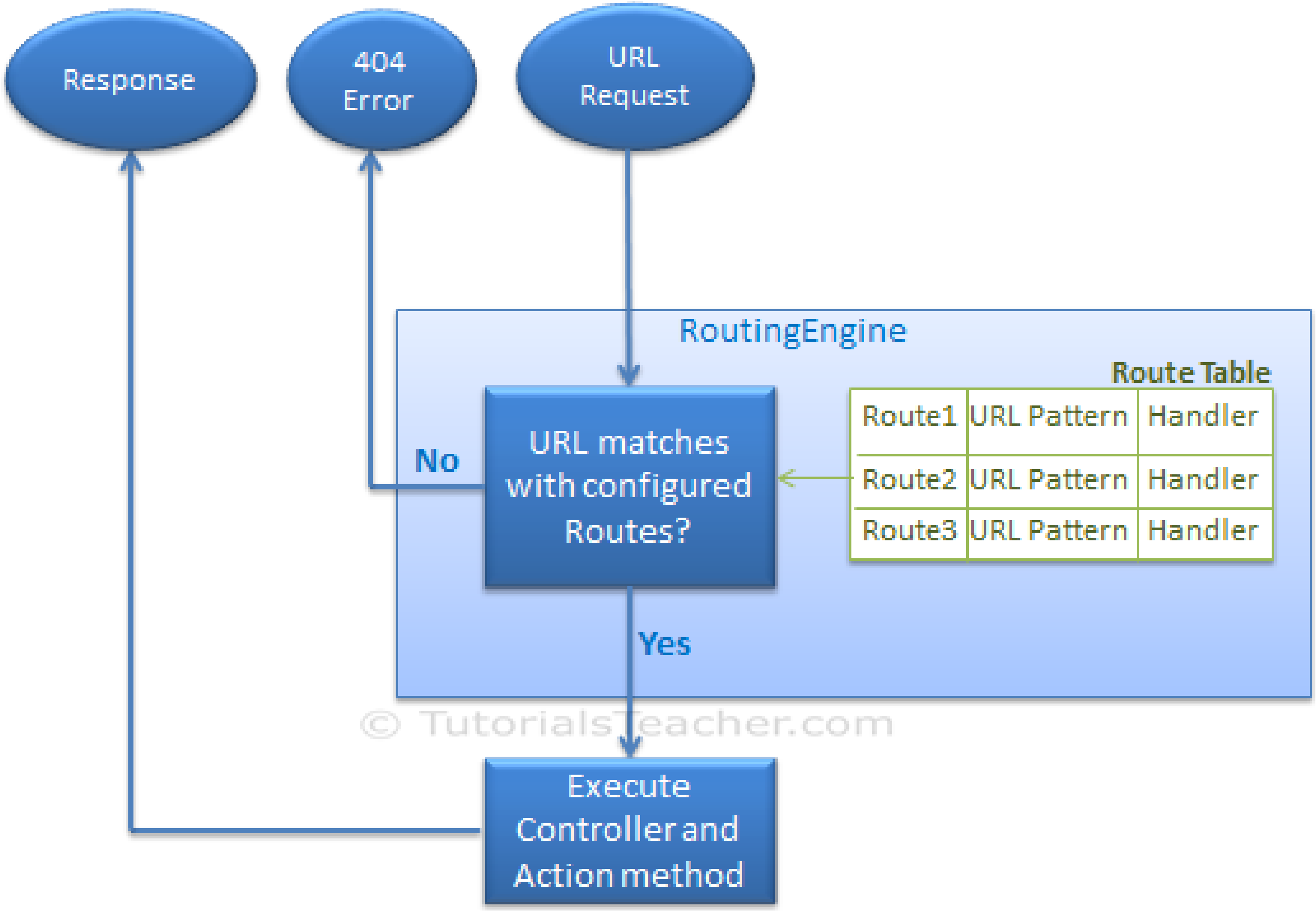
- វាជាលទ្ធផលឬ return type របស់ action method ដែលត្រូវផ្ដល់ទៅអោយ usersនៅពេលដែល action method របស់controller ណាមួយត្រូវតំណើរការ។

ActionResult	Helper Method	Description
ViewResult	View	បង្ហាញ view ជា web page.
PartialViewResult	PartialView	បង្ហាញជា បំណែកតូចនៃ web page(partial view)
RedirectResult	Redirect	បោះតំលៃនូវការទៅទីតាំងណាមួយ
RedirectToRouteResult	RedirectToRoute	ប្រើសម្រាប់ទៅតាំងណាម
ContentResult	Content	បោះតំលៃអក្សរធម្មតា(plain text)
JsonResult	Json	បោះតំលៃជា Json
EmptyResult	(None)	បោះតំលៃទទេ

Route

Route?

- វាជាតំណើរការនៃការផ្គុតផ្គងនូវ incoming requestជាមួយ controller’s action method ជាក់លាក់ណាមួយ។
- Routes និមួយៗមានដូចជា ឈ្មោះ, URL pattern(template), defaultនិង constrain


Response
404 Error
URL Request
RoutingEngine
URL matches with configured Routes?
No
Yes
Execute Controller and Action method

Route Table

Route1	URL Pattern	Handler
Route2	URL Pattern	Handler
Route3	URL Pattern	Handler

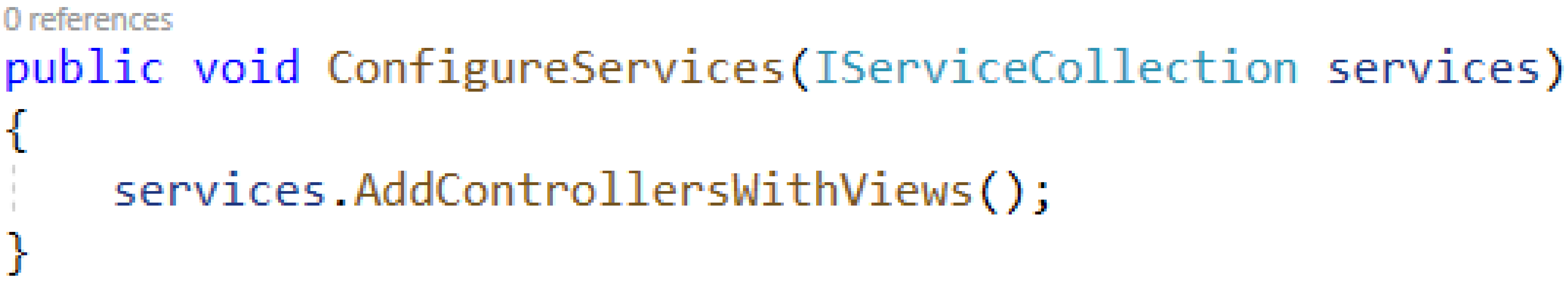
Route?

- មានវិធីសាស្ត្រពីរបីក្នុងបង្កើត Route
    - Conventional-base routing
        - ជាការបង្កើតroute នៅក្នុង startup class នៃ startup file
    - និង ការប្រើប្រាស់ attribute routing
        - ជាការបង្កើត route ដោយប្រើប្រាស់route attribute នៅក្នុង controller’s action method

Conventional Config

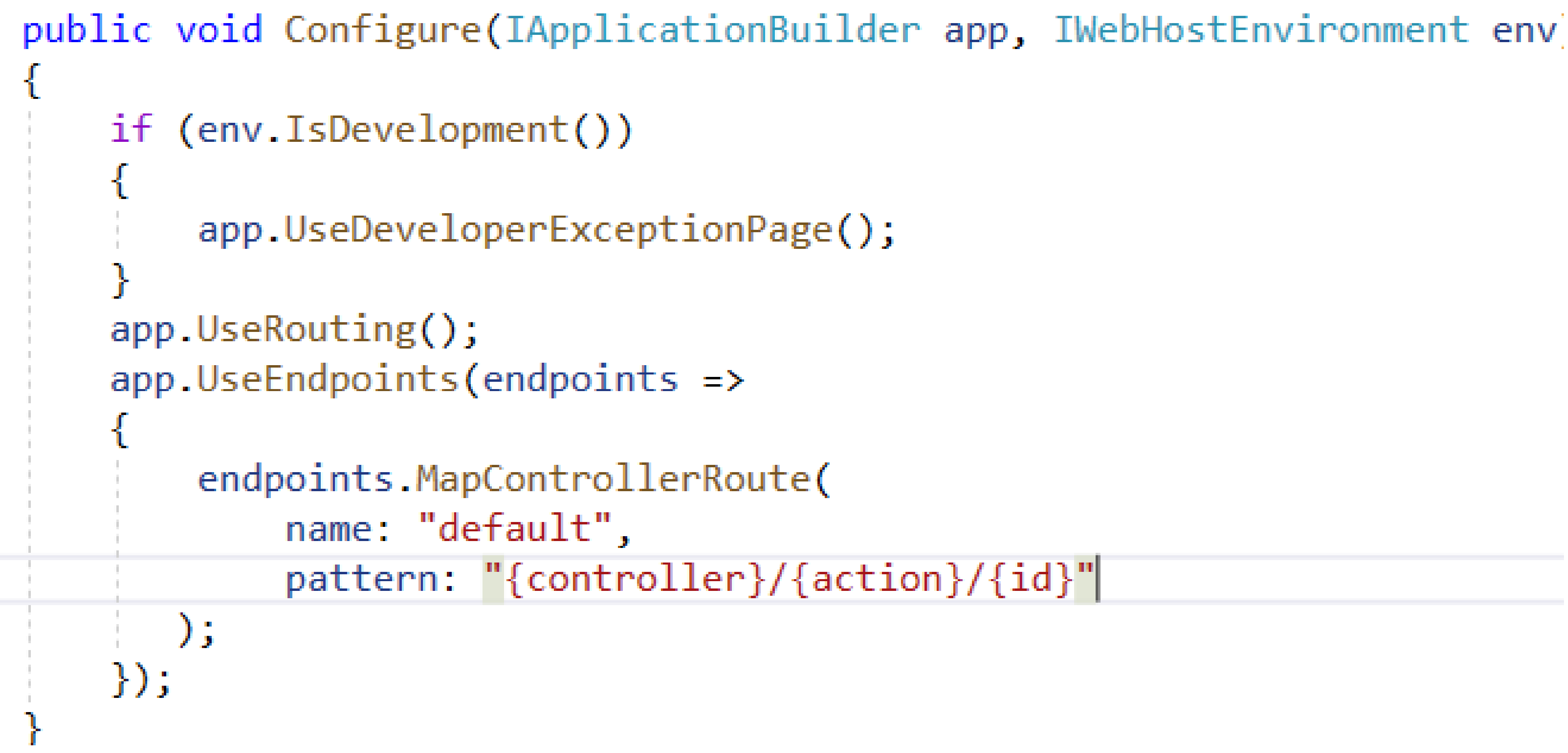
- ត្រូវបានបង្កើតនៅក្នុង configure method របស់ startup class
- Routing ត្រូវបានគ្រប់គ្រងដោយ router middlewareដែល asp.net mvc បន្ថែម routing middlewareទៅក្នុង middleware pipeline នៅពេលដែលយើងប្រើនូវ app.userRouting( ) និង app.useEndpoind( ) extension methods។

Adding Service



```
0 references
public void ConfigureServices(IServiceCollection services)
{
    services.AddControllersWithViews();
}
```

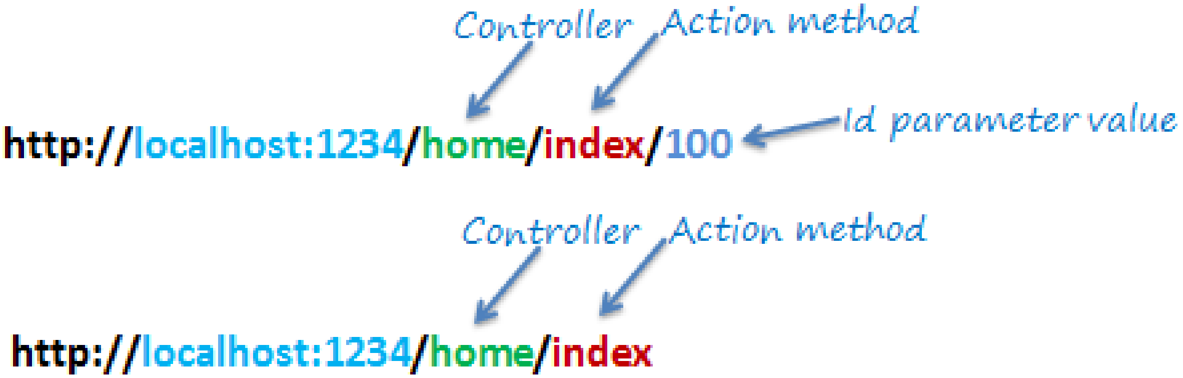
Anatomy of Route

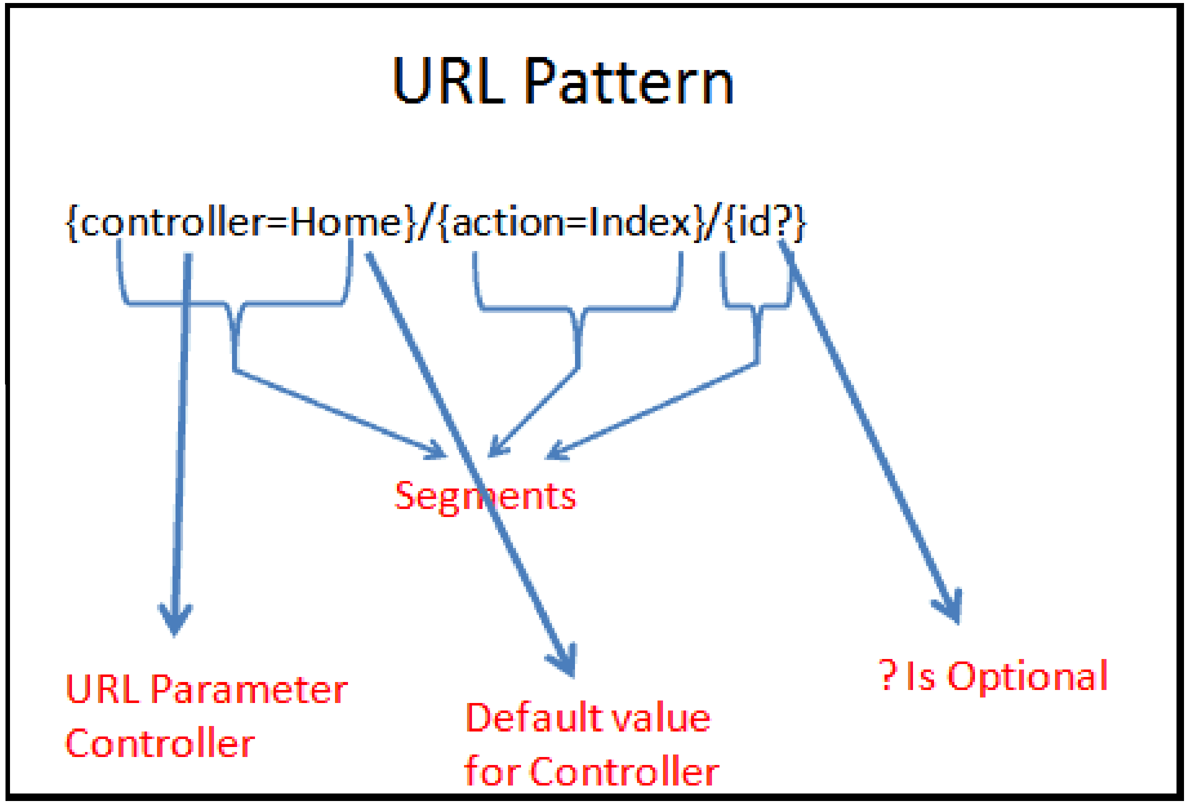


```
public void Configure(IApplicationBuilder app, IWebHostEnvironment env
{
    if (env.IsDevelopment())
    {
        app.UseDeveloperExceptionPage();
    }
    app.UseRouting();
    app.UseEndpoints(endpoints =>
    {
        endpoints.MapControllerRoute(
            name: "default",
            pattern: "{controller}/{action}/{id}"
        );
    });
}
```

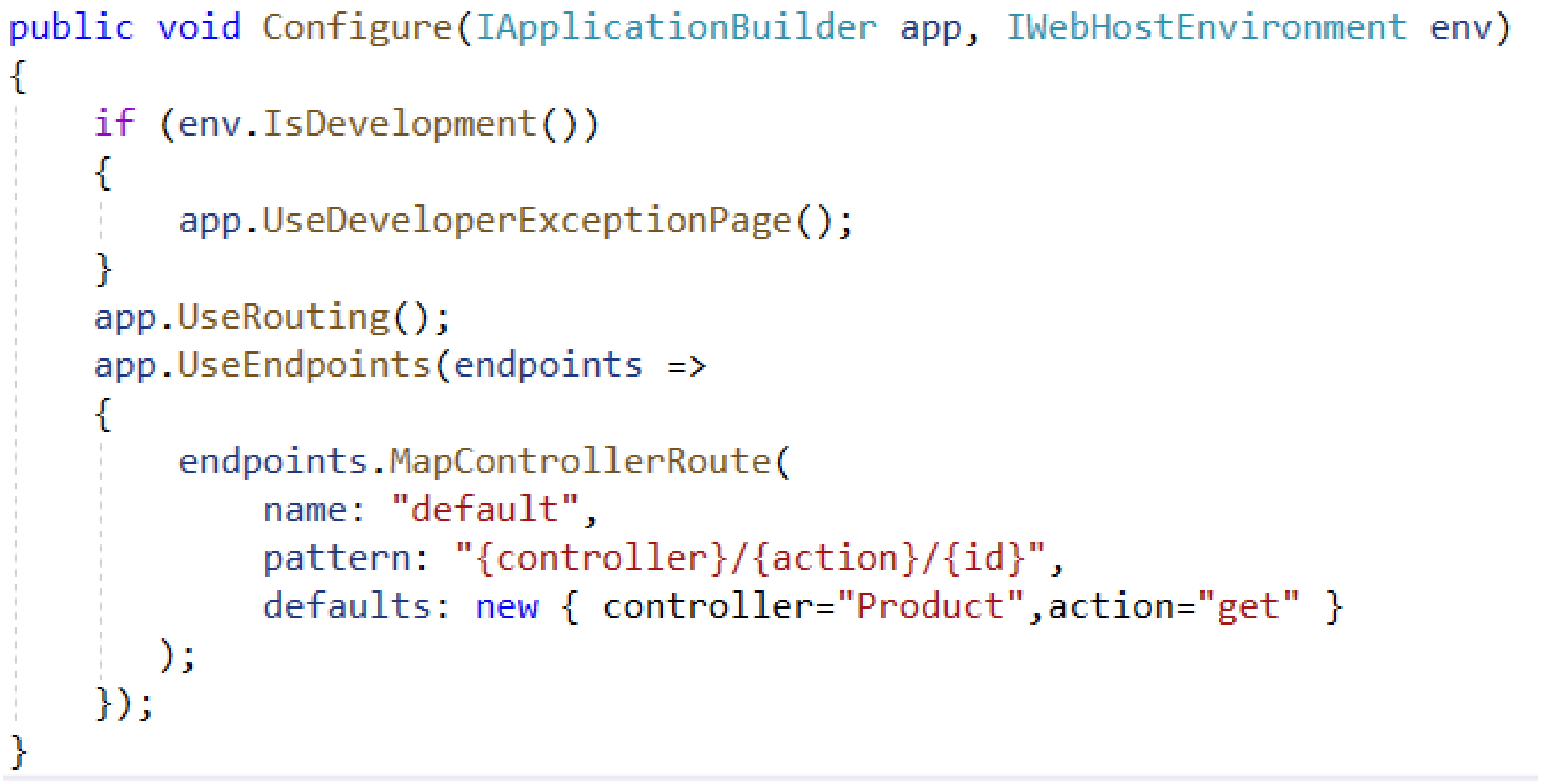
URL Patterns

- URL pattern ជាបំណែកតូចនៃ user’s request ដែលខណ្ឌគ្នាដោយសញ្ញា (/)


Controller
Action method
http://localhost:1234/home/index/100
Id parameter value
Controller
Action method
http://localhost:1234/home/index



Route Defaults



```
public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
{
    if (env.IsDevelopment())
    {
        app.UseDeveloperExceptionPage();
    }
    app.UseRouting();
    app.UseEndpoints(endpoints =>
    {
        endpoints.MapControllerRoute(
            name: "default",
            pattern: "{controller}/{action}/{id}",
            defaults: new { controller="Product",action="get" }
       );
    });
}
```

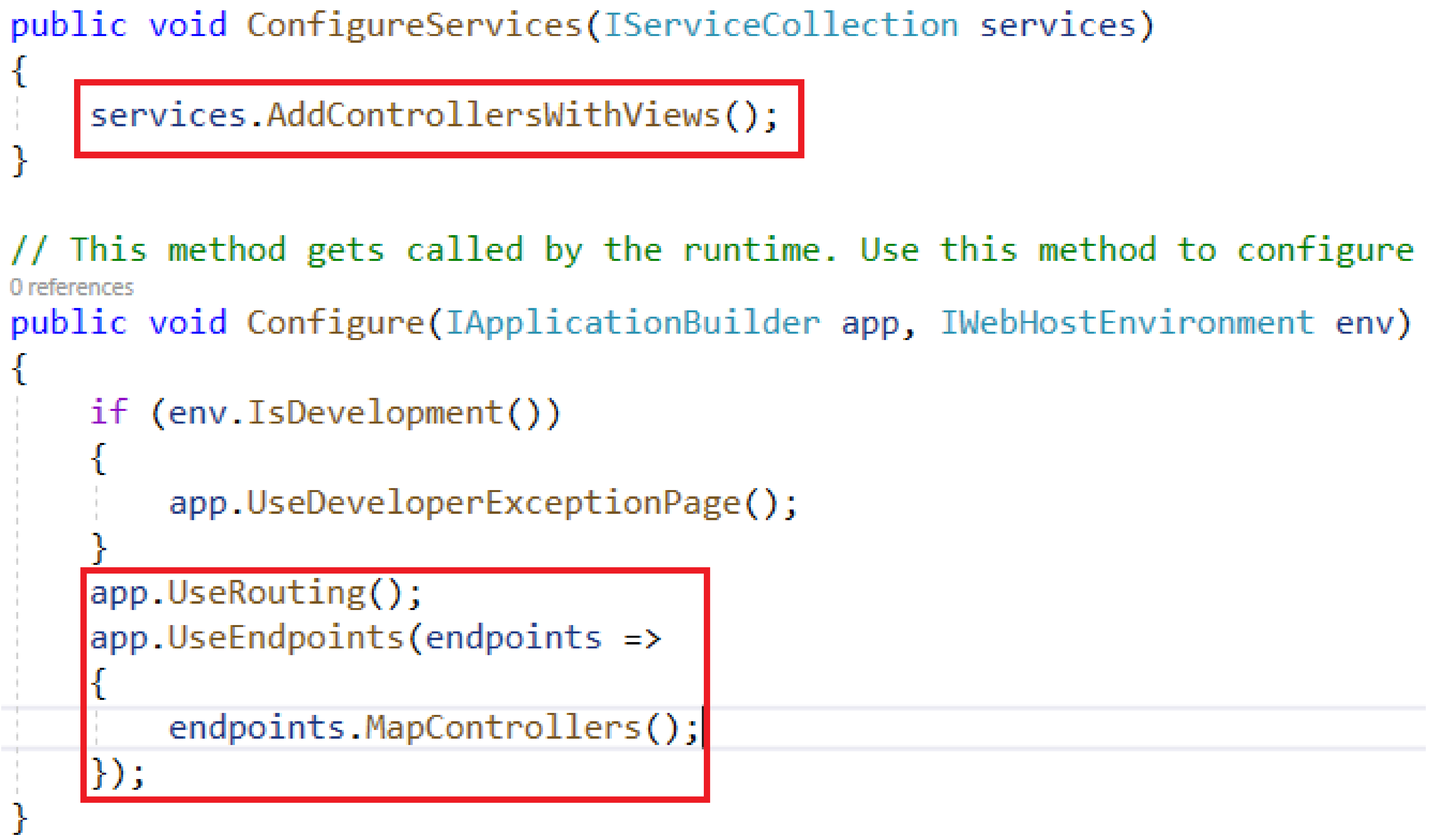
Route matching

URL	Controller	Action	Id
http://localhost/home	HomeController	Index	null
http://localhost/home/index/123	HomeController	Index	123
http://localhost/home/about	HomeController	About	null
http://localhost/home/contact	HomeController	Contact	null
http://localhost/student	StudentController	Index	null
http://localhost/student/edit/123	StudentController	Edit	123

Attribute’s route config

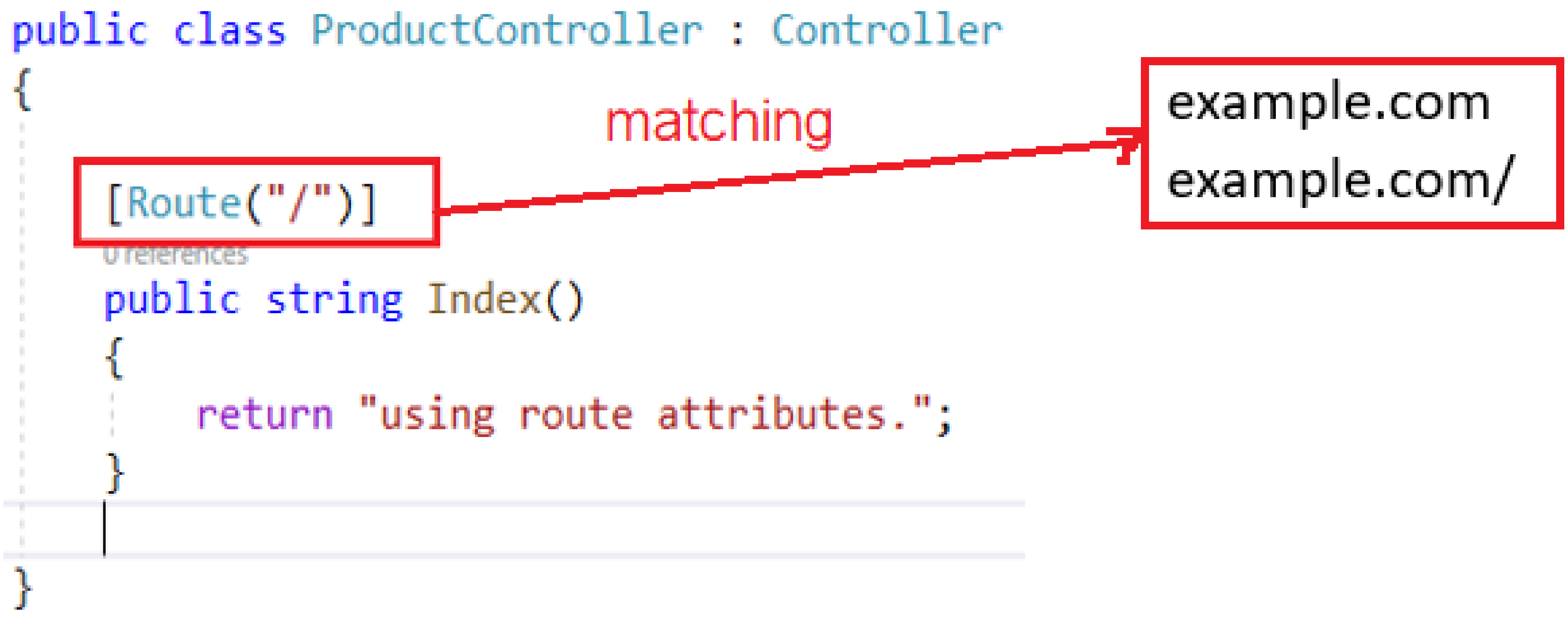
- គេអាចកំណត់ route attribute បានពីររបៀប៖
    - កំណត់ដោយផ្ទាល់ចំពោះ action methodណាមួយ
    - កំណត់ជាមួយ controllerណាមួយ

Attribute’s route config



```
public void ConfigureServices(IServiceCollection services)
{
    services.AddControllersWithViews();
}

// This method gets called by the runtime. Use this method to configure
0 references
public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
{
    if (env.IsDevelopment())
    {
        app.UseDeveloperExceptionPage();
    }
    app.UseRouting();
    app.UseEndpoints(endpoints =>
    {
        endpoints.MapControllers();
    });
}
```

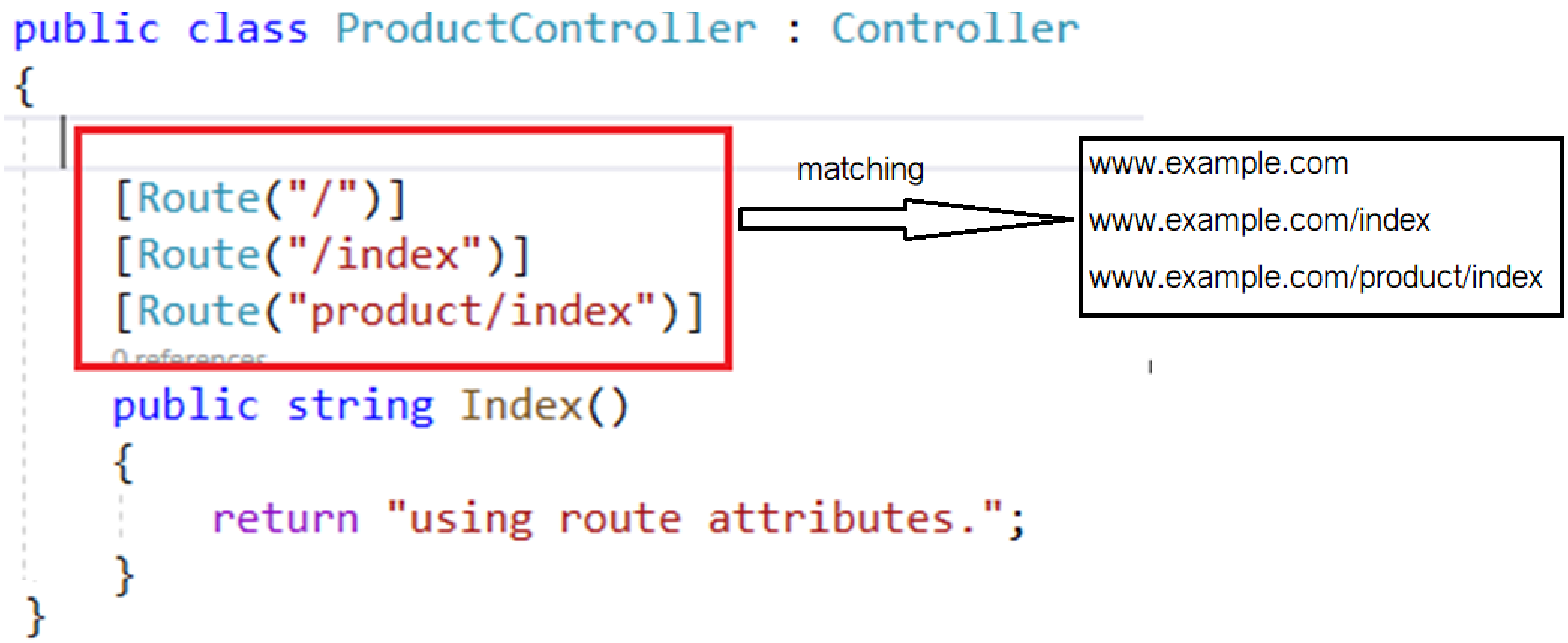
កំណត់ជាមួយaction method



```
public class ProductController : Controller
{

    [Route("/")]
    0 references
    public string Index()
    {
        return "using route attributes.";
    }

}
```

example.com
example.com/

កំណត់ជាមួយaction method(multiple route)



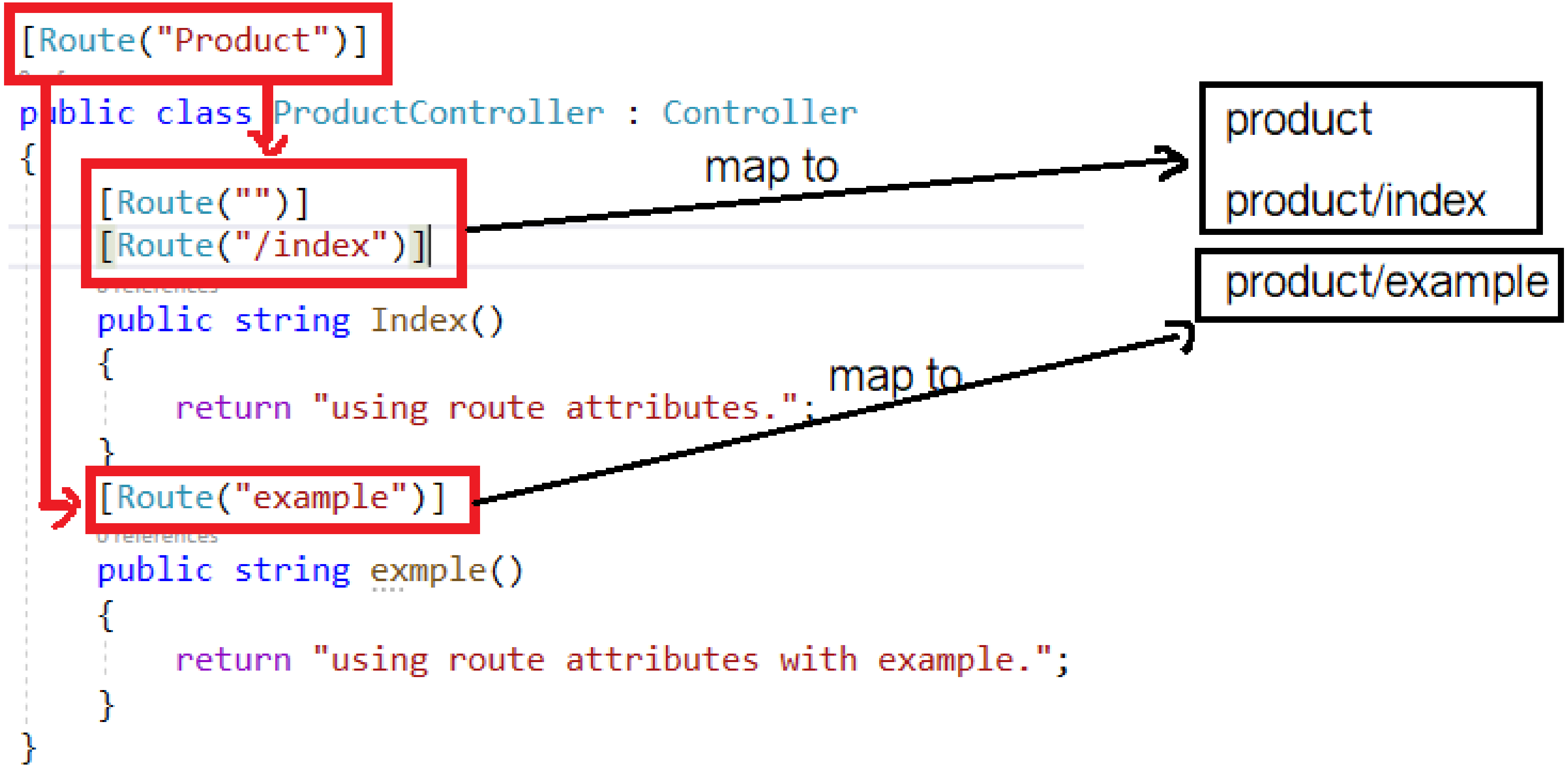
```
public class ProductController : Controller
{

    [Route("/")]
    [Route("/index")]
    [Route("product/index")]
    0 references
    public string Index()
    {
        return "using route attributes.";
    }
}
```

www.example.com

www.example.com/index

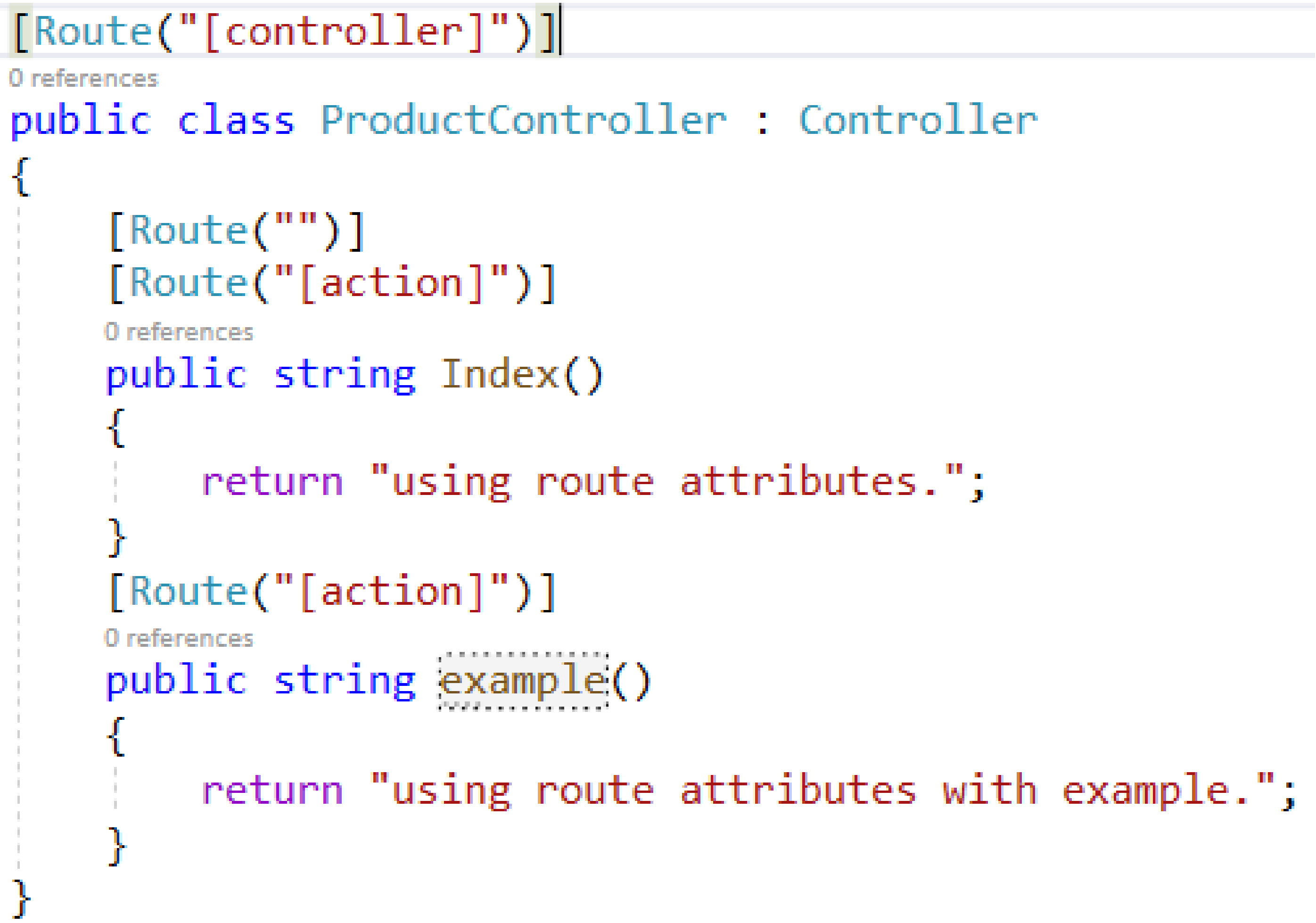
www.example.com/product/index

កំណត់ជាមួយ controller



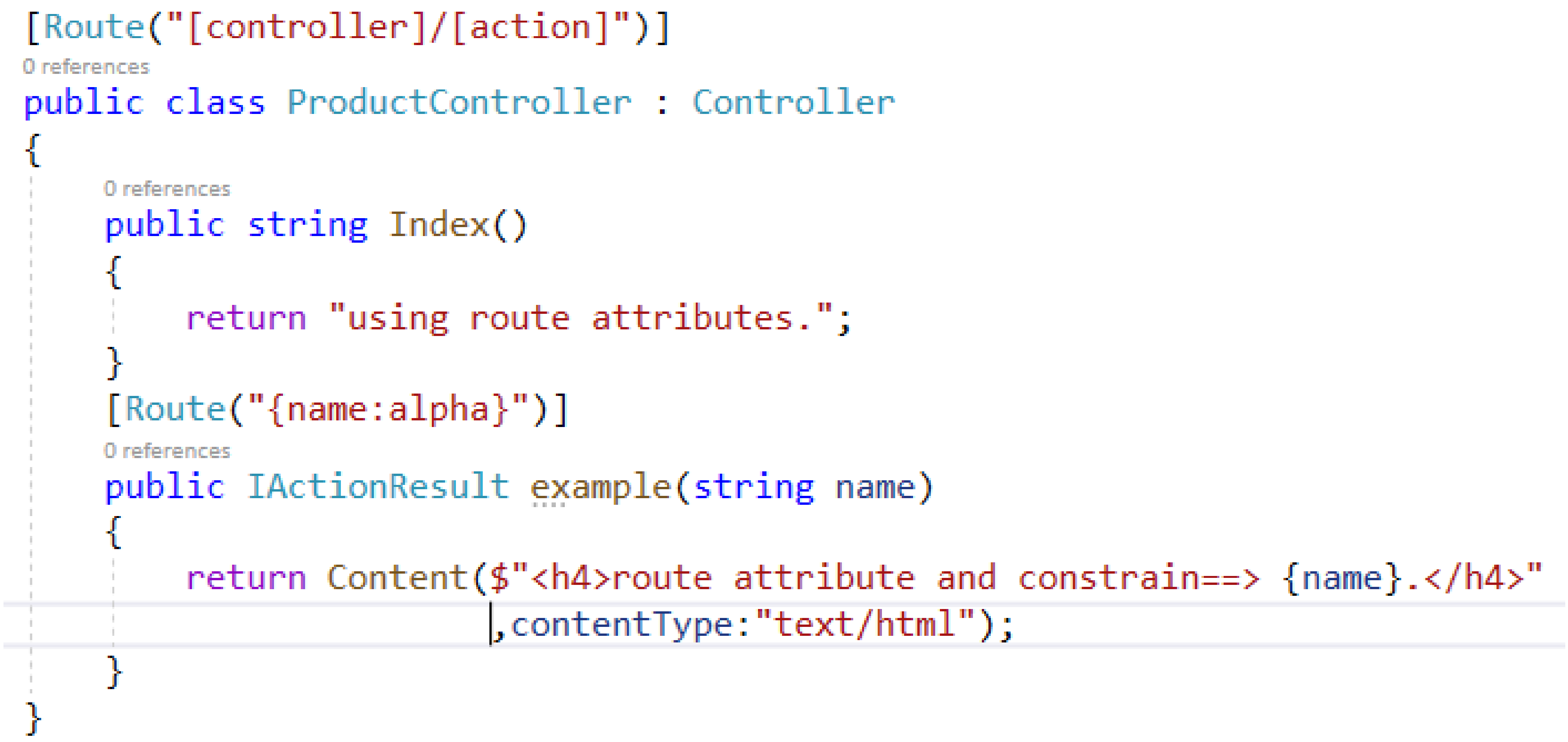
```
[Route("Product")]
public class ProductController : Controller
{
    [Route("")]
    [Route("/index")]
    public string Index()
    {
        return "using route attributes.";
    }
    [Route("example")]
    public string exmple()
    {
        return "using route attributes with example.";
    }
}
```

Token Replacement



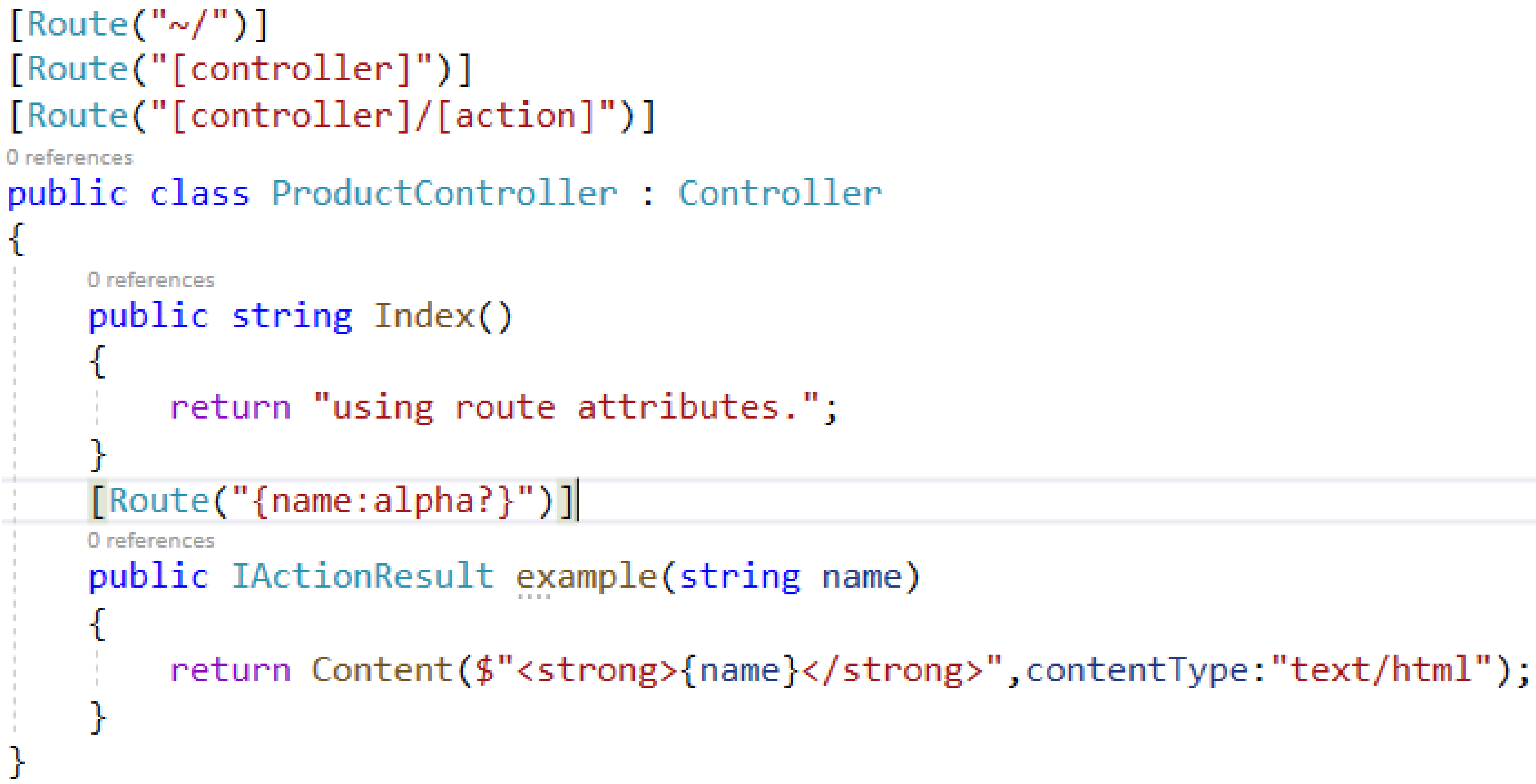
```
[Route("[controller]")]
0 references
public class ProductController : Controller
{
    [Route("")]
    [Route("[action]")]
    0 references
    public string Index()
    {
        return "using route attributes.";
    }
    [Route("[action]")]
    0 references
    public string example()
    {
        return "using route attributes with example.";
    }
}
```

Token Replacement



```
[Route("[controller]/[action]")]
0 references
public class ProductController : Controller
{
    0 references
    public string Index()
    {
        return "using route attributes.";
    }
    [Route("{name:alpha}")]
    0 references
    public IActionResult example(string name)
    {
        return Content($"<h4>route attribute and constrain==> {name}.</h4>"
                                        ,contentType:"text/html");
    }
}
```

Token Replacement(multiple route)



```
[Route("~/")]
[Route("[controller]")]
[Route("[controller]/[action]")]
0 references
public class ProductController : Controller
{
    0 references
    public string Index()
    {
        return "using route attributes.";
    }
    [Route("{name:alpha?}")]
    0 references
    public IActionResult example(string name)
    {
        return Content($"<strong>{name}</strong>",contentType:"text/html");
    }
}
```

Route constrain

- Route constrain ជួយឱ្យយើងច្រោះឬដាក់កម្រិតលើតម្លៃដែលមិនចង់បានពី controller action´s method parameter (The Route Constraints helps us to filter out or restrict the unwanted values from reaching the controller action)។

Route constrain

- គេមានវិធីសាស្រ្តចំនួនពីរដើម្បីប្រើនូវ route´s constrain:
  - Inline with the URL Parameter
  - Using the Constraint argument of the MapControllerRoute method.

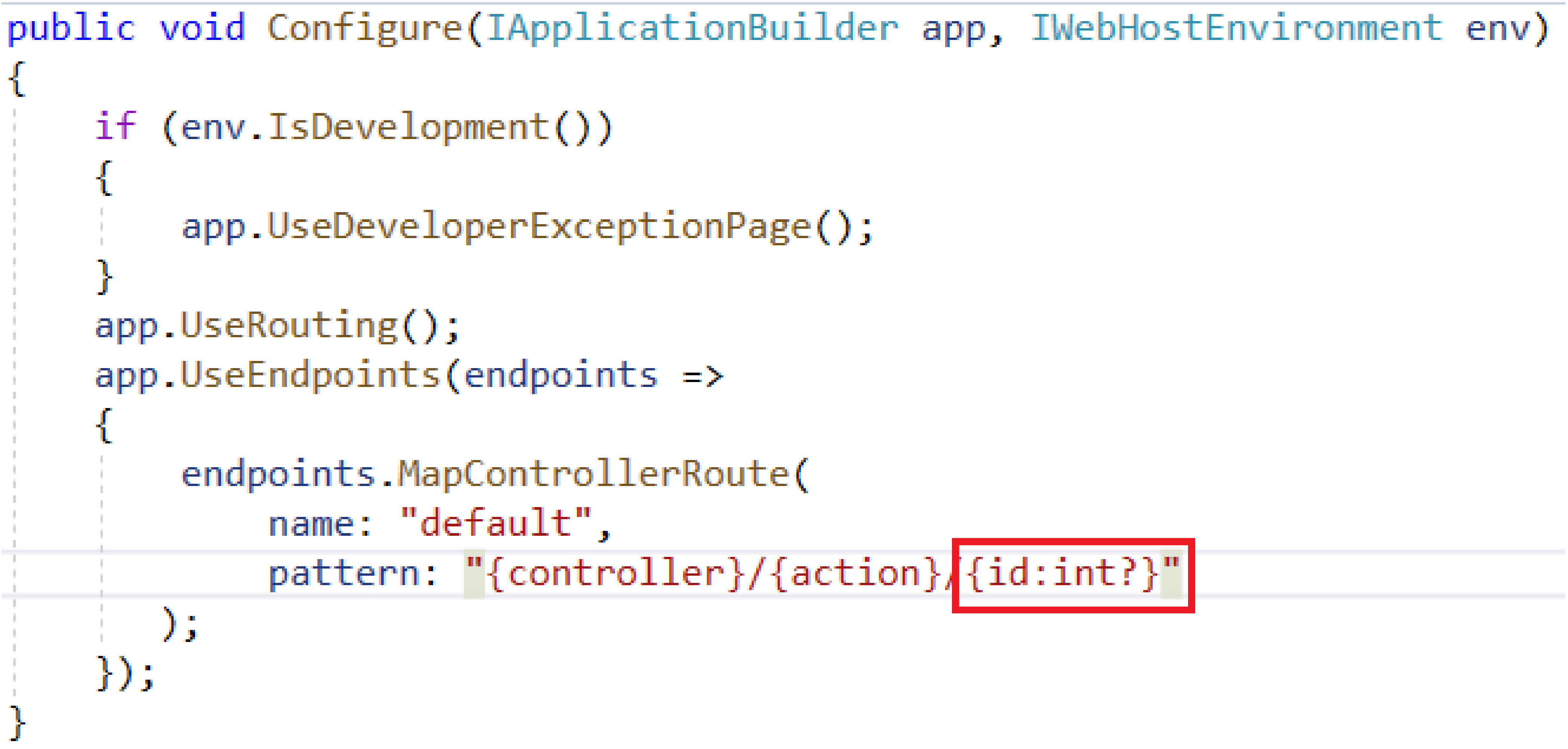
Inline with the URL Parameter

CONSTRAINT	INLINE	NOTES
int	{id:int}	Constrains a route parameter to represent only 32-bit integer values
alpha	{id:alpha}	Constrains a route parameter to contain only lowercase or uppercase letters A through Z in the English alphabet.
bool	{id:bool}	Constrains a route parameter to represent only Boolean values.
datetime	{id:datetime}	Constrains a route parameter to represent only DateTime values.
decimal	{id:decimal}	Constrains a route parameter to represent only decimal values.
double	{id:double}	Constrains a route parameter to represent only 64-bit floating-point values
float	{id:float}	Matches a valid float value (in the invariant culture - see warning)
guid	{id:guid}	Matches a valid Guid value

Inline with the URL Parameter

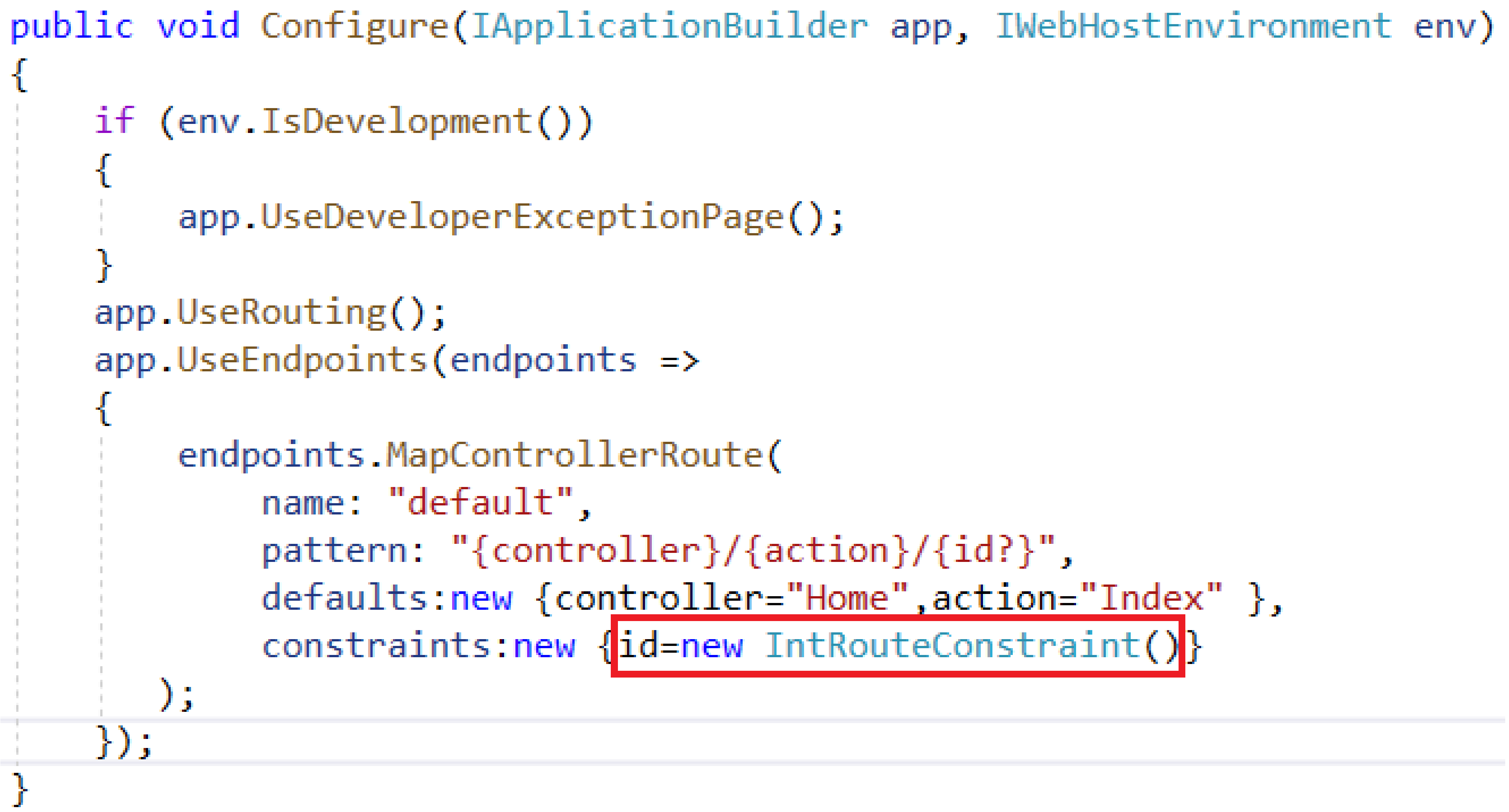
CONSTRAINT	INLINE	NOTES
length(length)	{id:length(12)}	Constrains a route parameter to be a string of a given length or within a given range of lengths.
maxlength(value)	{id:maxlength(8)}	Constrains a route parameter to be a string with a maximum length.
minlength(value)	{id:minlength(4)}	Constrains a route parameter to be a string with a maximum length.
range(min,max)	{id:range(18,120)}	Constraints a route parameter to be an integer within a given range of values.
min(value)	{id:min(18)}	Constrains a route parameter to be a long with a minimum value.
max(value)	{id:max(120)}	Constrains a route parameter to be an integer with a maximum value.
egex(expression)	{ssn:regex(^\\d{{3}}-\\d{{2}}-\\d{{4}}$)}	Constrains a route parameter to match a regular expression.

ឧទាហរណ៍



```
public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
{
    if (env.IsDevelopment())
    {
        app.UseDeveloperExceptionPage();
    }
    app.UseRouting();
    app.UseEndpoints(endpoints =>
    {
        endpoints.MapControllerRoute(
            name: "default",
            pattern: "{controller}/{action}/{id:int?}"
        );
    });
}
```

Constraint method of MapControllerRoute



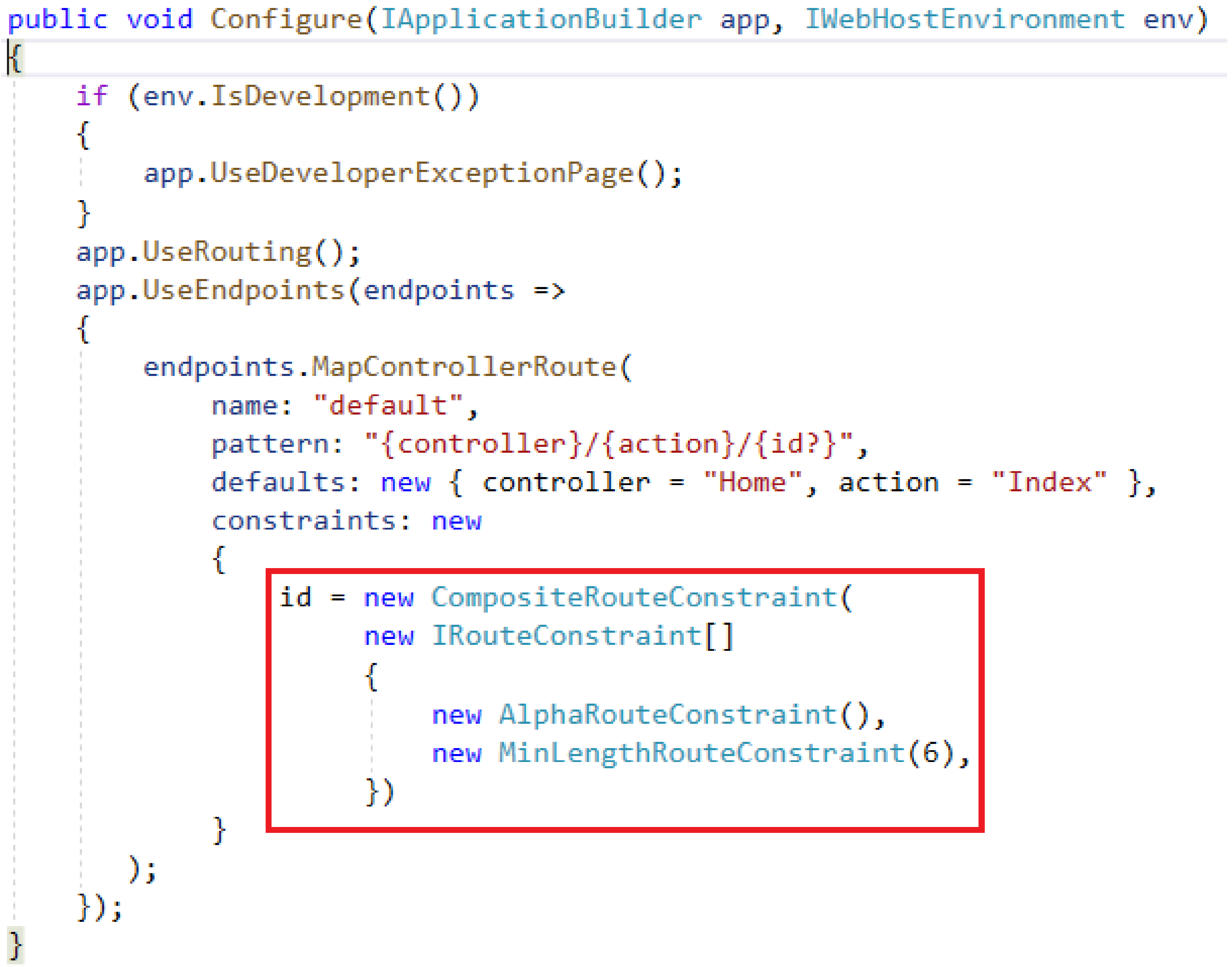
```
public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
{
    if (env.IsDevelopment())
    {
        app.UseDeveloperExceptionPage();
    }
    app.UseRouting();
    app.UseEndpoints(endpoints =>
    {
        endpoints.MapControllerRoute(
            name: "default",
            pattern: "{controller}/{action}/{id?}",
            defaults:new {controller="Home",action="Index" },
            constraints:new {id=new IntRouteConstraint()}
        );
    });
}
```

Constraint method of MapControllerRoute

CONSTRAINT	CLASS	NOTES
int	IntRouteConstraint	Constrains a route parameter to represent only 32-bit integer values
alpha	AlphaRouteConstraint	Constrains a route parameter to contain only lowercase or uppercase letters A through Z in the English alphabet.
bool	BoolRouteConstraint	Constrains a route parameter to represent only Boolean values.
datetime	DateTimeRouteConstraint	Constrains a route parameter to represent only DateTime values.
decimal	DecimalRouteConstraint	Constrains a route parameter to represent only decimal values.
double	DoubleRouteConstraint	Constrains a route parameter to represent only 64-bit floating-point values
float	FloatRouteConstraint	Matches a valid float value (in the invariant culture - see warning)
guid	GuidRouteConstraint	Matches a valid Guid value

Constraint method of MapControllerRoute

CONSTRAINT	CLASS	NOTES
length(length)	LengthRouteConstraint	Constrains a route parameter to be a string of a given length or within a given range of lengths.
maxlength(value)	MaxLengthRouteConstraint	Constrains a route parameter to be a string with a maximum length.
minlength(value)	MinLengthRouteConstraint	Constrains a route parameter to be a string with a maximum length.
range(min,max)	RangeRouteConstraint	Constraints a route parameter to be an integer within a given range of values.
min(value)	MinRouteConstraint	Constrains a route parameter to be a long with a minimum value.
max(value)	MaxRouteConstraint	Constrains a route parameter to be an integer with a maximum value.
egex(expression)	RegexRouteConstraint	Constrains a route parameter to match a regular expression.



```
public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
{
    if (env.IsDevelopment())
    {
        app.UseDeveloperExceptionPage();
    }
    app.UseRouting();
    app.UseEndpoints(endpoints =>
    {
        endpoints.MapControllerRoute(
            name: "default",
            pattern: "{controller}/{action}/{id?}",
            defaults: new { controller = "Home", action = "Index" },
            constraints: new
            {
                id = new CompositeRouteConstraint(
                     new IRouteConstraint[]
                     {
                         new AlphaRouteConstraint(),
                         new MinLengthRouteConstraint(6),
                     })
            }
       );
    });
}
```