Route Constraints

Route constraints let you restrict how the parameters in the route template are matched. The general syntax is "{parameter:constraint}". For example:

C#Copy

```
[Route("users/{id:int}")]
public User GetUserById(int id) { ... }

[Route("users/{name}")]
public User GetUserByName(string name) { ... }
```

Here, the first route will only be selected if the "id" segment of the URI is an integer. Otherwise, the second route will be chosen.

The following table lists the constraints that are supported.

ROUTE CONSTRAINTS

Constraint	Description	Example
alpha	Matches uppercase or lowercase Latin alphabet characters (a-z, A-Z)	{x:alpha}
bool	Matches a Boolean value.	{x:bool}
datetime	Matches a **DateTime** value.	{x:datetime}
decimal	Matches a decimal value.	{x:decimal}
double	Matches a 64-bit floating-point value.	{x:double}
float	Matches a 32-bit floating-point value.	{x:float}
guid	Matches a GUID value.	{x:guid}
int	Matches a 32-bit integer value.	{x:int}

length	Matches a string with the specified length or within a specified range of lengths.	{x:length(6)}
		{x:length(1,20)}

long	Matches a 64-bit integer value.	{x:long}
max	Matches an integer with a maximum value.	{x:max(10)}
maxlength	Matches a string with a maximum length.	{x:maxlength(10)}
min	Matches an integer with a minimum value.	{x:min(10)}
minlength	Matches a string with a minimum length.	{x:minlength(10)}
range	Matches an integer within a range of values.	{x:range(10,50)}
regex	Matches a regular expression.	{x:regex(^\d{3}-\d{3}-\d{4}$)}

Notice that some of the constraints, such as "min", take arguments in parentheses. You can apply multiple constraints to a parameter, separated by a colon.

C#Copy

```
[Route("users/{id:int:min(1)}")]
public User GetUserById(int id) { ... }
```

You can make a URI parameter optional by adding a question mark to the route parameter. If a route parameter is optional, you must define a default value for the method parameter.

C#Copy

```
public class BooksController : ApiController
{
    [Route("api/books/locale/{lcid:int?}")]
    public IEnumerable<Book> GetBooksByLocale(int lcid = 1033) { ... }
}
```

In this example, `/api/books/locale/1033` and `/api/books/locale` return the same resource.

Alternatively, you can specify a default value inside the route template, as follows:

C#Copy

```
public class BooksController : ApiController
{
    [Route("api/books/locale/{lcid:int=1033}")]
    public IEnumerable<Book> GetBooksByLocale(int lcid) { ... }
}
```

This is almost the same as the previous example, but there is a slight difference of behavior when the default value is applied.

- In the first example ("{lcid:int?}"), the default value of 1033 is assigned directly to the method parameter, so the parameter will have this exact value.
- In the second example ("{lcid:int=1033}"), the default value of "1033" goes through the model-binding process. The default model-binder will convert "1033" to the numeric value 1033. However, you could plug in a custom model binder, which might do something different.