



# Enabling CORS In ASP.NET Core API Applications

From this application, you will learn how to enable CORS in ASP.NET Core API applications



Gowtham K

Feb 19 2019

8

4

41.6k

---

[Download Free .NET & JAVA Files API](#)

## What is CORS?

Cross-Origin Resource Sharing (CORS) manages the cross-origin requests. Unlike same-origin policy, CORS allows making a request from one origin to another. CORS allows the servers to specify who can access the resource on the server from outside.

This is in continuation of my last [article](#) (create RESTful API using ASP.NET Core with Entity Framework Core) so I highly recommend you to go through my previous [article](#) for the basic set up of an ASP.NET Core application.

## Cross Domain call

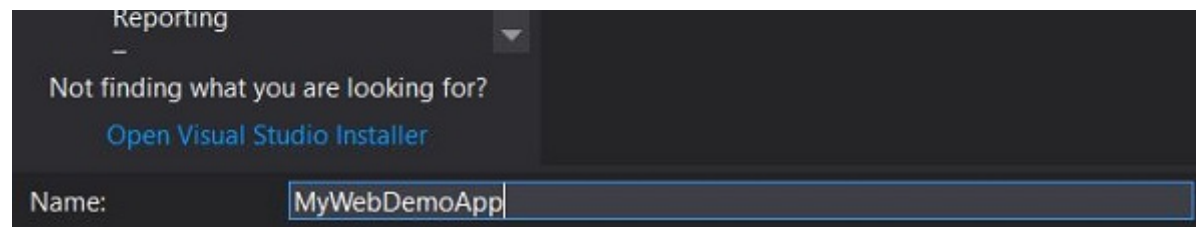
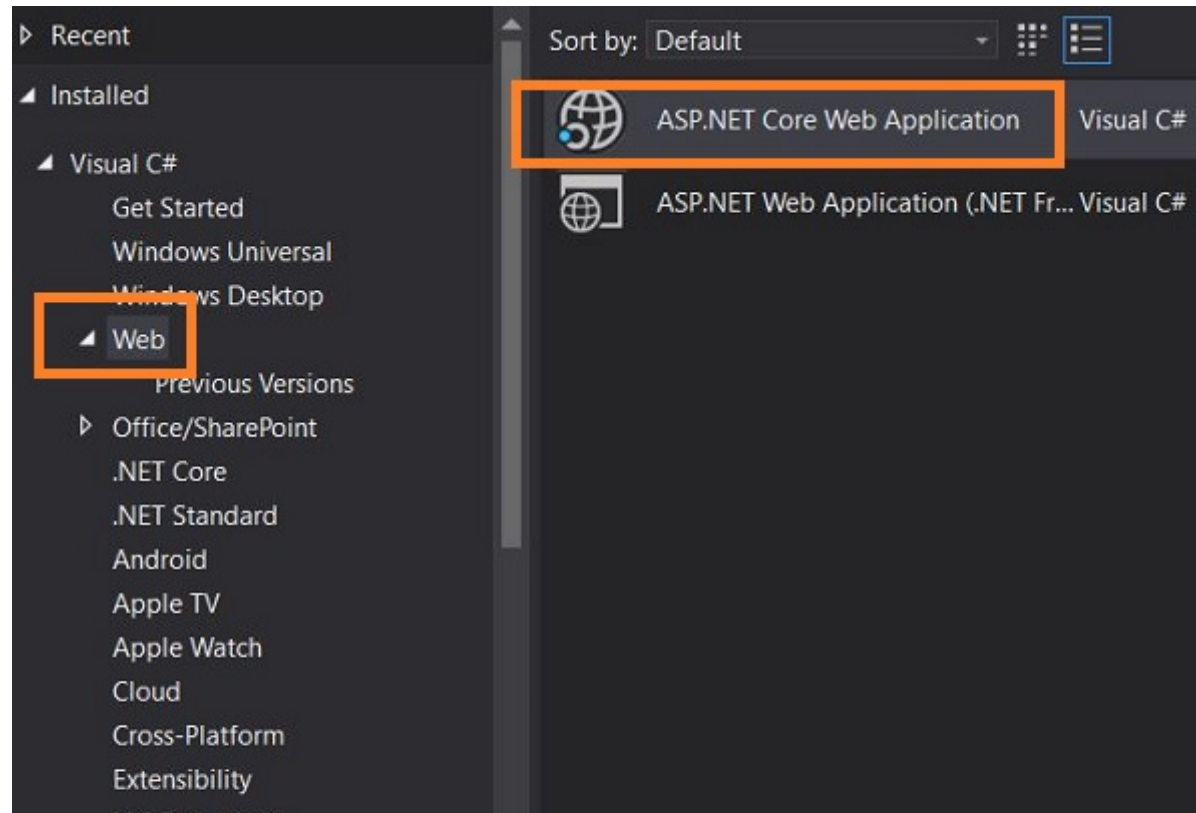
Before enabling the CORS, let's see how the cross-domain call is restricted. Let's create an ASP.NET Core web application.



[ASK A QUESTION](#)

[CONTRIBUTE](#)

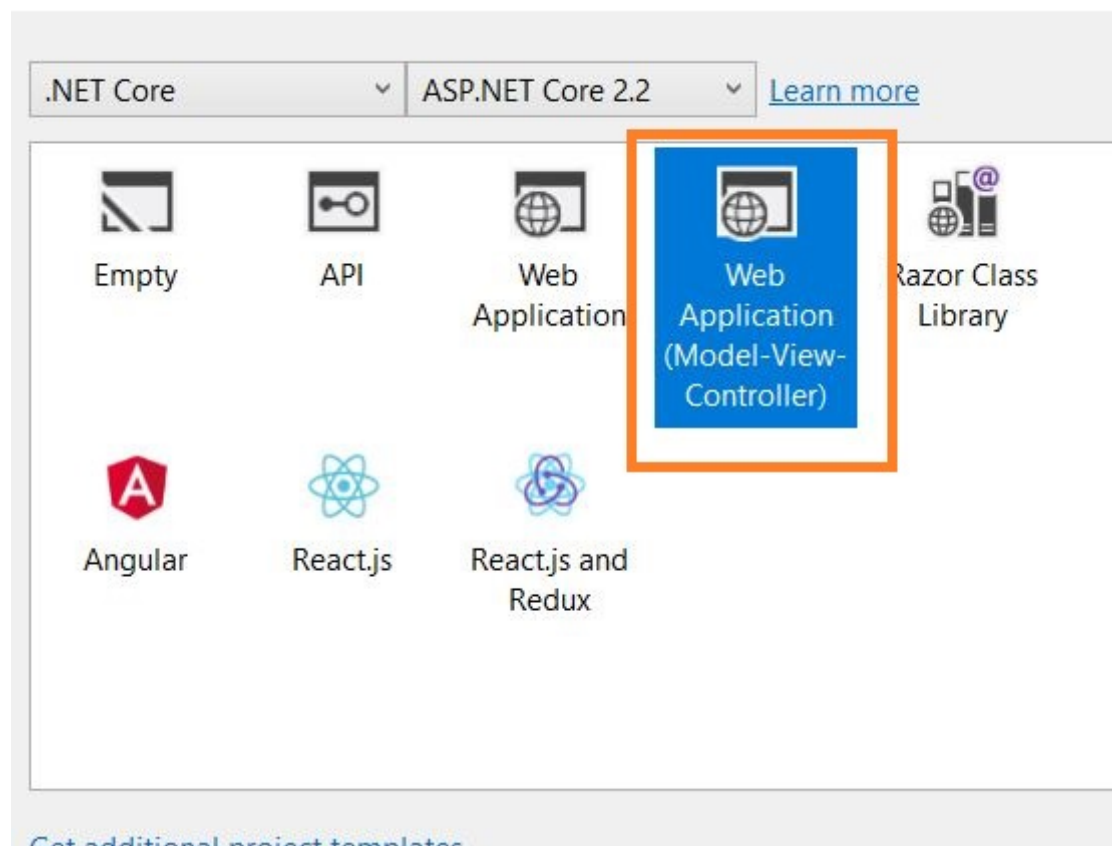
Open Visual Studio, click on NEW ->Project. Select ASP.NET Web Application template under web, as shown in the below figure.



Select web application (Model-View-Controller) template, as shown in the below figure,

[ASK A QUESTION](#)

[CONTRIBUTE](#)



### Step 3

Click OK. This will create a web application with a default template.

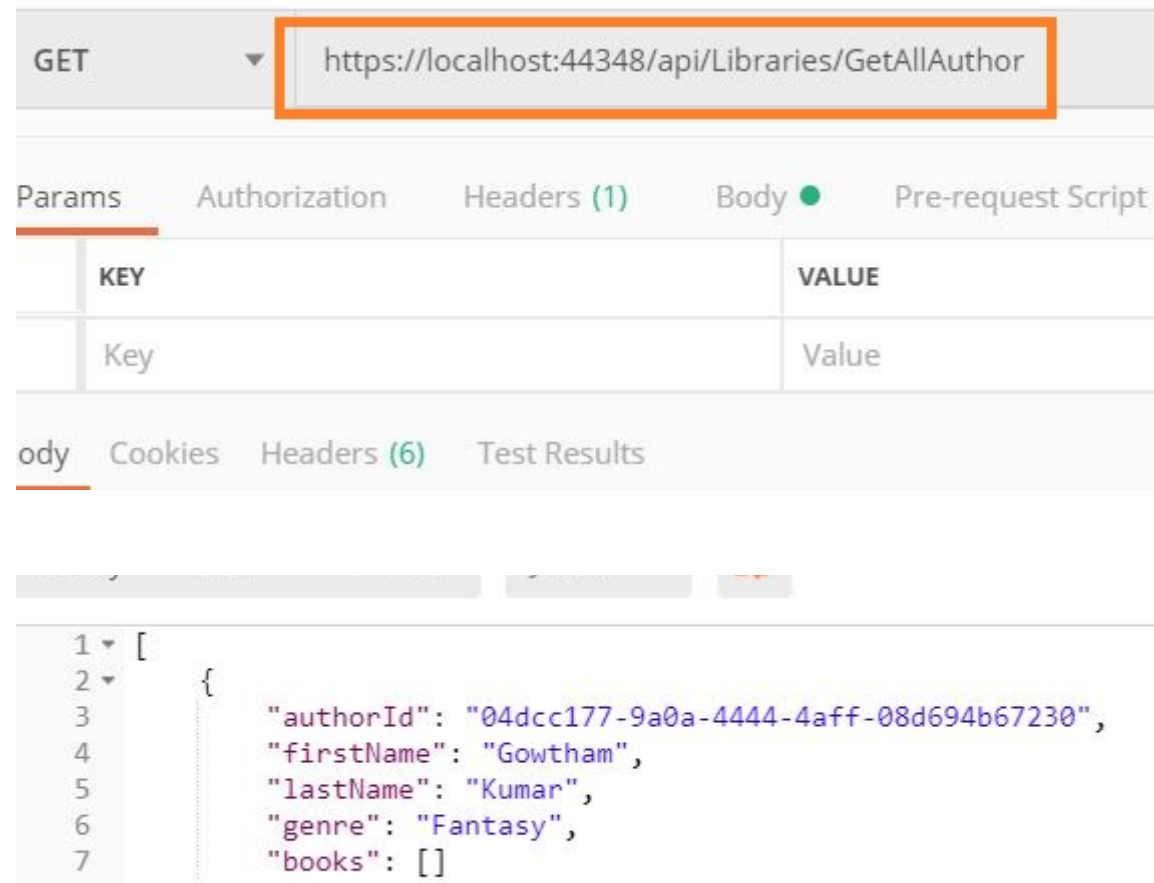
### Step 4

Go to the Index.cshtml page and add the below code and run the application.

cus  
 C# Corner  
 Half C# Corner MVPs Announced  
 03. url: "https://localhost:44348/api/Libraries/GetAllAuthor",  
 04. success: function (result) {  
 05. console.log(result);  
 06. }  
 07. })  
 08. </script>

From the above code, you can notice the AJAX call I made to access the API which is not from the same origin. This is from the ASP.NET Core API application which is created in my last [article](#).

Testing the API in the Postman tool.



```

10      "authorId": "4a20c720-4c71-4103-b4d3-ba0e0d78c33",
11      "firstName": "Bob",
12      "lastName": "Ross",
13      "genre": "Drama",
14      "books": []
15    },
16    {
17      "authorId": "0ec75223-99a4-4df8-8da3-847afa4c1b8a",
18      "firstName": "David",
19      "lastName": "Miller",
20      "genre": "Fantasy",
21      "books": []
22    }
23  ]

```

In the browser console, you will get an error message as shown in the below figure.

```

✖ Access to XMLHttpRequest at 'https://localhost:44348/api/Libraries/GetAllAuthor' (index):1
  from origin 'https://localhost:44342' has been blocked by CORS policy: No 'Access-Control-
  Allow-Origin' header is present on the requested resource.

```

Now it's time to Enable CORS in our API application so that we can access it from a different origin.

## Enable CORS in ASP.NET Core API Application

Open the ASP.NET Core API application which we created in my last article.

Go to Startup.cs file and add the below code in Configure method, which will inject CORS into a container.

```
01. app.UseCors(options => options.AllowAnyOrigin());
```

Add the below code in ConfigureServices method,

```
c.AddPolicy("AllowOrigin", options => options.AllowAnyOrigin());
```

ASK A QUESTION

CONTRIBUTE

The above code tells that the API's can be accessed from any origin globally.

Run the application,

top Filter ASK A QUESTION

```

▼ (3) [{...}, {...}, {...}] ⓘ
  ▼ 0:
    authorId: "04dcc177-9a0a-4444-4aff-08d694b67230"
    books: []
    firstName: "Gowtham"
    genre: "Fantasy"
    lastName: "Kumar"
    __proto__: Object
  ▼ 1:
    authorId: "4a2bc798-4c71-4169-84d9-0a0e0d778c95"
    books: []
    firstName: "Bob"
    genre: "Drama"
    lastName: "Ross"
    __proto__: Object
  ▼ 2:
    authorId: "0ec75223-99a4-4df8-8da3-847afa4c1b8a"
    books: []
    firstName: "David"
    genre: "Fantasy"
    lastName: "Miller"
    __proto__: Object

```

From the above figure you can notice we got a response from the API successfully and the response it printed in browser console was as expected.

## Enabling for origin

```
01. app.UseCors(options=>options.WithOrigins("https://localhost:44342")),
```

Add the below code in ConfigureServices method

```
01. services.AddCors(c =>
02.     {
03.         c.AddPolicy("AllowOrigin", options => options.WithOrigins("https://localhost:44342"));
04.     });
```

Go to controller and decorate the action with Enable CORS attribute, as given below,

```
01. // GET: api/Libraries/GetAllAuthor
02. [HttpGet]
03. [Route("GetAllAuthor")]
04. [EnableCors("AllowOrigin")]
05. public IActionResult GetAllAuthor()
06. {
07.     IEnumerable<Author> authors = _libraryRepository.GetAllAuthor();
08.     return Ok(authors);
09. }
```

Now this API can be accessed only from the origin https://localhost:44342.

We can also define EnableCors at the controller level so that all the actions under this controller can be accessed from the origin

```
01. [Route("api/Libraries")]
02. [ApiController]
03. [EnableCors("AllowOrigin")]
04. public class LibrariesController : ControllerBase
05. {
06.     private readonly ILibraryRepository<Author> _libraryRepository;
07.
08.
09. }
```



```

12.  _libraryRepository = libraryRepository;
13.  }
14.
15.  // GET: api/Libraries/GetAllAuthor
16.  [HttpGet]
17.  [Route("GetAllAuthor")]
18.
19.  public IActionResult GetAllAuthor()
20.  {
21.      IEnumerable<Author> authors = _libraryRepository.GetAllAuthor();
22.      return Ok(authors);
23.  }
24.  }
25.  }

```

ASK A QUESTION

CONTRIBUTE

## Conclusion

We saw how to enable the CORS in ASP.NET Core API applications, will see more about ASP.NET Core in my future articles. I hope you have enjoyed this article. Your valuable feedback, questions, or comments about this article are always welcomed.

API

ASP.NET CORE

ASP.NET Core API Application

Enabling CORS In ASP.NET Core



Gowtham K *TOP 100*



<https://www.c-sharpcorner.com/members/gowtham-k3>

82

3.2m

4

3

8

4