

How to enable CORS in ASP.NET Core



I am trying to enable cross origin resources sharing on my ASP.NET Core Web API, but I am stuck.

129

The `EnableCors` attribute accepts `policyName` of type `string` as parameter:



39

```
// Summary:  
//     Creates a new instance of the Microsoft.AspNetCore.Cors.Core.EnableCorsAttribute.  
//  
// Parameters:  
//     policyName:  
//     The name of the policy to be applied.  
public EnableCorsAttribute(string policyName);
```

What does the `policyName` mean and how can I configure **CORS** on an ASP.NET Core Web API?

c#

asp.net-core

edited Nov 25 '16 at 9:18



Henk Mollema

24.8k 9 64 84

asked Aug 11 '15 at 12:37



Oluwafemi

6,261 9 34 55

1 here is the article docs.microsoft.com/en-us/aspnet/core/security/cors – Elaine Dec 30 '16 at 4:45

7 Answers



317

You have to configure a CORS policy at application startup in the `ConfigureServices` method:

Join **Stack Overflow** to learn, share knowledge, and build your career.

Email Sign Up

OR SIGN IN WITH



Google





```

        .AllowAnyHeader();
    }));
    // ...
}

```

The `CorsPolicyBuilder` in `builder` allows you to configure the policy to your needs. You can now use this name to apply the policy to controllers and actions:

```
[EnableCors("MyPolicy")]
```

Or apply it to every request:

```

public void Configure(IApplicationBuilder app)
{
    app.UseCors("MyPolicy");
    // ...
}

```

edited Mar 22 '16 at 8:41

answered Aug 11 '15 at 12:42



[Henk Mollema](#)

24.8k 9 64 84

-
- 10 I found that `app.UseCors("MyPolicy")` didn't work for my API endpoints. I needed to explicitly `[EnableCors("MyPolicy")]` on the controller endpoints in question. – [robbpriestley](#) Jun 15 '16 at 1:26
-
- 6 You must place that line before any endpoints. – [Saša Četković](#) Feb 16 '17 at 3:00
-
- 6 From [the official docs](#): "To enable CORS for your entire application add the CORS middleware to your request pipeline using the `UseCors` extension method. Note that the CORS middleware must precede any defined endpoints in your app that you want to support cross-origin requests (ex. before any call to `UseMvc`)." – [Alex Klaus](#) Oct 12 '17 at 3:12
-
- 1 I'm trying this but it doesn't work for me. Any thoughts? See also stackoverflow.com/questions/49900141/... – [Patrick Szalapski](#) Apr 18 '18 at 13:03
-
- 1 I added `app.UseCors()` after `app.AddMvc()` and it didn't work. Because order of Use methods affects how the middleware works! – [Obay Abd-Algader](#) Nov 6 '18 at 11:39

Join **Stack Overflow** to learn, share knowledge, and build your career.

Email Sign Up

OR SIGN IN WITH



Google



83

Unfortunately the docs are very confusing in this specific case. So I'll make it dead-simple:

- Add `Microsoft.AspNetCore.Cors` nuget package to your project
- In `ConfigureServices` method, add `services.AddCors();`
- In `Configure` method, before calling `app.UseMvc()` and `app.UseStaticFiles()` , add:

```
app.UseCors(builder => builder
    .AllowAnyOrigin()
    .AllowAnyMethod()
    .AllowAnyHeader()
    .AllowCredentials());
```

That's it. Every client has access to your ASP.NET Core Website/API.

For .Net-Core 2.0

- Add `Microsoft.AspNetCore.Cors` nuget package to your project
- in `ConfigureServices` method, before calling `services.AddMvc()` , add:

```
services.AddCors(options =>
{
    options.AddPolicy("AllowAll",
        builder =>
        {
            builder
                .AllowAnyOrigin()
                .AllowAnyMethod()
                .AllowAnyHeader()
                .AllowCredentials();
        });
});
```

- In `Configure` method, before calling `app.UseMvc()` , add `app.UseCors("AllowAll");`

`AllowAll` is the policy name which we need to mention in `app.UserCors`. It could be any name.



Join **Stack Overflow** to learn, share knowledge, and build your career.

Email Sign Up

OR SIGN IN WITH

 Google

Facebook 

- 7 Thanks. I had my AddCors after UseMvc and that was causing it to not work properly. Yours is the only answer that mentions this. – [JesseNewman19](#) May 25 '17 at 19:44 
- 1 This doesn't work I don't think. When supplying credentials, you can't allow any origin. I'm still trying to get this to work. – [Jason Goemaat](#) Oct 7 '17 at 4:31
- 5 That's the key piece: .UseCors() before .UseMvc() – [Neps](#) Oct 22 '17 at 23:15
- 1 Wasted time with not knowing about UseCors before UserMvc... thanks for the help! – [Thomas](#) Nov 21 '18 at 21:56 

Based on [Henk's answer](#) I have been able to come up with the specific domain, the method I want to allow and also the header I want to enable CORS for:

32

```
public void ConfigureServices(IServiceCollection services)
{
    services.AddCors(options =>
        options.AddPolicy("AllowSpecific", p => p.WithOrigins("http://localhost:1233")
            .WithMethods("GET")
            .WithHeaders("name")));

    services.AddMvc();
}
```

usage:

```
[EnableCors("AllowSpecific")]
```

edited Apr 3 '18 at 7:54

answered Aug 11 '15 at 12:57



[Oluwafemi](#)

6,261 9 34 55

This does not provide an answer to the question. To critique or request clarification from an author, leave a comment below their post. – [zrvan](#) Aug 11 '15 at 14:06

- 12 To critique or request clarification from an author I don't think that's the case here and Henk's answer have been marked already. My

Join Stack Overflow to learn, share knowledge, and build your career.

Email Sign Up

OR SIGN IN WITH



Google



You have to configure in Startup.cs class

4

```
services.AddCors(options =>
{
    options.AddPolicy("CorsPolicy",
        builder => builder.AllowAnyOrigin()
            .AllowAnyMethod()
            .AllowAnyHeader()
            .AllowCredentials());
});
```

edited Apr 10 at 15:21



nPcomp

3,414 1 22 33

answered Oct 12 '18 at 8:05



Sunil Dhappadhule

1,034 4 20

3

```
public void ConfigureServices(IServiceCollection services)
{
    services.AddCors(options =>
    {
        options.AddPolicy("AllowAnyOrigin",
            builder => builder
                .AllowAnyOrigin()
                .AllowAnyMethod()
                .AllowAnyHeader());
    });

    services.Configure<MvcOptions>(options => {
        options.Filters.Add(new CorsAuthorizationFilterFactory("AllowAnyOrigin"));
    });
}
```

Join Stack Overflow to learn, share knowledge, and build your career.

Email Sign Up

OR SIGN IN WITH



Google





Wonderful way to circumvent the already deployed browser security. Just don't! – [JCKödel](#) Jan 17 '18 at 17:50

I don't think this should be down-voted. This is an "alternate" way. You can either use the "UseCors()" method or else add a global MVC filter like what Nathan did here. – [Josh Mouch](#) May 3 '18 at 13:38

this works for me on .net core 2.2 plus adding app.UseCors("AllowAnyOrigin"); in "Configure" method – [Ali](#) Feb 8 at 17:19



If you are hosting on IIS, one possible reason is you are getting this is because IIS is blocking `OPTIONS` verb. I spent almost an hour because of this:

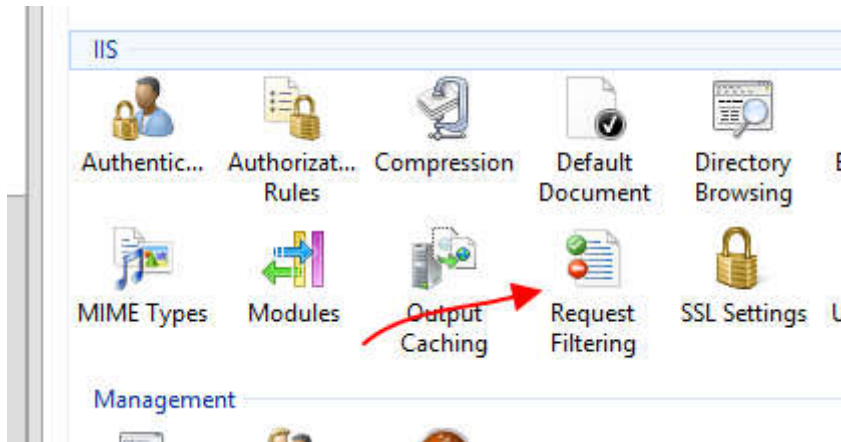
2



One telltale indication is you are getting `404` error during `OPTIONS` request.

To fix this, you need to explicitly tell IIS **not** to block `OPTIONS` request.

Go to Request Filtering:



Make sure `OPTIONS` is allowed:

Join **Stack Overflow** to learn, share knowledge, and build your career.

Email Sign Up

OR SIGN IN WITH



Google





Or, just create a `web.config` with the following setting:

```
<system.webServer>
  <security>
    <requestFiltering>
      <verbs>
        <remove verb="OPTIONS" />
        <add verb="OPTIONS" allowed="true" />
      </verbs>
    </requestFiltering>
  </security>
</system.webServer>
```

edited Nov 14 '18 at 13:39

answered Nov 13 '18 at 11:47



[Rosdi Kasim](#)

12.6k 16 88 125

Thanks..its worked for me – [Nagaraj S](#) May 10 at 9:08

Join **Stack Overflow** to learn, share knowledge, and build your career.

Email Sign Up

OR SIGN IN WITH



Google





instead `.AllowAnyOrigin()` with `.AllowCredentials()`

<https://trailheadtechnology.com/breaking-change-in-aspnetcore-2-2-for-signalr-and-cors/>

<https://github.com/aspnet/AspNetCore/issues/4483>

edited Apr 24 at 18:01

answered Apr 12 at 20:34



PJ3

1,142 15 23

Join **Stack Overflow** to learn, share knowledge, and build your career.

Email Sign Up

OR SIGN IN WITH



Google

Facebook 