* Course Overview
  + Cross-origin resource sharing(CORS)
* Module Overview
  + Configuring CORS in ASP.NET Core
  + Same origin policy
  + CORS request types
* What is Cross-origin Resource…
  + Server needs to be configured to allow request from origin other then the one that it is hosting
  + Same-origin policy
    - Enforced by all browsers
    - Control access between web applications
    - Limit scripts accessing resources from their own origin only
    - Prevent webpages from accessing the DOM of other web pages
    - Prevent webpages from making request on users behalf on other webpages
  + Malicious website can make request to other website using cookies of the other website
    - Cookies get included on each request by the browser
    - So the malicious website makes a request to the other website and cookies for the other website will be attached to the request
  + Origin
    - Protocol + domain + port
    - https + mysite + 80
    - <https://www.mysite.com:80>
  + These three(protocol, domain, port) needs to be exactly the same for origin to be considered the same
  + Examples of same origin
    - <http://www.mysite.com> => <https://www.mysite.com/users>
  + Cross-origin Resource Sharing(CORS)
    - When web clients needs to access api not in the same origin
    - http mechanism that utilizes http headers to define origin permissions
  + web browsers will read cors headers and if origin is part of allowed origins it will allow us access to the response
* Configuring CORS in ASP.NET …
  + Define a policy and assign it to application
  + In Startup.cs
    - In configureServices
    - public void ConfigureServices(IServcieCollection services)
    - {
    - services.AddCors(options => options.AddPolicy(“AllowEverything”, builder => builder.AllowAnyOrigin().AllowAnyMethod().AllowAnyHeaer()));
    - services.AddControllers();
    - }
  + Then use it in Configue
    - public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
    - {
    - …..
    - app.UseCors(“AllowEverything”);
    - ….
    - }
  + app.UseCors have to come before other middleware for Core 3.0
* Restricting CORS Requests to…