

# CORS-Cross-Origin resource Sharing

- ❑ Cross-origin resource sharing is a mechanism that allows restricted resources on a web page to be requested from another domain outside the domain from which the first resource was served. A web page may freely embed cross-origin images, stylesheets, scripts, iframes, and videos.
- ❑ Cross-origin resource sharing (CORS) is a browser mechanism which enables controlled access to resources located outside of a given domain. It extends and adds flexibility to the same-origin policy (SOP). However, it also provides potential for cross-domain based attacks, if a website's CORS policy is poorly configured and implemented. CORS is not a protection against cross-origin attacks such as cross-site request forgery (CSRF).

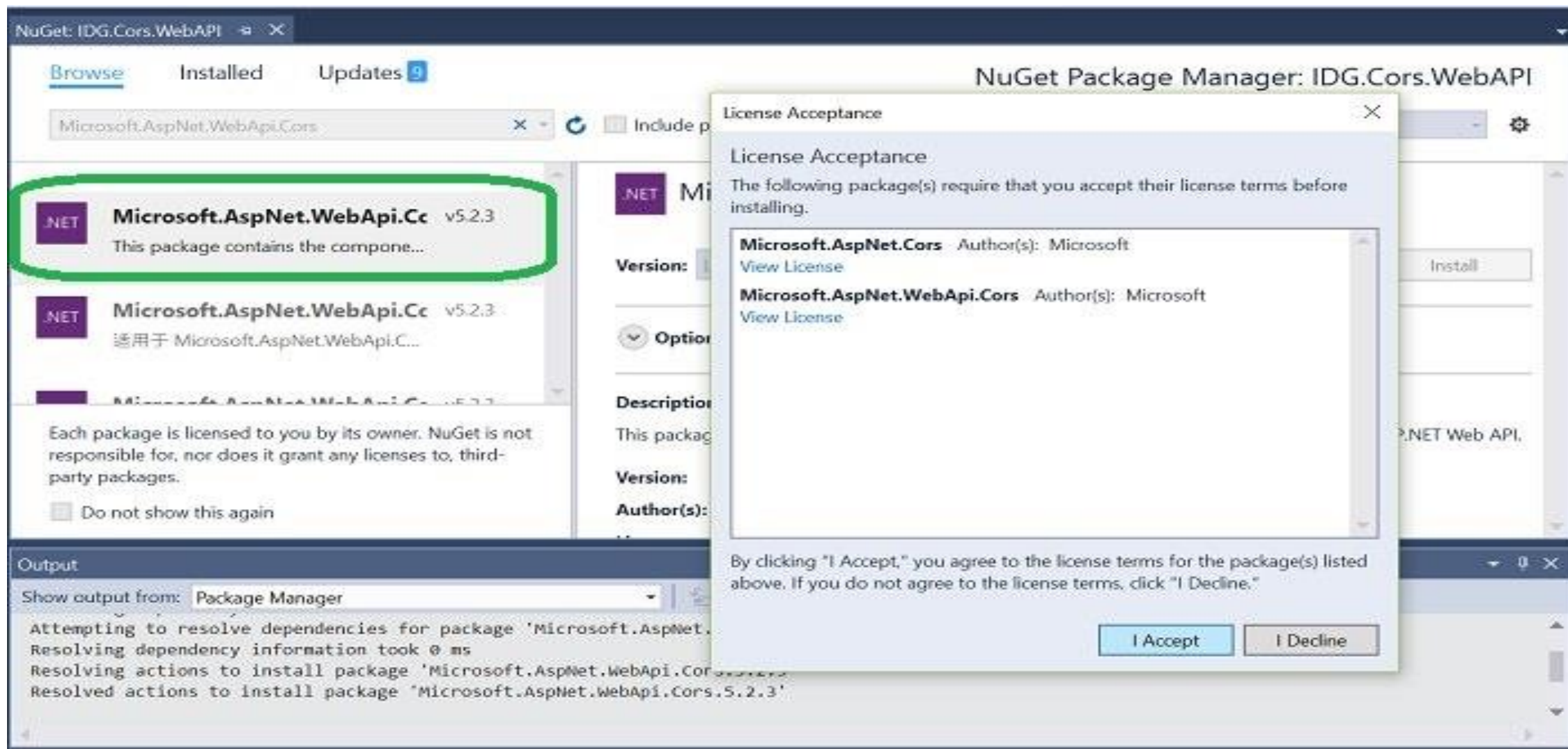
# CORS-Cross-Origin resource Sharing

- ❑ CORS stands for Cross-Origin Resource Sharing. It is a standard that works by adding HTTP headers that allow servers to describe the set of origins that are permitted to read information using a web browser and the kind of request that are allowed.

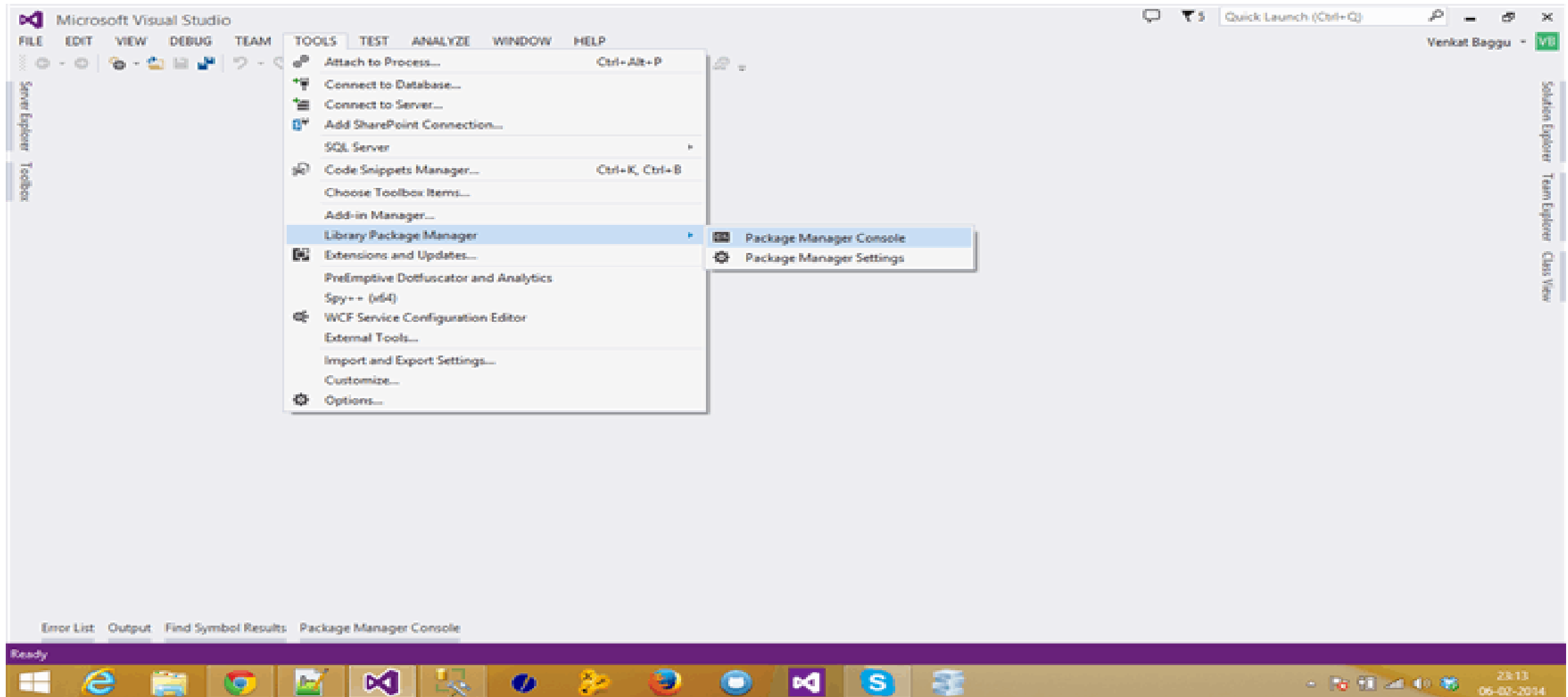
# CORS-Cross-Origin resource Sharing



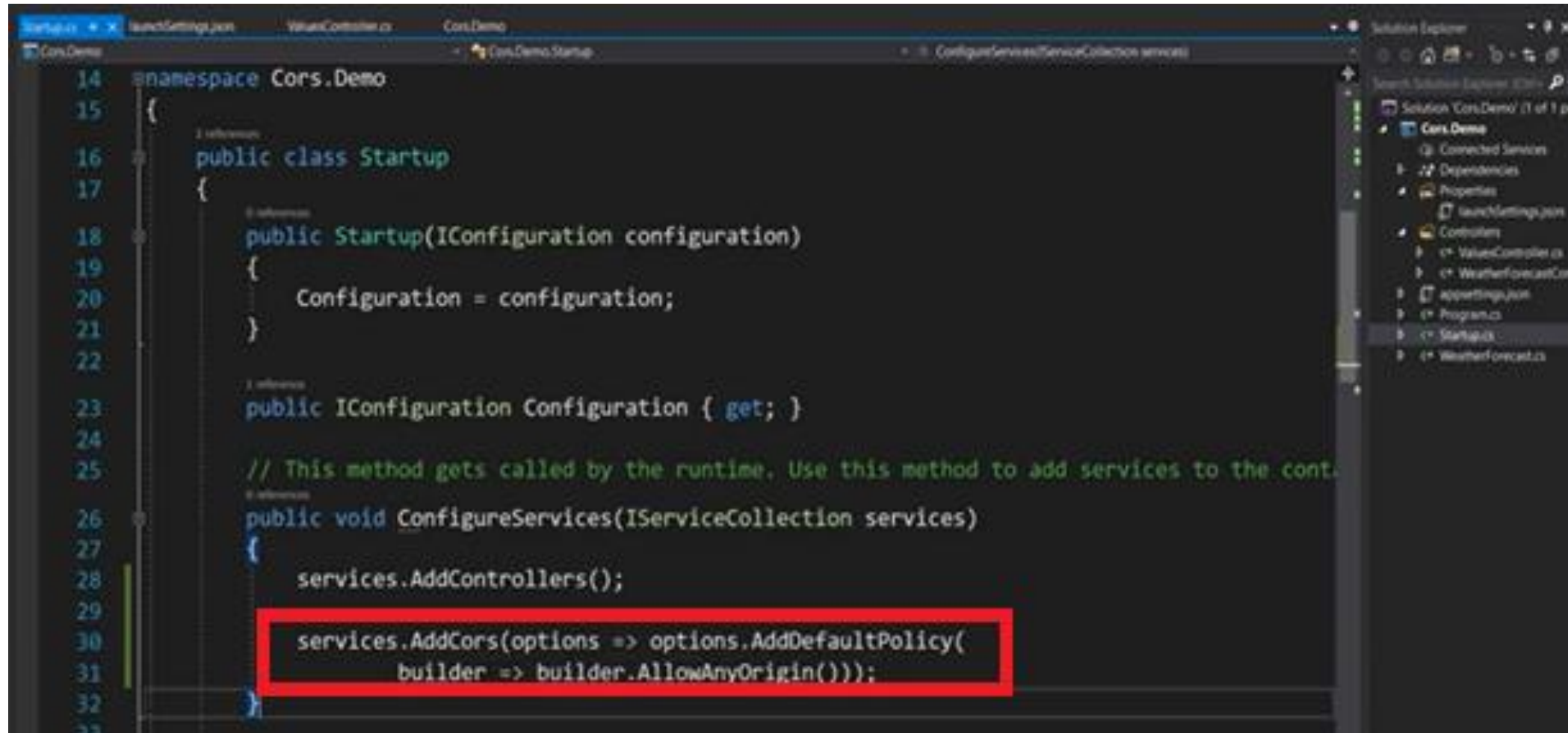
# CORS-Cross-Origin resource Sharing



# CORS-Cross-Origin resource Sharing



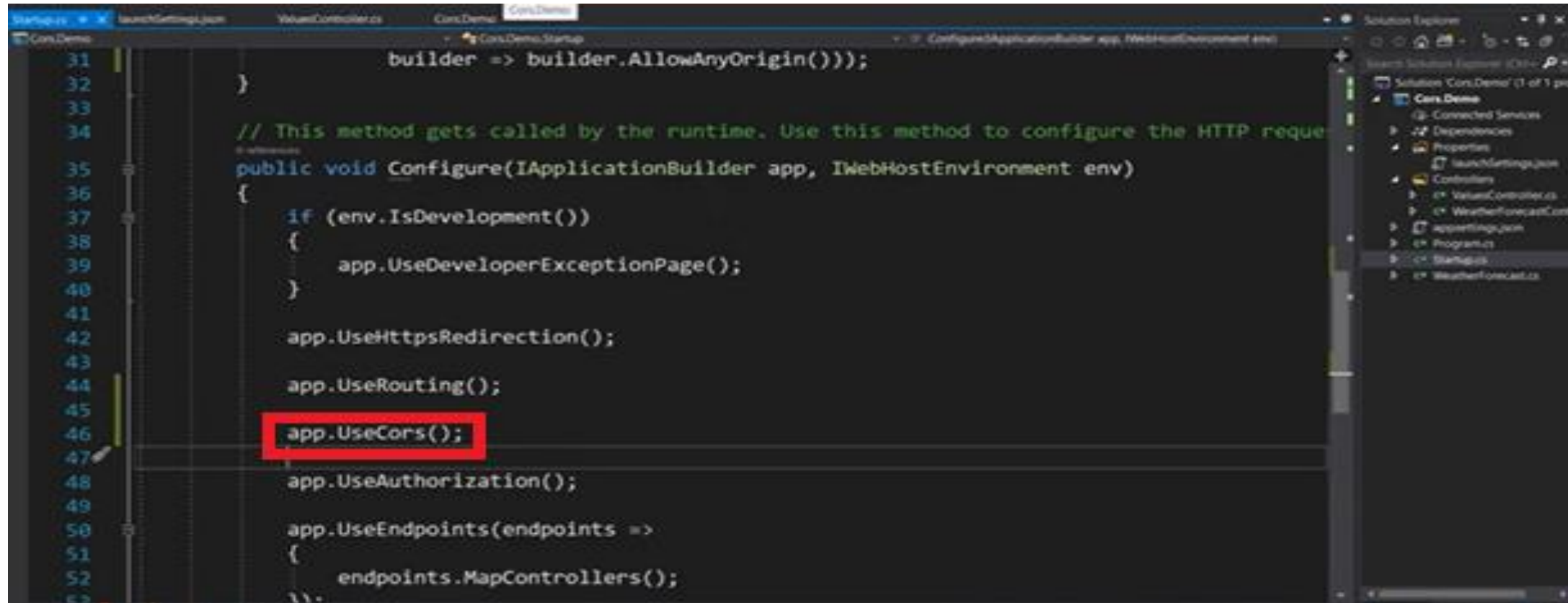
# CORS STEPS-ASP.NET CORE MVC



```
14 namespace Cors.Demo
15 {
16     public class Startup
17     {
18         public Startup(IConfiguration configuration)
19         {
20             Configuration = configuration;
21         }
22
23         public IConfiguration Configuration { get; }
24
25         // This method gets called by the runtime. Use this method to add services to the container.
26         public void ConfigureServices(IServiceCollection services)
27         {
28             services.AddControllers();
29
30             services.AddCors(options => options.AddDefaultPolicy(
31                 builder => builder.AllowAnyOrigin()));
32         }
33     }
```

The image shows a screenshot of the Visual Studio IDE. The main editor window displays the `Cors.Demo.Startup.cs` file. The code defines a `Startup` class with a `ConfigureServices` method. In this method, `services.AddCors` is called with a lambda expression that adds a default CORS policy allowing any origin. This line of code is highlighted with a red rectangular box. The Solution Explorer on the right shows the project structure for `Cors.Demo`, including files like `launchSettings.json`, `Program.cs`, `Startup.cs`, and `WeatherForecast.cs`.

# CORS STEPS-ASP.NET CORE MVC



The screenshot shows the Visual Studio IDE with a C# project named 'Cors.Demo'. The main editor displays the 'Configure' method in 'Program.cs', which is part of the 'ConfigureApplicationBuilder' class. The method is used to configure the HTTP request pipeline. The line 'app.UseCors();' is highlighted with a red rectangle, indicating the step to enable CORS. The Solution Explorer on the right shows the project structure, including 'Controllers', 'Properties', and 'Startup.cs'.

```
31         builder => builder.AllowAnyOrigin());
32     }
33
34     // This method gets called by the runtime. Use this method to configure the HTTP request pipeline
35     public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
36     {
37         if (env.IsDevelopment())
38         {
39             app.UseDeveloperExceptionPage();
40         }
41
42         app.UseHttpsRedirection();
43
44         app.UseRouting();
45
46         app.UseCors();
47
48         app.UseAuthorization();
49
50         app.UseEndpoints(endpoints =>
51         {
52             endpoints.MapControllers();
53         });
54     }
```

# CONSUME -ASP.NET CORE MVC CORS PROJECT IN REACT

The screenshot displays the Visual Studio Code interface for a project titled "App.js - consuming-web-api-in-react". The Explorer sidebar on the left shows the file structure, with the "src" directory expanded to reveal "components", "node\_modules", "public", and "src". The "src" directory contains "components", "App.css", "App.js", "App.test.js", "index.css", "index.js", "logo.svg", "serviceWorker.js", and ".gitignore". The "components" directory contains "contacts.js". The Explorer sidebar also shows "OUTLINE".

The main editor area displays the "App.js" file, which is currently open. The code in "App.js" is as follows:

```
9 }
10
11 state = {
12   contacts: []
13 };
14
15 componentDidMount() {
16   fetch('http://localhost:53832/weatherforecast')
17     .then(res => res.json())
18     .then((data) => {
19       this.setState({ contacts: data })
20     })
21     .catch(console.log)
22 }
23
24 export default App;
```

The status bar at the bottom indicates the current position is "Ln 21, Col 32", the file encoding is "UTF-8", the line ending is "LF", and the language is "JavaScript". The system tray at the bottom right shows the date and time as "16/01/2022 1:15 PM".



# CONSUME -ASP.NET CORE MVC CORS PROJECT IN REACT

contacts.js - consuming-web-api-in-react - Visual Studio Code

EXPLORER

OPEN EDITORS

- JS App.js src
- JS contacts.js src\components
- JS test5.js

CONSUMING-WEB-API-IN-...

- node\_modules
- public
- src
  - components
    - JS contacts.js
- App.css
- App.js
- App.test.js
- index.css
- index.js
- logo.svg
- serviceWorker.js
- .gitignore
- package-lock.json
- package.json
- README.md
- test5.js

OUTLINE

```
src > components > JS contacts.js > [🔗] Contacts > [📦] contacts.map() callback
1  import React from 'react'
2
3  const Contacts = ({contacts}) => {
4    return (
5      <div>
6        <center><h1>Contact List</h1></center>
7        {contacts.map((contact) => (
8          <div class="card">
9            <div class="card-body">
10             <h5 class="card-title">{contact.date}</h5>
11             <p class="card-text">{contact.summary}</p>
12           </div>
13         </div>
14       )]}
15     </div>
16   )
17 };
18
19 export default Contacts
```

Ln 10, Col 67 Spaces: 4 UTF-8 LF {} JavaScript

Type here to search

ENG IN 1:18 PM 16/01/2022

# CORS-Cross-Origin resource Sharing

[Configure CORS STEP BY STEP](#)

# CORS-Cross-Origin resource Sharing

More Information

<https://www.youtube.com/watch?v=4bOVhjA7Nz4>