

# ASP.NET Web APIS

ILogger

Middleware

CORS – react client:

Serving static files

# ASP.NET Core Web API

ILogger

```
✓ public class WeatherForecastController(ILogger<WeatherForecastController> logger) : ControllerBase
{
    [HttpGet]
    0 references
    ✓ public IActionResult Get()
    {
        logger.LogInformation("info");
        logger.LogError("error");
        logger.LogDebug("debug");
        logger.LogTrace("trace");
        logger.LogWarning("warning");
        logger.LogCritical("critical");

        return Ok("Cool!");
    }
}
```

הזרקה של הלוגר  
Controller

שימוש בלוגר

Trace

Debug

Information

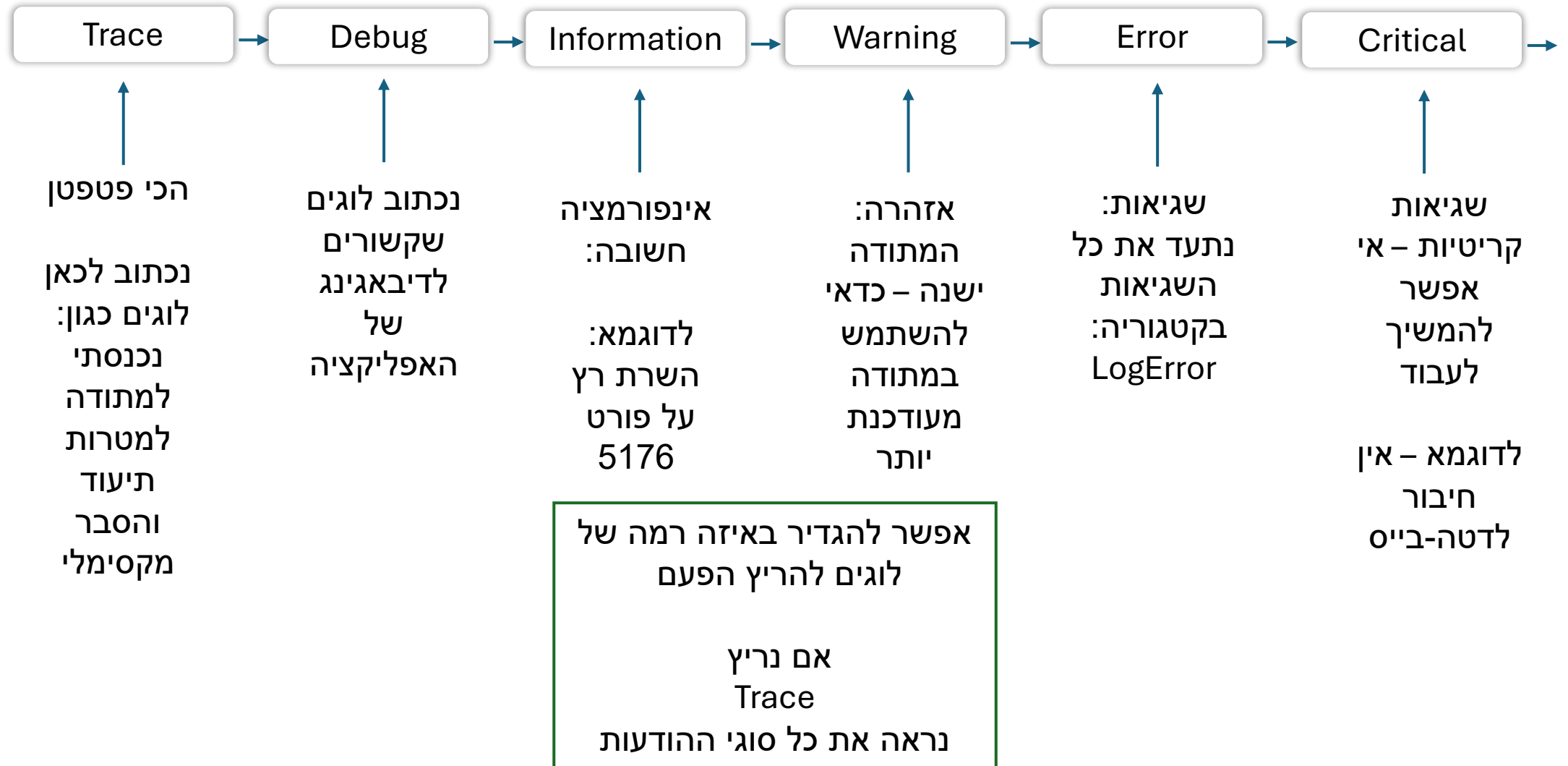
Warning

Error

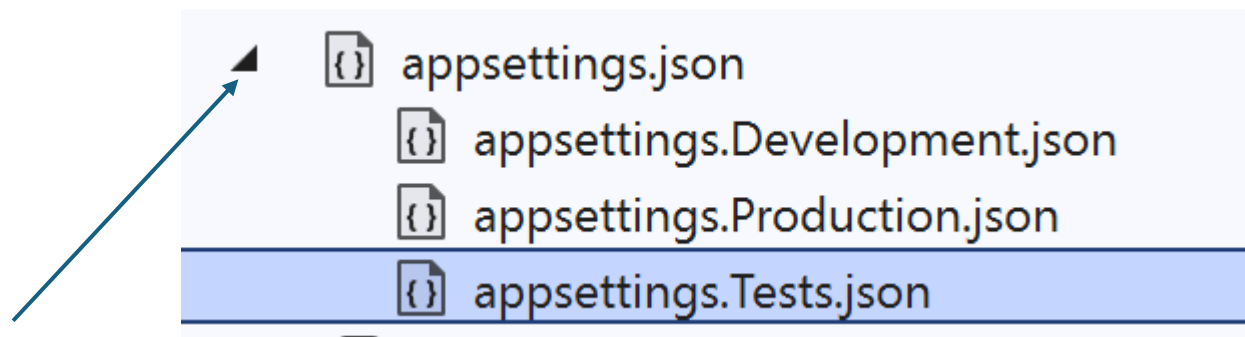
Fatal

# ASP.NET Core Web API

## Log Levels



# קונפיגורציות להרצה



# קונפיגורציות להרצה

הגדרות ב appsettings  
לסינון הלוגים לפי הרמה הרצויה

```
{  
  "Logging": {  
    "LogLevel": {  
      "Default": "Information",  
      "Microsoft.AspNetCore": "Warning"  
    }  
  },  
}
```

{ } appsettings.Development.json

{ } appsettings.json

אפשר להגדיר את זה בהתאם לסביבה שמריצים  
אפשר ליישם הגדרות שונות בהתאם למצב ההרצה – פיתוח/בדיקות/ייצור

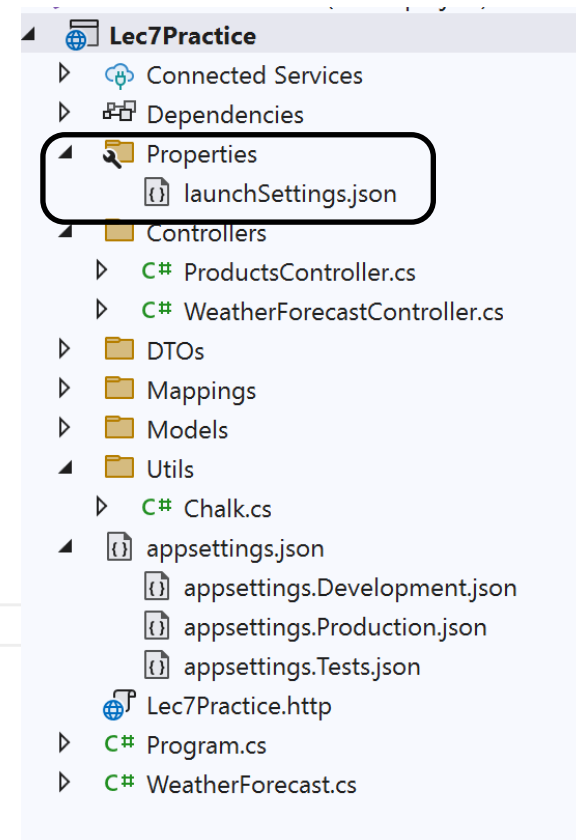
# קביעת הסביבה בקובץ launchSettings.json

## לדוגמא סביבת Tests במקום Development

launchSettings.json

ApisModuleLec3 > Properties > launchSettings.json > {} profiles > {} IIS Express

```
18     "ASPNETCORE_ENVIRONMENT": "Tests",
19   },
20   "dotnetRunMessages": true
21 },
22 "https": {
23   "commandName": "Project",
24   "launchBrowser": true,
25   "launchUrl": "swagger",
26   "applicationUrl": "https://localhost:7032;http://localhost:5059",
27   "environmentVariables": {
28     "ASPNETCORE_ENVIRONMENT": "Tests",
29   },
30   "dotnetRunMessages": true
31 },
32 "IIS Express": {
33   "commandName": "IISExpress",
34   "launchBrowser": true,
35   "launchUrl": "swagger",
36   "environmentVariables": {
37     "ASPNETCORE_ENVIRONMENT": "Tests",
38   }
39 }
```



# הדפסה בצבעים לקונסולה:

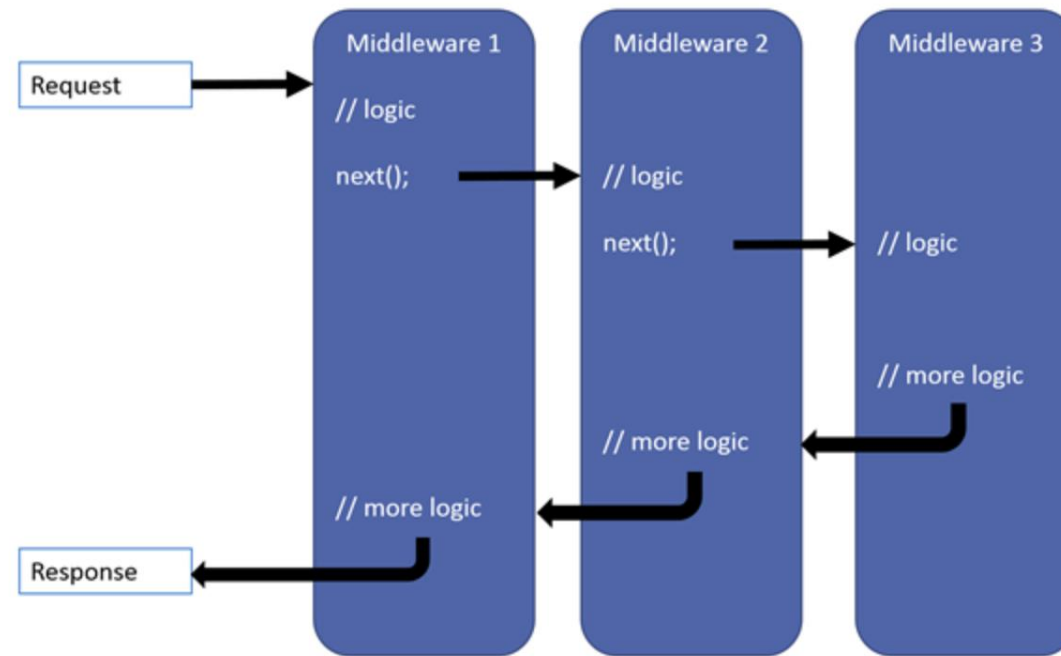
```
namespace ApisModuleLec3.Logs;

public static class Chalk
{
    // Chalk Color and Style methods
    // $" \u001b[31m{text}\u001b[0m"
    // Ansi sequence for red color {text} and reset
    // 1st part is color/style,
    // 2nd part is text,
    // 3rd part is reset to 0m (the default color)
    public static string Red(string text) => $" \u001b[31m{text}\u001b[0m";
    public static string Green(string text) => $" \u001b[32m{text}\u001b[0m";
    public static string Blue(string text) => $" \u001b[34m{text}\u001b[0m";
    public static string Yellow(string text) => $" \u001b[33m{text}\u001b[0m";
    // Custom orange color
    public static string Orange(string text) => $" \u001b[38;5;214m{text}\u001b[0m";
    public static string Bold(string text) => $" \u001b[1m{text}\u001b[0m";
    public static string Underline(string text) => $" \u001b[4m{text}\u001b[0m";

    // Extension methods for chaining
    public static string Red(this string me, string other) => $"{me}{Red(other)}";
    public static string Green(this string me, string other) => $"{me}{Green(other)}";
    public static string Blue(this string me, string other) => $"{me}{Blue(other)}";
    public static string Yellow(this string me, string other) => $"{me}{Yellow(other)}";
    public static string Orange(this string me, string other) => $"{me}{Orange(other)}";
    public static string Bold(this string me, string other) => $"{me}{Bold(other)}";
    public static string Regular(this string me, string other) => $"{me}{other}"; // append text
    public static string Add(this string me, string text) => $"{me} {text}"; // Appends text with space
}
```

```
logger.LogInformation(
    Chalk.Green("Seeding roles completed").Orange("🍊").Yellow("foo").Red("Stuff")
);
```

# Middleware





# Middleware

האזנה לכל הבקשות  
שנכנסות לאפליקציה

אחריות – עלינו לקרוא  
לNEXT

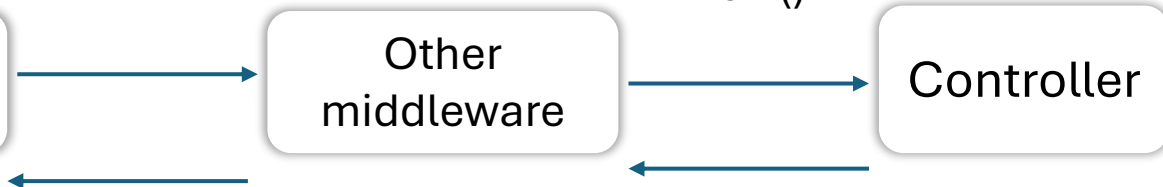
next()

next()

Logger middleware

Other  
middleware

Controller



# Middleware

GET /api/products 200

```
✓ app.Use(async (context, next) =>
{
    //collect data on the incoming request: method path
    var sb = new StringBuilder();
    sb.Append(context.Request.Method)
    .Append(' ')
    .Append(context.Request.Path);

    //let the controller handle it
    await next();

    //check the status
    sb.Append(' ').Append(context.Response.StatusCode);

    //log:
    Console.WriteLine(sb);
});
```

# הגדרה של Middleware שמקבל את כל הבקשות הנכנסות:

```
app.Use(  
    async (context, next) =>  
    {  
        var logger = context.RequestServices.GetRequiredService<ILogger<Program>>();  
        logger.LogInformation("Setting custom response header");  
        context.Response.Headers["MyResponseHeader"] = "My Response Header Value";  
        await next();  
    }  
);
```

# יצירת פרוייקט בראקט עם VITE

נפתח VS CODE בתיקה הרצויה OPEN FOLDER

נפתח טרמינל

`npm create vite`

נגרור את התיקה לvscode

`npm install`

`npm run dev`

# Middleware

## לשליחת CORS Headers

תקשורת בין צד לקוח לצד שרת:

הבקשה נחסמת בגלל מדיניות CORS

לפי המדיניות הזאת - השרת צריך להחזיר Header שמותר ללקוח שלנו לגשת לשרת:

```
App.tsx ×
src > App.tsx > App > useEffect() callback

6 function App() {
7   const [count, setCount] = useState(0);
8
9   useEffect(() => {
10    fetch("https://localhost:7032/api/Cards")
11      .then((response) => response.json())
12      .then((data) => console.log(data));
13  }, []);
14  return (
15    <>
16      <div>
17        <a href="https://vitejs.dev" target="_"
18          <img src={viteLogo} className="logo"
19        </a>
20        <a href="https://react.dev" target="_"
21          <img src={reactLogo} className="logo"
22        </a>
23      </div>
```

# Middleware

## לשליחת CORS Headers

הבקשה נחסמת בגלל מדיניות CORS

לפי המדיניות הזאת - השרת צריך להחזיר Header שמותר ללקוח שלנו לגשת לשרת:

השרת צריך לשלוח את ה Header הבא - כדי לאותת לצד הלקוח שהוא מוזמן לתקשר:

### Program.cs

```
app.Use(  
    async (context, next) =>  
    {  
        context.Response.Headers["Access-Control-Allow-Origin"] = "http://localhost:5173";  
        await next();  
    }  
);
```

עוד ערך אפשרי הוא \*  
כלומר - כל צד לקוח יכול לתקשר  
(טוב לֹא apı שמיועד להיות ציבורי)