CORS IN C#

LET'S LEARN

Written by
Petros Sargsian

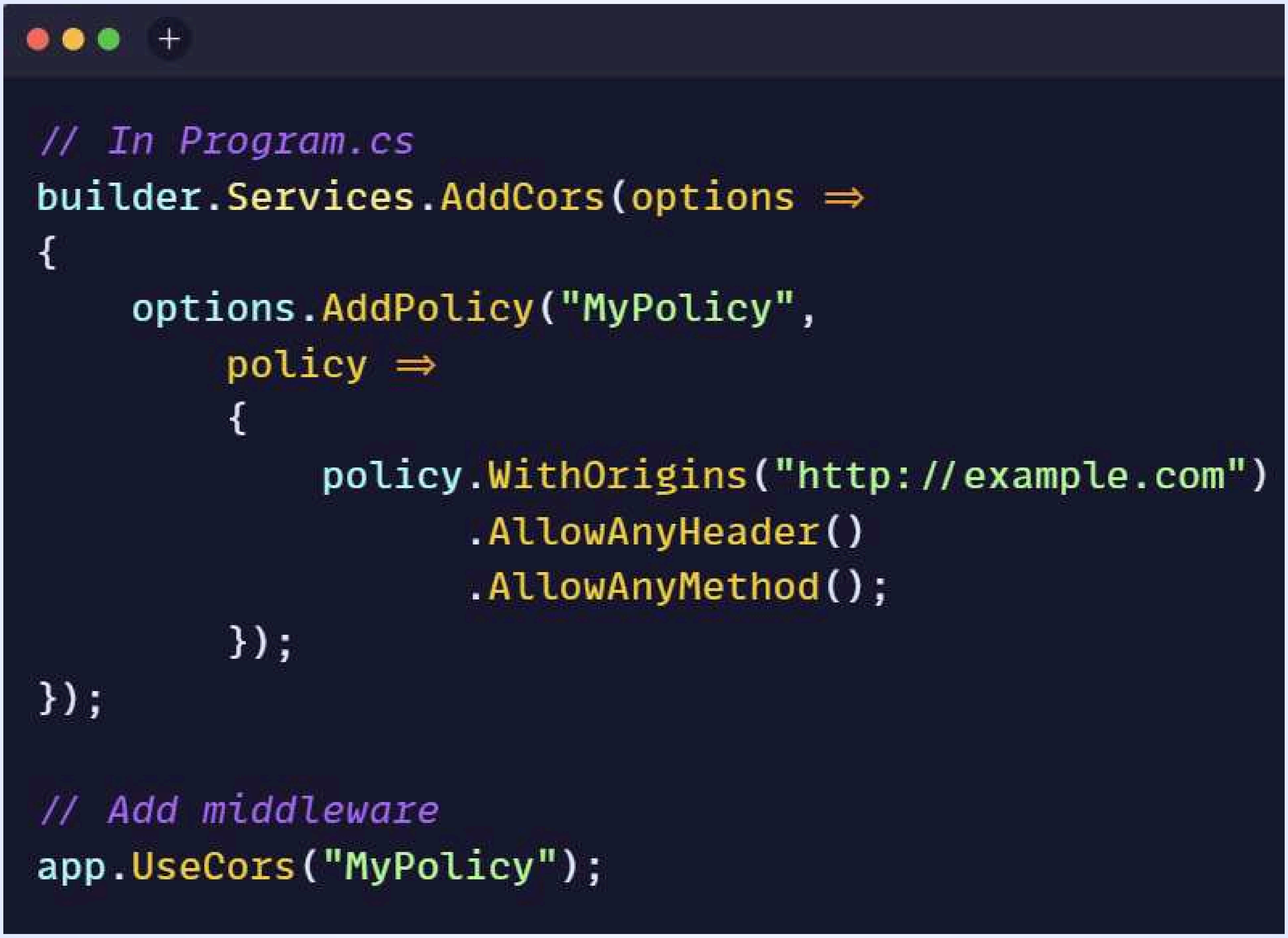
WHAT IS CORS?

Cross-Origin Resource Sharing (CORS) is a security mechanism that allows web applications running on one domain to access resources from a different domain.

- Same-Origin Policy by default
- Preflight requests for complex operations
- Headers control access permissions
- Browser enforces the restrictions

Written by
Petros Sargsian

IMPLEMENTATION IN C#



```
// In Program.cs
builder.Services.AddCors(options =>
{
    options.AddPolicy("MyPolicy",
        policy =>
        {
            policy.WithOrigins("http://example.com")
                  .AllowAnyHeader()
                  .AllowAnyMethod();
        });
});

// Add middleware
app.UseCors("MyPolicy");
```

Written by
Petros Sargsian

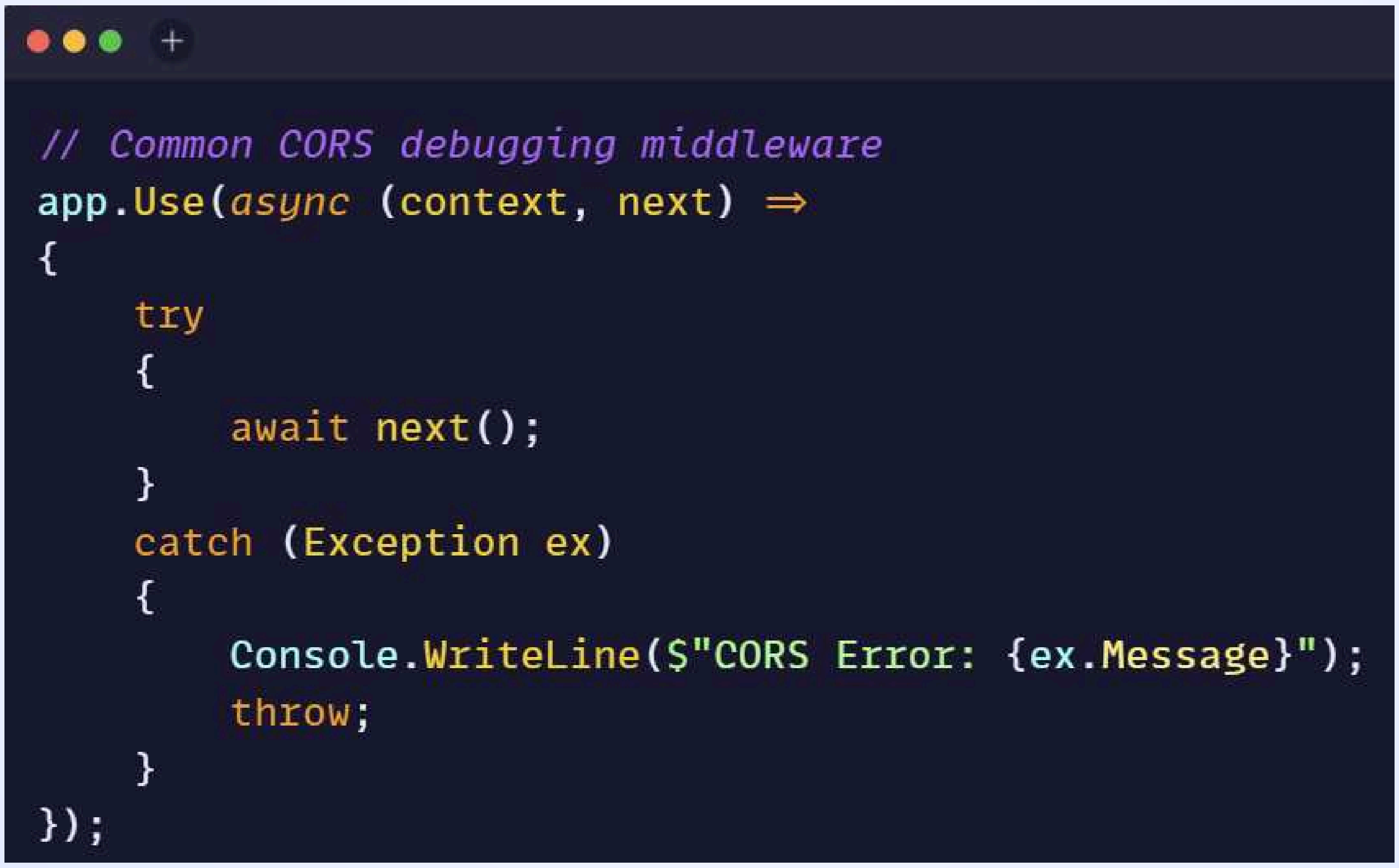
BENEFITS

- Enhanced security through controlled access
- Enables microservices architecture
- Supports modern web development
- Prevents unauthorized data access

Written by
Petros Sargsian

CHALLENGES

- Complex configuration requirements
- Debugging CORS issues can be tricky

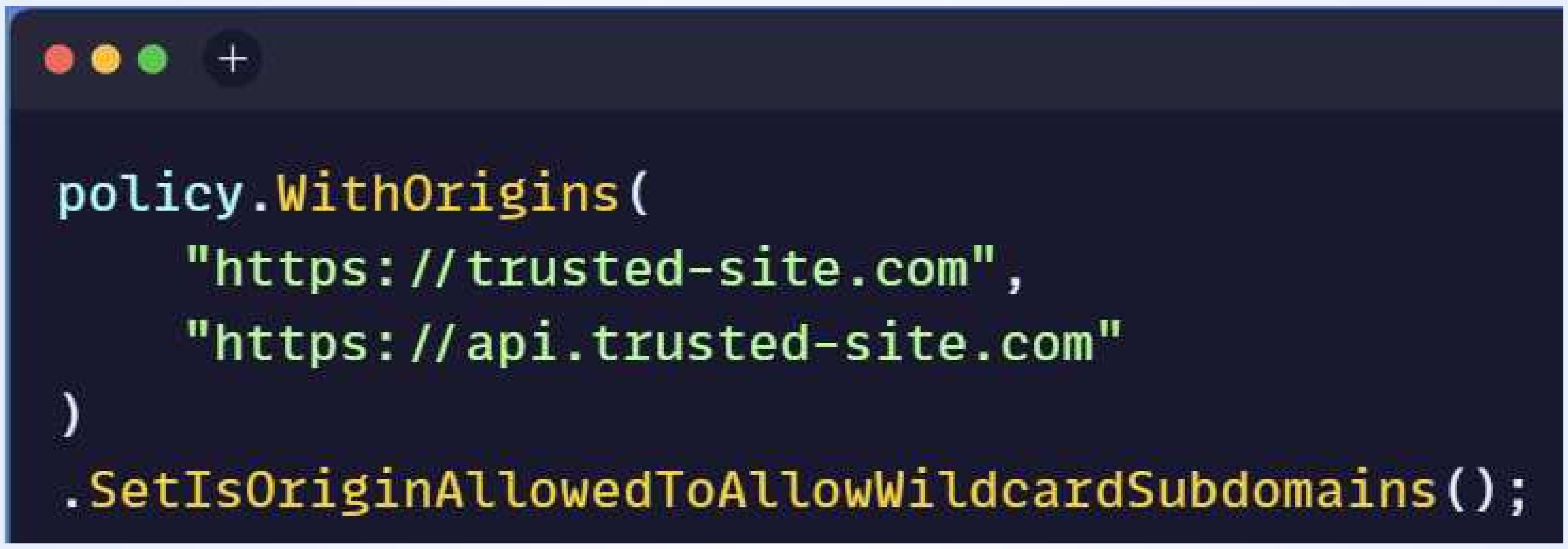


```
// Common CORS debugging middleware
app.Use(async (context, next) =>
{
    try
    {
        await next();
    }
    catch (Exception ex)
    {
        Console.WriteLine($"CORS Error: {ex.Message}");
        throw;
    }
});
```

Written by
Petros Sargsian

BEST PRACTICES

- Be specific with allowed origins



```
policy.WithOrigins(
    "https://trusted-site.com",
    "https://api.trusted-site.com"
)
.SetIsOriginAllowedToAllowWildcardSubdomains();
```

- Use different policies for different endpoints
- Always validate CORS in production environment
- Avoid using AllowAnyOrigin() with credentials

Written by
Petros Sargsian

Follow for more content like this.

Repost   + Follow me