Asp.Net Core MVC and WebApi have great integration with AntiForgery feature:

How to implement this feature in both back-end and front-end

1) Server Site : at "startup.cs" file

* Enable AntiForgery:

services.AddMvc().SetCompatibilityVersion(CompatibilityVersion.Version\_2\_1);

services.AddAntiforgery(o => o.HeaderName = "XSRF-TOKEN");

* Disable AntiForgery globally:

services.AddMvc().AddRazorPagesOptions(options =>

       {

           options.Conventions.ConfigureFilter(new IgnoreAntiforgeryTokenAttribute());

       }

).SetCompatibilityVersion(CompatibilityVersion.Version\_2\_1);

2) Controller

[HttpPost]

[ValidateAntiForgeryToken] - Enable AntiForgery for this action

public IActionResult PostDB(Data db)

{ ... }

[HttpPost]

[IgnoreAntiforgeryToken] - Disable AntiForgery only for this action

public IActionResult PostDB(Data db)

{ ... }

3) Front-end : HTML inject AntiForgery Token ; Ajax Call inject token to headers

 MVC Razor: Add @Html.AntiForgeryToken(); to HTML form

 @Html.AntiForgeryToken();

Ajax Headers:

       $.ajax({

           dataType: 'json',

           data: data,

           beforeSend: function (xhr) {

               xhr.setRequestHeader("XSRF-TOKEN",

                   $('input:hidden[name="\_\_RequestVerificationToken"]').val());

           }, .......

});