

are-restful-apis)

Create REST API

(<https://www.pragimtech.com/blog/asp.net-core-rest-api>)

REST API DbContext

(<https://www.pragimtech.com/blog/asp.net-core-rest-api-dbcontext>)

Repository Pattern in REST API

(<https://www.pragimtech.com/blog/asp.net-core-rest-api-repository-pattern>)

REST API Get List

(<https://www.pragimtech.com/blog/asp.net-core-rest-api-list-of-resources>)

REST API Get By Id

(<https://www.pragimtech.com/blog/asp.net-core-rest-api-get-by-id>)

Post in REST API

(<https://www.pragimtech.com/blog/asp.net-core-rest-api-post>)

REST API Model Validation

(<https://www.pragimtech.com/blog/asp.net-core-rest-api-model-validation>)

Put in REST API

(<https://www.pragimtech.com/blog/asp.net-core-rest-api-put>)

Delete in REST API

(<https://www.pragimtech.com/blog/asp.net-core-rest-api-delete>)

Search in REST API

(<https://www.pragimtech.com/blog/asp.net-core-rest-api-search>)

Blazor authorize attribute

In our previous video we discussed `AuthorizeView` component. This component is used to authorize the display of specific parts within a page, including child components.

AuthorizeView Component

Protects specific parts with in a page or child components

Authorize Attribute

Protects routable components

in-asp.net-core-rest-api)
Call REST API from Blazor

are-restful-apis)

Create REST API

(<https://www.pragimtech.com/blog/asp.net-core-rest-api>)

REST API DbContext

(<https://www.pragimtech.com/blog/asp.net-core-rest-api-dbcontext>)

Repository Pattern in REST API

(<https://www.pragimtech.com/blog/asp.net-core-rest-api-repository-pattern>)

REST API Get List

(<https://www.pragimtech.com/blog/asp.net-core-rest-api-list-of-resources>)

REST API Get By Id

(<https://www.pragimtech.com/blog/asp.net-core-rest-api-get-by-id>)

Post in REST API

(<https://www.pragimtech.com/blog/asp.net-core-rest-api-post>)

REST API Model Validation

(<https://www.pragimtech.com/blog/asp.net-core-rest-api-model-validation>)

Put in REST API

(<https://www.pragimtech.com/blog/asp.net-core-rest-api-put>)

Delete in REST API

(<https://www.pragimtech.com/blog/asp.net-core-rest-api-delete>)

Search in REST API

(<https://www.pragimtech.com/blog/asp.net-core-rest-api-search>)

We use `[Authorize]` attribute to protect routable components (i.e components with `@page` directive). We reach these components via the router and authorization is performed while being routed to these components.

MỚI	MỚI	-25%	MỚI	-21%
-24%	-30%	MỚI	MỚI	-15%

Phong Vũ

So this means, `[Authorize]` attribute cannot be used to protect parts of a page or child components. For that we use `AuthorizeView` component and for routable components we use `[Authorize]` attribute.

Authorize attribute in Blazor and AuthorizeRouteView component

In the following example, `[Authorize]` attribute is used in it's simplest form, without any parameters (i.e roles or policies), so, it only checks if the user is authenticated.

in-asp.net-core-rest-api)
Call REST API from Blazor

are-restful-apis)

Create REST API

(<https://www.pragimtech.com/blog/asp.net-core-rest-api>)

REST API DbContext

(<https://www.pragimtech.com/blog/core-rest-api-dbcontext>)

Repository Pattern in REST API

(<https://www.pragimtech.com/blog/api-repository-pattern>)

REST API Get List

(<https://www.pragimtech.com/blog/core-rest-api-list-of-resources>)

REST API Get By Id

(<https://www.pragimtech.com/blog/core-rest-api-get-by-id>)

Post in REST API

(<https://www.pragimtech.com/blog/in-aspnet-core-rest-api>)

REST API Model Validation

(<https://www.pragimtech.com/blog/api-model-validation>)

Put in REST API

(<https://www.pragimtech.com/blog/in-asp.net-core-rest-api>)

Delete in REST API

(<https://www.pragimtech.com/blog/in-asp.net-core-rest-api>)

Search in REST API

(<https://www.pragimtech.com/blog>)

If the user is authenticated, then the authorization is granted otherwise not.

```
@page "/"
@inherits EmployeeListBase

@attribute [Authorize]

@*Rest of the component code*@
```

Blazor authorize attribute not working

- For the authorization to work as expected, along with the `[Authorize]` attribute we must also use `AuthorizeRouteView` component instead of `RouteView` component.
- In `App.razor` file, use `AuthorizeRouteView` component instead of `RouteView` component.

```
<AuthorizeRouteView RouteData="@routeData" DefaultLayout="@typeof(MainLayout)" />
```

in-asp.net-core-rest-api)

Call REST API from Blazor

are-restful-apis)

Create REST API

(https://www.pragimtech.com/blog/

asp.net-core-rest-api)

REST API DbContext

(https://www.pragimtech.com/blog/

core-rest-api-dbcontext)

Repository Pattern in REST API

(https://www.pragimtech.com/blog/

api-repository-pattern)

REST API Get List

(https://www.pragimtech.com/blog/

core-rest-api-list-of-resources)

REST API Get By Id

(https://www.pragimtech.com/blog/

core-rest-api-get-by-id)

Post in REST API

(https://www.pragimtech.com/blog/

in-aspnet-core-rest-api)

REST API Model Validation

(https://www.pragimtech.com/blog/

api-model-validation)

Put in REST API

(https://www.pragimtech.com/blog/

in-asp.net-core-rest-api)

Delete in REST API

(https://www.pragimtech.com/blog/

in-asp.net-core-rest-api)

Search in REST API

(https://www.pragimtech.com/blog/

MỚI	MỚI	-25%	MỚI	-21%
-24%	-30%	MỚI	MỚI	-15%

Phong Vũ

Role-based or Policy-based authorization

The [Authorize] attribute also supports role-based and policy-based authorization. For role-based authorization, use the `Roles` parameter:

```
@page "/"
@attribute [Authorize(Roles = "administrator, manager")]

<p>Only users in administrator or manager role are allowed access</p>
```

For policy-based authorization, use the `Policy` parameter:

in-asp.net-core-rest-api)
Call REST API from Blazor

are-restful-apis)

Create REST API
(<https://www.pragimtech.com/blog/asp.net-core-rest-api>)

REST API DbContext
(<https://www.pragimtech.com/blog/asp.net-core-rest-api-dbcontext>)

Repository Pattern in REST API
(<https://www.pragimtech.com/blog/asp.net-core-rest-api-repository-pattern>)

REST API Get List
(<https://www.pragimtech.com/blog/asp.net-core-rest-api-list-of-resources>)

REST API Get By Id
(<https://www.pragimtech.com/blog/asp.net-core-rest-api-get-by-id>)

Post in REST API
(<https://www.pragimtech.com/blog/asp.net-core-rest-api-post>)

REST API Model Validation
(<https://www.pragimtech.com/blog/asp.net-core-rest-api-model-validation>)

Put in REST API
(<https://www.pragimtech.com/blog/asp.net-core-rest-api-put>)

Delete in REST API
(<https://www.pragimtech.com/blog/asp.net-core-rest-api-delete>)

Search in REST API
(<https://www.pragimtech.com/blog/asp.net-core-rest-api-search>)

@page "/"

@attribute [Authorize(Policy = "admin-policy")]

<p>Only users who satisfy admin-policy are allowed access</p>

Customize unauthorized content

If access is not allowed, "Not authorized" is the content that is displayed by default. However, we can customize this unauthorized content.



Phong Vũ

in-asp.net-core-rest-api)
Call REST API from Blazor

are-restful-apis)

Create REST API
(<https://www.pragimtech.com/blog/asp.net-core-rest-api>)

REST API DbContext
(<https://www.pragimtech.com/blog/asp.net-core-rest-api-dbcontext>)

Repository Pattern in REST API
(<https://www.pragimtech.com/blog/asp.net-core-rest-api-repository-pattern>)

REST API Get List
(<https://www.pragimtech.com/blog/asp.net-core-rest-api-list-of-resources>)

REST API Get By Id
(<https://www.pragimtech.com/blog/asp.net-core-rest-api-get-by-id>)

Post in REST API
(<https://www.pragimtech.com/blog/asp.net-core-rest-api>)

REST API Model Validation
(<https://www.pragimtech.com/blog/asp.net-core-rest-api-model-validation>)

Put in REST API
(<https://www.pragimtech.com/blog/asp.net-core-rest-api>)

Delete in REST API
(<https://www.pragimtech.com/blog/asp.net-core-rest-api>)

Search in REST API
(<https://www.pragimtech.com/blog/asp.net-core-rest-api>)

```
<CascadingAuthenticationState>
  <Router AppAssembly="@typeof(Program).Assembly">
    <Found Context="routeData">
      <AuthorizeRouteView RouteData="@routeData" DefaultLayout="@typeof(MainLayout)">
        <NotAuthorized>
          <p>Sorry, you're not authorized to reach this page.</p>
          <p>You may need to log in as a different user.</p>
        </NotAuthorized>
      </AuthorizeRouteView>
    </Found>
    <NotFound>
      <LayoutView Layout="@typeof(MainLayout)">
        <p>Sorry, there's nothing at this address.</p>
      </LayoutView>
    </NotFound>
  </Router>
</CascadingAuthenticationState>
```



(<https://www.pragimtech.com/blog/blazor/authorization-in-blazor>)
(<https://www.pragimtech.com/blog/blazor/blazor-authentication-and-authorization-in-code>)



© 2020 Pragimtech (<https://www.pragimtech.com>). All Rights Reserved.

in-asp.net-core-rest-api)

Call REST API from Blazor

are-restful-apis)

Create REST API

(<https://www.pragimtech.com/blog/asp.net-core-rest-api>)

REST API DbContext

(<https://www.pragimtech.com/blog/asp.net-core-rest-api-dbcontext>)

Repository Pattern in REST API

(<https://www.pragimtech.com/blog/asp.net-core-rest-api-repository-pattern>)

REST API Get List

(<https://www.pragimtech.com/blog/asp.net-core-rest-api-list-of-resources>)

REST API Get By Id

(<https://www.pragimtech.com/blog/asp.net-core-rest-api-get-by-id>)

Post in REST API

(<https://www.pragimtech.com/blog/asp.net-core-rest-api>)

REST API Model Validation

(<https://www.pragimtech.com/blog/asp.net-core-rest-api-model-validation>)

Put in REST API

(<https://www.pragimtech.com/blog/asp.net-core-rest-api>)

Delete in REST API

(<https://www.pragimtech.com/blog/asp.net-core-rest-api>)

Search in REST API

(<https://www.pragimtech.com/blog/asp.net-core-rest-api>)

