are-restful-apis)
Create REST API (https://www.pragimtech.com/blo asp.net-core-rest-api)
REST API DbContext (https://www.pragimtech.com/blo core-rest-api-dbcontext)
Repository Pattern in REST API (https://www.pragimtech.com/blo api-repository-pattern)
REST API Get List (https://www.pragimtech.com/blo core-rest-api-list-of-resources)
REST API Get By Id (https://www.pragimtech.com/blo core-rest-api-get-by-id)
Post in REST API (https://www.pragimtech.com/blo in-aspnet-core-rest-api)
REST API Model Validation (https://www.pragimtech.com/blo api-model-validation)
Put in REST API (https://www.pragimtech.com/blo in-asp.net-core-rest-api)
Delete in REST API (https://www.pragimtech.com/blo in-asp.net-core-rest-api)
Search in REST API (https://www.pragimtech.com/blo

Authentication and authorization state data in code in blazor

In our previous video we discussed, how to use `[Authorize]` attribute to authorize access to routable components (i.e components with `@page` directive). We did this in the component html.

In this video we will discuss, how to obtain authentication and authorization state data in code in blazor.

Cascading AuthenticationState parameter

- Cascading AuthenticationState parameter (`Task<AuthenticationState>`) provides authentication and authorization state data.
- If the user is not authenticated, the request is redirected to the `login` page.
- The return url is also passed as the query string parameter to the login page.
- Upon successful login, the user will be redirected to the page he was trying to access.

in-asp.net-core-rest-api)
Call REST API from Blazor
are-restful-apis)
Create REST API
(https://www.pragimtech.com/blo
asp.net-core-rest-api)
REST API DbContext
(https://www.pragimtech.com/blo
core-rest-api-dbcontext)
Repository Pattern in REST API
(https://www.pragimtech.com/blo
api-repository-pattern)
REST API Get List
(https://www.pragimtech.com/blo
core-rest-api-list-of-resources)
REST API Get By Id
(https://www.pragimtech.com/blo
core-rest-api-get-by-id)
Post in REST API
(https://www.pragimtech.com/blo
in-aspnet-core-rest-api)
REST API Model Validation
(https://www.pragimtech.com/blo
api-model-validation)
Put in REST API
(https://www.pragimtech.com/blo
in-asp.net-core-rest-api)
Delete in REST API
(https://www.pragimtech.com/blo
in-asp.net-core-rest-api)
Search in REST API
(https://www.pragimtech.com/blo

in-asp.net-core-rest-api)
Call REST API from Blazor
are-restful-apis)
Create REST API
(https://www.pragimtech.com/blo
asp.net-core-rest-api)
REST API DbContext
(https://www.pragimtech.com/blo
core-rest-api-dbcontext)
Repository Pattern in REST API
(https://www.pragimtech.com/blo
api-repository-pattern)
REST API Get List
(https://www.pragimtech.com/blo
core-rest-api-list-of-resources)
REST API Get By Id
(https://www.pragimtech.com/blo
core-rest-api-get-by-id)
Post in REST API
(https://www.pragimtech.com/blo
in-aspnet-core-rest-api)
REST API Model Validation
(https://www.pragimtech.com/blo
api-model-validation)
Put in REST API
(https://www.pragimtech.com/blo
in-asp.net-core-rest-api)
Delete in REST API
(https://www.pragimtech.com/blo
in-asp.net-core-rest-api)
Search in REST API
(https://www.pragimtech.com/blo

```
public class EditEmployeeBase : ComponentBase
{
    [CascadingParameter]
    private Task<AuthenticationState> authenticationStateTask { get; set; }

    [Inject]
    public NavigationManager NavigationManager { get; set; }

    protected async override Task OnInitializedAsync()
    {
        var authenticationState = await authenticationStateTask;

        if (!authenticationState.User.Identity.IsAuthenticated)
        {
            string returnUrl = WebUtility.UrlEncode($"/editEmployee/{Id}");
            NavigationManager.NavigateTo($"/identity/account/login?returnUrl={retur
        }

        // rest of the code
    }
}
```

in-asp.net-core-rest-api)
Call REST API from Blazor
are-restful-apis)
Create REST API
(https://www.pragimtech.com/blo
asp.net-core-rest-api)
REST API DbContext
(https://www.pragimtech.com/blo
core-rest-api-dbcontext)
Repository Pattern in REST API
(https://www.pragimtech.com/blo
api-repository-pattern)
REST API Get List
(https://www.pragimtech.com/blo
core-rest-api-list-of-resources)
REST API Get By Id
(https://www.pragimtech.com/blo
core-rest-api-get-by-id)
Post in REST API
(https://www.pragimtech.com/blo
in-aspnet-core-rest-api)
REST API Model Validation
(https://www.pragimtech.com/blo
api-model-validation)
Put in REST API
(https://www.pragimtech.com/blo
in-asp.net-core-rest-api)
Delete in REST API
(https://www.pragimtech.com/blo
in-asp.net-core-rest-api)
Search in REST API
(https://www.pragimtech.com/blo

Check if authenticated user is in a specific role

```
if (authenticationState.User.IsInRole("Administrator"))
{
    // Execute Admin logic
}
```

Check if authenticated user satisfies a specific policy

`Task<AuthenticationState>` can be combined with `IAuthorizationService`, to check if a specific aothorization policy is satisfied.

Phong Vũ

in-asp.net-core-rest-api)
Call REST API from Blazor
are-restful-apis)
Create REST API
(https://www.pragimtech.com/blo
asp.net-core-rest-api)
REST API DbContext
(https://www.pragimtech.com/blo
core-rest-api-dbcontext)
Repository Pattern in REST API
(https://www.pragimtech.com/blo
api-repository-pattern)
REST API Get List
(https://www.pragimtech.com/blo
core-rest-api-list-of-resources)
REST API Get By Id
(https://www.pragimtech.com/blo
core-rest-api-get-by-id)
Post in REST API
(https://www.pragimtech.com/blo
in-aspnet-core-rest-api)
REST API Model Validation
(https://www.pragimtech.com/blo
api-model-validation)
Put in REST API
(https://www.pragimtech.com/blo
in-asp.net-core-rest-api)
Delete in REST API
(https://www.pragimtech.com/blo
in-asp.net-core-rest-api)
Search in REST API
(https://www.pragimtech.com/blo

```
public class EditEmployeeBase : ComponentBase
{
    [CascadingParameter]
    private Task<AuthenticationState> authenticationStateTask { get; set; }

    [Inject]
    private IAuthorizationService AuthorizationService { get; set; }

    protected async override Task OnInitializedAsync()
    {
        var user = (await authenticationStateTask).User;

        if ((await AuthorizationService.AuthorizeAsync(user, "admin-policy"))
        .Succeeded)
        {
            // Execute code specific to admin-policy
        }
    }
}
```

Prev (https://www.pragimtech.com/blog/blazor/blazor-authorize-attribute)
Next (https://www.pragimtech.com/blog/blazor/blazor-course-wrap-up)

© 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/blo
asp.net-core-rest-api)
REST API DbContext
(https://www.pragimtech.com/blo
core-rest-api-dbcontext)
Repository Pattern in REST API
(https://www.pragimtech.com/blo
api-repository-pattern)
REST API Get List
(https://www.pragimtech.com/blo
core-rest-api-list-of-resources)
REST API Get By Id
(https://www.pragimtech.com/blo
core-rest-api-get-by-id)
Post in REST API
(https://www.pragimtech.com/blo
in-aspnet-core-rest-api)
REST API Model Validation
(https://www.pragimtech.com/blo
api-model-validation)
Put in REST API
(https://www.pragimtech.com/blo
in-asp.net-core-rest-api)
Delete in REST API
(https://www.pragimtech.com/blo
in-asp.net-core-rest-api)
Search in REST API
(https://www.pragimtech.com/blo