

Custom Filter

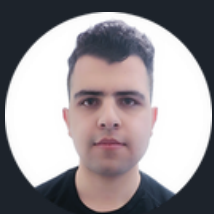
ExcludeRolesFilter is a custom **authorization filter** to exclude certain roles from accessing an endpoint.

```
public class ExcludeRolesFilter : IAuthorizationFilter
{
    private readonly List<string> _roles;

    public ExcludeRolesFilter(string roles)
    {
        _roles = roles.Split(',').ToList();
    }

    public void OnAuthorization(AuthorizationFilterContext context)
    {
        var user = context.HttpContext.User;

        foreach (var role in _roles)
        {
            if (user.IsInRole(role))
            {
                context.Result =
                    new StatusCodeResult(
                        (int)System.Net.HttpStatusCode.Forbidden);
                return;
            }
        }
    }
}
```



Elliot One

Respective Attribute

Attributes will be helpful to **encapsulate** the logic of applying filters. These attributes improve **reusability**.

```
public class ExcludeRolesAttribute : TypeFilterAttribute
{
    public ExcludeRolesAttribute(string roles) :
        base(typeof(ExcludeRolesFilter))
    {
        Arguments = new object[] { roles };
    }
}
```

```
[ExcludeRolesAttribute("Writer, Customer")]
[Area("Admin")]
public class AdminController : Controller
{
}
```



Elliot One



Elliot One

Enjoyed Reading This?

Reshare and Spread Knowledge.

