Übung ASP.NET Core OpenIddict

Gegeben ist eine Visual Studio 2017 Solution mit einem .Net Core Client Projekt und einem Webservice. Ein Großteil der Authentifizierungsinfrastruktur ist bereits gegeben. Außerdem werden folgende User bereits angelegt:

|  |  |
| --- | --- |
| User: Chuck Norris LoginName: chuck Passwort: norris  Claims  Department: Customer Management | User: Jason Bourne  LoginName: jason Passwort: bourne  Claims  Department: Customer Support |

## Aufgabe 1: Policies definieren

Suchen Sie in der Solution nach „TODO Policies“

Definieren Sie folgende Policies:

|  |  |  |
| --- | --- | --- |
| Policy | ClaimType | Werte |
| CanReadCustomerAge | Department | Customer Management |
| CanDeleteCustomer | Department | Customer Management |
| CanUpdateCustomer | Department | Customer Management und Customer Support |
| CanCreateCustomer | Department | Customer Management und Customer Support |

Tipp: Sie können gerne die bestehenden Klassen: AppPolicies, AppClaimTypes und Departments nutzen.

## Aufgabe 2: CustomersController absichern

Sichern Sie bitte den CustomersController wie folgt:

* Der Controller soll allgemein eine Autorisation einfordern, nur die Methoden Get() und GetById() dürfen anonym aufgerufen werden.
* Die anderen CRUD-Operationen sollen durch die von Ihnen in Aufgabe 1 definierten Policies geschützt werden.

## Aufgabe 3: Client

* Suchen Sie in der Solution nach „TODO Client“
* Implementieren Sie die fehlende Funktionalität
* Hinweis: Die Klasse TokenResponse ist bereits vorhanden
* Testen Sie Ihren Client gegen Ihre API

## Aufgabe 4: Alter schützen

* Stellen Sie in Get() und GetById sicher, dass nur Benutzer welche die Policy „CanReadCustomerAge“ erfüllen das Alter des Kunden ausgeliefert bekommen
* Hinweis: Nutzen Sie den IAuthorizationService im Controller

Tipps:

FormUrlEncodedContent, response.Content.ReadAsAsync<TokenResponse>(), AuthenticationHeaderValue, IAuthorizationService

Viel Erfolg!