Checka ut branch azureadaccess\_Start.

Kontrollera att följande Nuget Package finns tillagda till Bunt.Web:

Microsoft.AspNetCore.Identity 2.0.1  
Microsoft.AspNetCore.Authorization 2.0.1  
Microsoft.AspNetCore.Authentication 2.0.1  
Microsoft.AspNetCore.Authentication.Cookies 2.0.1  
Microsoft.AspNetCore.Authentication.OpenIdConnect 2.0.1  
  
Lägg till nedan namespaces för ASP.NET Core-säkerhet samt nya klasser i Bunt.Core coh Bunt.Web:

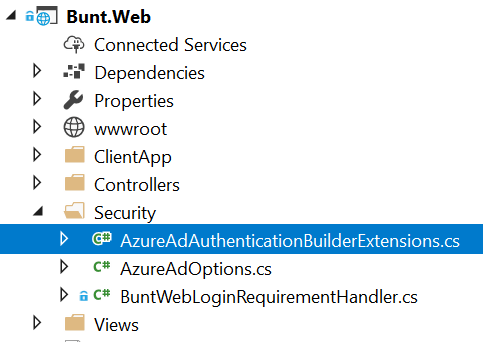
|  |
| --- |
| …  using Microsoft.AspNetCore.Authentication.Cookies;  using Microsoft.AspNetCore.Authentication.OpenIdConnect;  using Microsoft.AspNetCore.Authentication;  using Microsoft.AspNetCore.Authorization;  using Microsoft.AspNetCore.Mvc.Authorization;  using Bunt.Core.Security;  using Microsoft.AspNetCore.Http;  using Bunt.Web.Security;  … |

Ändra i klassen Startup.cs och metoden ConfigureServices till nedan konfigurering:

|  |
| --- |
| public void ConfigureServices(IServiceCollection services)  {  var logger = new LoggerConfiguration()  .WriteTo.MSSqlServer(Configuration.GetConnectionString("BuntDb"), "Log")  .Enrich.FromLogContext()  .CreateLogger();  services.AddAuthentication(sharedOptions =>  {  sharedOptions.DefaultScheme = CookieAuthenticationDefaults.AuthenticationScheme;  sharedOptions.DefaultChallengeScheme = OpenIdConnectDefaults.AuthenticationScheme;  })  .AddAzureAd(options => Configuration.Bind("AzureAd", options))  .AddCookie();  services.AddSingleton<ILogger>(logger);  services.AddTransient(typeof(IPipelineBehavior<,>), typeof(LoggingBehavior<,>));  services.AddTransient(typeof(IPipelineBehavior<,>), typeof(AccessBehavior<,>));  services.AddTransient<IConnectionFactory>(provider => new SqlConnectionFactory(Configuration.GetConnectionString("BuntDb")));  services.AddDbContext<BuntDbContext>(options => options.UseSqlServer(Configuration.GetConnectionString("BuntDb")));  services.AddMediatR(typeof(ListaBuntladeStallen.Handler));  services.AddTransient<IAuthorizationHandler, BuntWebLoginRequirementHandler>();  services.AddTransient(provider => provider.GetService<IHttpContextAccessor>().HttpContext.User);  services.AddTransient<ICurrentUser, ClaimsPrincipalCurrentUser>();  services.AddTransient<IAccessClient>(provider => new HardcodedAccessClient(new AccessUser { UserId = 0 }));  services.AddMvc(options =>  {  var builder = new AuthorizationPolicyBuilder()  .AddRequirements(new BuntWebLoginRequirement())  .RequireAuthenticatedUser();  options.Filters.Add(new AuthorizeFilter(builder.Build()));  });  } |

För att hantera konfigureringen av AzureAD används en extension-metod på AuthenticationBuilder:  
 .AddAzureAd(options => Configuration.Bind("AzureAd", options))

Undersök hur denna ser ut i projektet Bunt.Web under Security/AzureAdExtensions och AzureAdAuthenticationBuilderExtensions.cs.



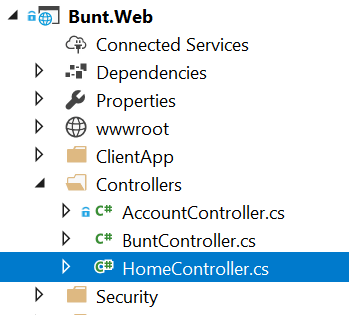
Lägg till konfiguration för Azure AD Options i appsettings.json

|  |
| --- |
| {  "AzureAd": {  "Instance": "https://login.microsoftonline.com/",  "Domain": "bringcitymail.onmicrosoft.com",  "TenantId": "822f2ac5-9222-40f7-80d6-6b7a3b1fb8b4",  "ClientId": "0e06c027-ce7c-4686-afe3-5668cac9045c",  "CallbackPath": "/signin-oidc"  },  "Logging": {  "IncludeScopes": false,  "Debug": {  "LogLevel": {  "Default": "Warning"  }  },  "Console": {  "LogLevel": {  "Default": "Warning"  }  }  }  } |

Lägg till app.UseAuthentication i Configure-metoden i Startup.cs:

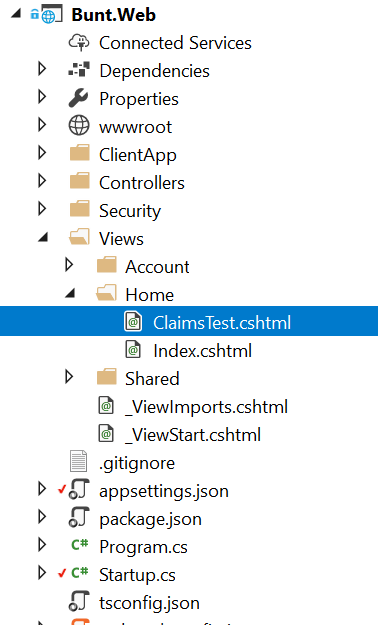
|  |
| --- |
| public void Configure(IApplicationBuilder app, IHostingEnvironment env)  {  if (env.IsDevelopment())  {  app.UseDeveloperExceptionPage();  app.UseWebpackDevMiddleware(new WebpackDevMiddlewareOptions  {  HotModuleReplacement = true  });  }  else  {  app.UseExceptionHandler("/Home/Error");  }  app.UseStaticFiles();  app.UseAuthentication();  app.UseMvc(routes =>  {  routes.MapRoute(  name: "default",  template: "{controller=Home}/{action=Index}/{id?}");  routes.MapSpaFallbackRoute(  name: "spa-fallback",  defaults: new { controller = "Home", action = "Index" });  });  }  } |

Lägg till en ny MVC Action i Home Controller ’ClaimsTest’ – och lägg till injicering av ICurrentUser



|  |
| --- |
| using System.Diagnostics;  using Microsoft.AspNetCore.Mvc;  using System.Collections.Generic;  using System.Security.Claims;  using System.Linq;  using Bunt.Core.Security;  namespace Bunt.Web.Controllers  {  public class HomeController : Controller  {  private readonly ICurrentUser \_currentUser;  public HomeController(ICurrentUser currentUser)  {  \_currentUser = currentUser;  }  public IActionResult Index()  {  return View();  }  public IActionResult Error()  {  ViewData["RequestId"] = Activity.Current?.Id ?? HttpContext.TraceIdentifier;  return View();  }  // GET: /Home/ClaimsTest  public ActionResult ClaimsTest()  {  var claims = \_currentUser.Claims;  return View(claims);  }  }  } |

Lägg till en ny MVC View - ClaimsTest.cshtml - under Views/Home-katalogen:



|  |
| --- |
| @model IEnumerable<System.Security.Claims.Claim>  @{  ViewData["Title"] = "ClaimsTest";  }  <h2>Display Claims</h2>  @if (Model == null)  {  <p class="alert-danger">No claims found</p>  }  else  {  <table class="table table-bordered">  <tr>  <th>Subject</th>  <th>Issuer</th>  <th>Type</th>  <th>Value</th>  </tr>  @foreach (var claim in Model.OrderBy(x => x.Type))  {  <tr>  <td>@claim.Subject.Name</td>  <td>@claim.Issuer</td>  <td>@Html.Raw(claim.Type)</td>  <td>@claim.Value</td>  </tr>  }  </table>  } |

Debugga och se att du nu bör få logga in genom Azure AD mot ADFS.

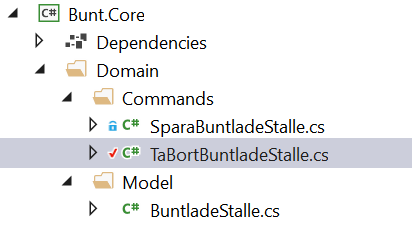
Undersök vilka ”Claims” som du får tillbaka genom vyn du skapade genom att gå till:

<http://localhost:3289/Home/ClaimsTest>

# Del 2

Lägga på attribut-baserad säkerhet på Command.

Lägg till attributet FunctionAccess med värdet TABORTBUNTLADESTALLE på Command TaBortBuntLadeStalle:



|  |
| --- |
| namespace Bunt.Core.Domain.Commands  {  [FunctionAccess("TABORTBUNTLADESTALLE")]  public class TaBortBuntladeStalle  {  public class Command : IRequest  {  public Guid Id { get; set; }  }  … |

LDebugga projektet igen och se hur AccessBehavior använder FunctionAccess-attributet för att kontrollera säkerheten.

# Utmaning

Lägg till funktionalitet som gör att knappen ”Ta bort” disablas när man inte har access till TABORTBUNTLADESTALLE.