# Implementing & Extending the Entity Framework Default Schema



Scott Brady
IDENTITY & ACCESS CONTROL LEAD

@scottbrady91 www.scottbrady91.com

#### Overview



Simplifying apps using the default schema

Quick & easy data access with the default user store

When & how to customize the default schema



# ASP.NET Identity Package Structure

Microsoft. Extensions. Identity. Core

Microsoft.AspNetCore.Identity

Microsoft. Extensions. Identity. Stores

Microsoft.AspNetCore.Identity .EntityFrameworkCore



public class IdentityUser : IdentityUser<string>

public class IdentityUser<TKey> where TKey : IEquatable<TKey>

IdentityUser

Designed to use every feature of ASP.NET Core Identity



```
string UserName
string NormalizedUserName
string Email
string NormalizedEmail
bool EmailConfirmed
string PasswordHash
string SecurityStamp
string PhoneNumber
bool PhoneNumberConfirmed
bool TwoFactorEnabled
DateTime? LockoutEndDateUtc
bool LockoutEnabled
int AccessFailedCount
```

**◄** IUserEmailStore

- IUserPasswordStore
- IUserSecurityStampStore
- IUserPhoneNumberStore

- IUserTwoFactorStore
- IUserLockoutStore



# IdentityUserClaim

```
public class IdentityUserClaim<TKey> {
   public int Id { get; set; }
   public string ClaimType { get; set; }
   public string ClaimValue { get; set; }
   public TKey UserId { get; set; }
   public Claim ToClaim() { ... }
```

## IdentityUserLogin

```
public class IdentityUserLogin<TKey> {
   public string LoginProvider { get; set; }
   public string ProviderKey { get; set; }
   public string ProviderDisplayName { get; set; }
   public TKey UserId { get; set; }
}
```



## IdentityUserToken

```
public class IdentityUserToken<TKey> {
   public string Name { get; set; }
   public string Value { get; set; }
   public string LoginProvider { get; set; }
   public TKey UserId { get; set; }
}
```



# IdentityRole

```
public class IdentityRole<TKey> {
    public TKey Id { get; set; }
    public string Name { get; set; }
    public string NormalizedName { get; set; }
}
public class IdentityRole : IdentityRole<string> { ... }
```



# IdentityRoleClaim

```
public class IdentityRoleClaim<TKey> {
   public int Id { get; set; }
   public string ClaimType { get; set; }
   public string ClaimValue { get; set; }
   public TKey RoleId { get; set; }
   public Claim ToClaim() { ... }
```

# IdentityUserRole

```
public class IdentityUserRole<TKey> {
   public TKey UserId { get; set; }
   public TKey RoleId { get; set; }
}
```



# Stop trying to make ApplicationUser a thing



# Simplifying Custom User Stores



```
public abstract class UserStoreBase<TUser, TRole, TKey,
TUserClaim, TUserRole, TUserLogin, TUserToken, TRoleClaim>
{
}
```

#### UserStoreBase

Implements all IUserStore interfaces (IUserRoleStore optional)

All get/set methods completed for IdentityUser

Some additional helper methods



# Demo



Implementing UserStoreBase



# Entity Framework Core



#### DbContext



IdentityUserContext (Users only)



IdentityDbContext (Users & roles)



# |IdentityDbContext

```
IdentityDbContext<TUser, TRole, TKey, TUserClaim, TUserRole,
TUserLogin, TRoleClaim, TUserToken>
where TUser : IdentityUser<TKey>
where TRole : IdentityRole<TKey>
where TKey : IEquatable<TKey>
where TUserClaim : IdentityUserClaim<TKey>
where TUserRole : IdentityUserRole<TKey>
where TUserLogin : IdentityUserLogin<TKey>
where TRoleClaim : IdentityRoleClaim<TKey>
where TUserToken: IdentityUserToken<TKey>
```



# Stores



UserOnlyStore



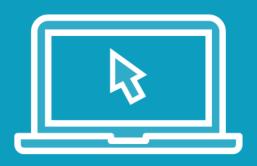
UserStore



RoleStore



#### Demo



Implementing the Entity Framework Core default user store

**Database migrations** 



# Extending IdentityUser



# Claim or Property?

Claim

**Property** 

Simple structure

Searchable (slow)

Flexible

**Complex structure** 

Searchable (fast)

Indexable



#### Demo



**Extending the Entity Framework Core** default user and DbContext



# Summary



Simplifying apps using the default schema

Quick & easy data access with the default user store

When & how to customize the default schema

