Microsoft.AspNet.Identity.UserManager<TUser, TKey>

where TUser : class, IUser<TKey>

where TKey : IEquatable<TKey>

{

//构造函数，传入IUserStore

public UserManager(IUserStore<TUser, TKey> store);

protected internal IUserStore<TUser, TKey> Store { get; set; }

Virtual UserValidator、PasswordValidator、CreateAsync、UpdateAsync、

AddToRoleAsync等

调用IUserStore的方法;

调用IXXXStore接口方法：

IUserRoleStore<TUser, TKey> store = this.Store as IUserRoleStore<TUser, TKey>;

}

//继承Microsoft.AspNet.Identity.IXXXStore接口

public abstract class AbpUserStore<TRole, TUser> :

IUserStore<TUser, long>,

IUserPasswordStore<TUser, long>,

IUserEmailStore<TUser, long>,

IUserLoginStore<TUser, long>,

IUserRoleStore<TUser, long>,

IQueryableUserStore<TUser, long>,

IUserLockoutStore<TUser, long>,

IUserPermissionStore<TUser>,

IUserPhoneNumberStore<TUser, long>,

IUserClaimStore<TUser, long>,

IUserSecurityStampStore<TUser, long>,

IUserTwoFactorStore<TUser, long>,

{

XXXRepository;

调用XXXRepository访问数据库

}

public class UserStore : AbpUserStore<Role, User>

{

}

public class UserManager : AbpUserManager<Role, User>

{

}

public abstract class AbpUserManager<TRole, TUser>

: UserManager<TUser, long>,

IDomainService

where TRole : AbpRole<TUser>, new()

where TUser : AbpUser<TUser>

{

在构造函数中传入AbpUserStore<TRole, TUser> AbpStore

重载或调用实现上述方法

}

[Table("AbpUsers")]

abstract AbpUserBase : FullAuditedEntity<long>, IMayHaveTenant, IPassivable

{

UserName、TenantId、EmailAddress、Name、……

}

AbpUser<TUser> : AbpUserBase, Microsoft.AspNet.Identity.IUser<long>()

{

DeleterUser、CreatorUser、LastModifierUser

}

User : AbpUser<User>

{

}