**зарегистрировать**

var user = new AppUser() { UserName = model.UserName };

var result = await UserManager.CreateAsync(user, model.Password);

**Войти**

ClaimsIdentity ident = await UserManager.CreateIdentityAsync(user, DefaultAuthenticationTypes.ApplicationCookie);

AuthManager.SignOut();

AuthManager.SignIn(new AuthenticationProperties() { IsPersistent = true }, ident);

*Или*

SignInStatus result = await SignInManager.PasswordSignInAsync(model.Email, model.Password, model.RememberMe, shouldLockout: false);

**Выйти**

AuthManager.SignOut();

**Проверка пользователя для двукратной аутентификации** (Get the user id that has been verified already or null в SignInManager.PasswordSignInAsync или AuthManager.SignIn)

string userId = await SignInManager.GetVerifiedUserIdAsync();

**Получение имен валидных способов двукратной аудентификации (**Returns a list of valid two factor providers for a user**)**

IList<string> userFactors = await UserManager.GetValidTwoFactorProvidersAsync(userId);

**Отправить код пользователю (двукратная аудентификация)** Send a two factor code to a user

bool isSend = await SignInManager.SendTwoFactorCodeAsync(model.SelectedProvider)

**Имеет пользователь был проверен (т.е. либо с помощью пароля или логина внешнего)**

**(двукратная аудентификация)**

Has the user been verified (ie either via password or external login)

bool isVerified = await SignInManager.HasBeenVerifiedAsync()

**Войти при помощи двукратной аудентификации**

The following code protects for brute force attacks against the two factor codes.

If a user enters incorrect codes for a specified amount of time then the user account will be locked out for a specified amount of time. You can configure the account lockout settings in IdentityConfig

SignInStatus result = await SignInManager.TwoFactorSignInAsync(model.Provider, model.Code, isPersistent: model.RememberMe, rememberBrowser: model.RememberBrowser);

**Генерация ключа для подтверждения по почте**

string code = await UserManager.GenerateEmailConfirmationTokenAsync(user.Id);