Startup(es el que levanta el archivo)

Como es el que inicia la aplicación se puede crear roles del administrador y otros desde el inicio, esto se hace utilizando al roleManager, por ejemplo

if (!roleManager.RoleExists("Admin"))

{

//Crear el rol para el superAdmin

var role = new IdentityRole();

role.Name = "Admin";

roleManager.Create(role);

//Crear el rol para el superAdmin

var user = new ApplicationUser();

user.UserName = "Cliente";

user.Email = "cliente@hola.com";

string userPWD = "cliente123";

var chkUser = UserManager.Create(user, userPWD);

//Añadir el id del usuario al rol

if (chkUser.Succeeded)

{

var result1 = UserManager.AddToRole(user.Id, "Admin");

}

}

Si yo quiero que los usuarios puedas seleccionar sus roles debo crearlos desde el principio en el archivo startup

if (!roleManager.RoleExists("Ventas"))

{

var role = new IdentityRole();

role.Name = "Ventas";

roleManager.Create(role);

}

if (!roleManager.RoleExists("Compras"))

{

var role = new IdentityRole();

role.Name = "Compras";

roleManager.Create(role);

}

if (!roleManager.RoleExists("Clientes"))

{

var role = new IdentityRole();

role.Name = "Clientes";

roleManager.Create(role);

}

Y entonces hay que modificar la vista correspondiente al registro incluyendo radiobutton o un dropdown list para el userRoles.

Debo modificar el AccountViewModel para incluir el atributo de userRoles

<div class="form-group">

@Html.Label("Rol de usuario", new { @class = "col-md-2 control-label" })

<div class="col-md-10">

@Html.DropDownListFor(m => m.UserRoles, new List<SelectListItem> {

new SelectListItem { Value = "Ventas", Text = "Ventas"},

new SelectListItem { Value = "Compras", Text = "Compras"},

new SelectListItem { Value = "Clientes", Text = "Clientes"}

}, new { @class="myselect"})

</div>

</div>

Hay que crear un contexto para la aplicación en el controlador AccountController

ApplicationDbContext context;

public AccountController()

{

context = new ApplicationDbContext();

}

Para desplegar si usé un dropdownlist debo modificar el método register(GET) de mi clase AccountController y colocar en un ViewBag un selectlist de todos los roles exceptuando el del administrador

Lo mismo debo hacer del lado del método register POST e incluir en el user del ApplicationUser mi atrobuto userRoles

var user = new ApplicationUser { UserName = model.Email, Email = model.Email, userRole = model.userRole };

Y agregar la espera por la llamada al método de añadir roles incluyendo el atributo de UserRoles

await this.UserManager.AddToRoleAsync(user.Id,model.userRole);

Ahora en las vistas del modelo, si yo quiero que mi vista de login en lugar de pedirme el correo quiero que me pida el nombre del usuario, esto es cuando no quiero que el login sea por correo, debo agregar en el loginViewModel el atributo UserName y en la correspondiente vista de login en lugar de pedir el Email pedir el UserName

También modificar en el AccountController en el método login por (POST) hacer el cambio de Email por UserName