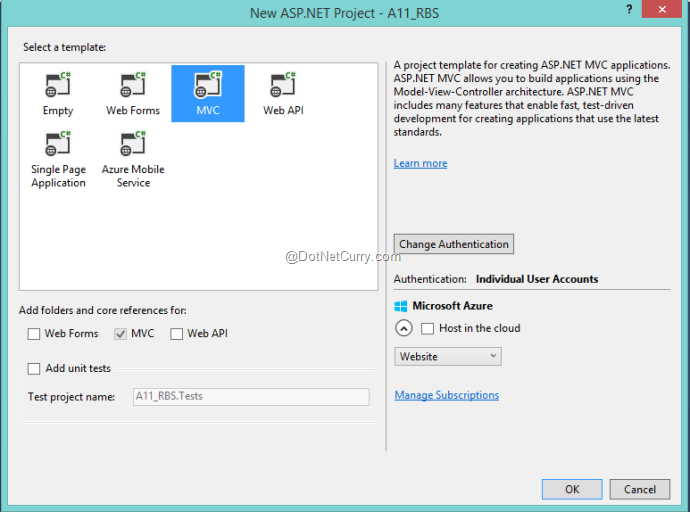
## Sử dụng chuỗi connection chung có nam là default.. để khi đăng ký thì tự database sẽ sinh ra được những bảng liên quan

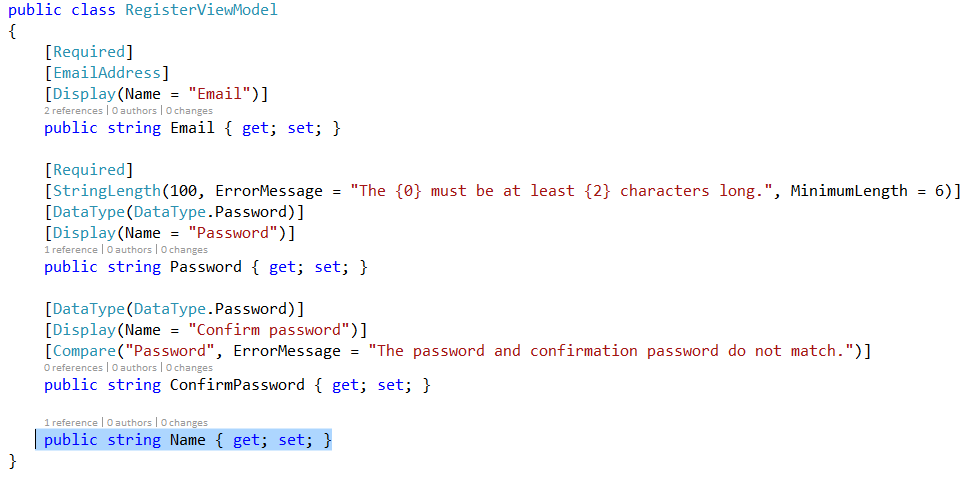
## How to use role-based security

* **Step 1**: Create a new ASP.NET MVC

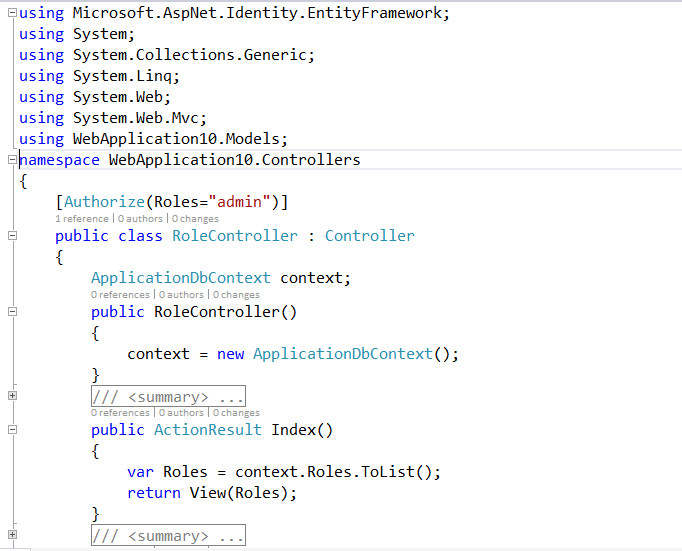


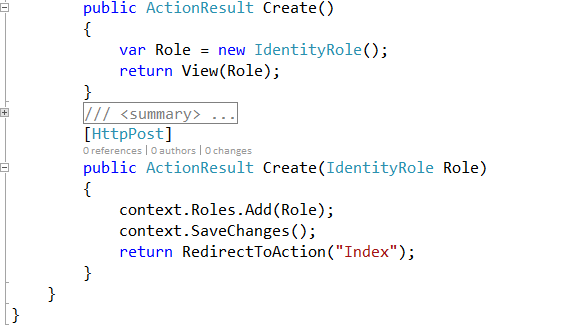
* **Step 2**: Add following code in *RegisterViewModel***class, this is used to add role name**

public string Name { get; set; }



* **Step 3**: Add new Empty Controller class and name *RoleController, add the code as given below*





**Role creation is done using *IdentityRole* class. This class provides properties e.g. Id, Name, etc for creating roles for the applications.**

* **Step 4**: Create views for RoleController
* **Index.cshtml**

|  |
| --- |
| @model IEnumerable<Microsoft.AspNet.Identity.EntityFramework.IdentityRole>  @{      ViewBag.Title = "Index";  }    <h2>Available Roles For Application</h2>    @Html.ActionLink("Create Role","Create","Role")    <style type="text/css">      #tbrole,.c {      border:double;      }  </style>    <table id="tbrole">      <tr>          <td class="c">              Role Name          </td>      </tr>      @foreach (var item in Model)      {          <tr>              <td class="c">                  @item.Name              </td>          </tr>      }  </table> |

* **Create.cshtml**

|  |
| --- |
| @model Microsoft.AspNet.Identity.EntityFramework.IdentityRole  @{      ViewBag.Title = "Create";  }  <h2>Create</h2>    <style type="text/css">      #tbrole, .c {          border: double;      }  </style>    @using (Html.BeginForm())  {      <table id="tbrole">          <tr>              <td class="c">Enter Role Name To be Created:</td>              <td class="c">                  @Html.EditorFor(m => m.Name)              </td>          </tr>      </table>      <input type="submit" value="Create Role" />  } |

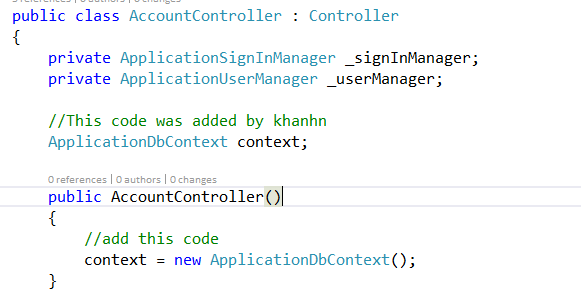
* **Step 5**: Open **AccountController** and ***add following code***

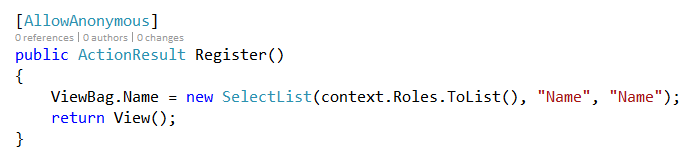
|  |
| --- |
| ApplicationDbContext context;      public AccountController()  {      context = new ApplicationDbContext();  } |

Add the following code in the *Register()*action method:

|  |
| --- |
| [AllowAnonymous]  public ActionResult Register()  {      ViewBag.Name = new SelectList(context.Roles.ToList(), "Name", "Name");      return View();  } |

The result will look like this





Open **Register.cshtml** view in the **Account** sub folder of Views folder and **add the following Html Helper in it above the *Submit* button.**

<!--Select the Role Type for the User-->

<div class="form-group">

    @Html.Label("Select Your User Type", new { @class = "col-md-2 control-label" })

    <div class="col-md-10">

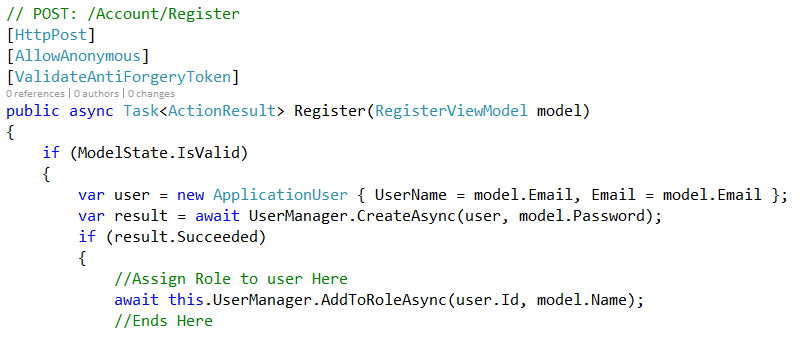
        @Html.DropDownList("Name")

    </div>

</div>

<!--Ends Here-->

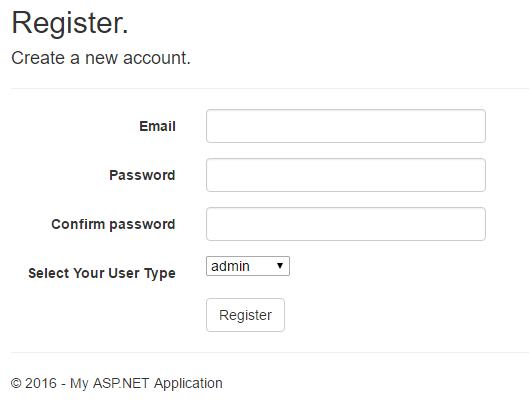
To complete the operation of Assigning Role to the user, change the *Register()*action method with *HttpPost*in the AccountController as shown below:



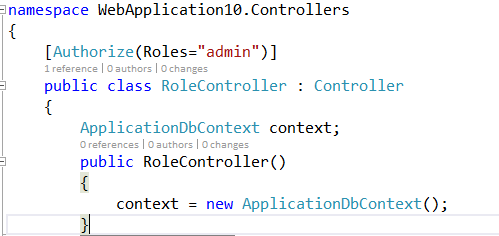
* **Step 6**: In the \_Layout.cshtml view of the Shared sub folder of the Views folder, add the following piece of code for showing link for Role in a <div> element with the class *navbar-collapse collapse.*

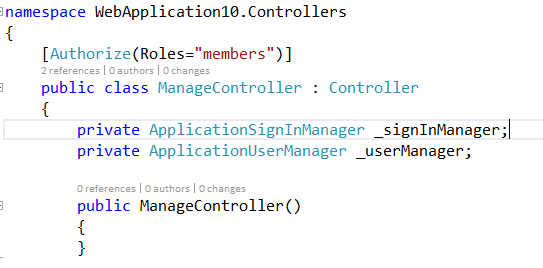
|  |
| --- |
| <li>@Html.ActionLink("Role", "Index", "Role")</li> |

* **Step 7:**Run the Application, navigate to the Create View for the **RoleController** and create ***Manager***and ***Sales Executive* roles**. Now navigate to the **Register** Action of the Account Controller and create users with roles:



* Thiết lập Authorize cho các Controller





**Step 8**:

1. AddUserToRole.cshtml

@{

ViewBag.Title = "AddUserToRole";

}

<h2>AddUserToRole</h2>

@using (Html.BeginForm("AddUserToRole", "Role", FormMethod.Post, new { @class = "form-horizontal", role = "form" }))

{

@Html.AntiForgeryToken()

<h4>Create a new account.</h4>

<hr />

@Html.ValidationSummary("", new { @class = "text-danger" })

<div class="form-group">

@Html.Label("Select User:", new { @class = "col-md-2 control-label" })

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

@Html.DropDownList("UserName", ViewBag.UserName as SelectList, new { @class = "form-control" })

</div>

</div>

<!--Select the Role Type for the User-->

<div class="form-group">

@Html.Label("Select role:", new { @class = "col-md-2 control-label" })

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

@Html.DropDownList("Name", ViewBag.Name as SelectList, new { @class = "form-control" })

</div>

</div>

<!--Ends Here-->

<div class="form-group">

<div class="col-md-offset-2 col-md-10">

<input type="submit" class="btn btn-default" value="Assign" />

</div>

</div>

}

@section Scripts {

@Scripts.Render("~/bundles/jqueryval")

}

2. RolesController.cs class

public class RoleController : Controller

{

ApplicationDbContext context;

public RoleController()

{

context = new ApplicationDbContext();

}

public ActionResult X()

{

return View();

}

public ActionResult Index()

{

var Roles = context.Roles.ToList();

// return View();

return View(Roles);

// return View("ViewName");

// return View("ViewName",Roles);

}

public string GetRoles(){

return "Admin, Managers";

}

[HttpGet]

public ActionResult AddUserToRole()

{

using (var context = new ApplicationDbContext())

{

var users = from u in context.Users

select u;

// ViewBag.Users = new SelectList(users, "Id", "UserName");

ViewBag.UserName = users.Select(x => new SelectListItem {Text = x.UserName, Value = x.Id }).ToList();

var roles = from r in context.Roles

select r;

ViewBag.Name = roles.Select(x => new SelectListItem { Text = x.Name, Value = x.Name }).ToList();

}

return View();

}

[HttpPost]

public ActionResult AddUserToRole(string UserName, string Name)

{

//var context = new AccountController();

var context = new ApplicationDbContext();

//var roleStore = new RoleStore<IdentityRole>(context);

//var roleManager = new RoleManager<IdentityRole>(roleStore);

var userStore = new UserStore<ApplicationUser>(context);

var userManager = new UserManager<ApplicationUser>(userStore);

userManager.AddToRole(UserName,Name);

//userManager.AddToRole(UserName, Name);

ViewBag.Name = UserName;

ViewBag.Role = Name;

return View("AddConfirm");

}

public ActionResult Create()

{

var Role = new IdentityRole();

return View(Role);

}

[HttpPost]

public ActionResult Create(IdentityRole Role)

{

context.Roles.Add(Role);

context.SaveChanges();

return RedirectToAction("Index");

}

}