.NET and SQL Skills – 2018-2019

Content

bigrixin@hotmail.com

One to one relationship

Solution A.

```
public class Stock
{
        [Key]
        [ForeignKey("Product")]
        public int ID { get; set; }
        public Product Product { get; set; }
}
>>>>PK_dbo.Stock
>>>>FK_dbo.Stock_dbo.Project_ID

public class Product
{
        public int ID { get; set; }
        public Stock Stock { get; set; }
}

>>>>>PK._dbo.Product
```

Solution B: (used)

```
modelBuilder.Entity<Product>().HasRequired(p => p.Stock).WithRequiredDependent(p => p.Product);

Product ==>PK_dbo.Product, FK_dbo.Product_dbo.Stock_ID
Stock   ==>PK_dbo.Stock

Add New Product:
public Product(string barcode)
{
        Barcode = barcode;
        Stock = new Stock(0, 0);
}

update: (two ways)
Product currentProduct = _readEntities.Get<Product>(a => a.ID == stockVM.ProductID).SingleOrDefault();
if (currentProduct != null)
{
        Stock stock = _readEntities.Get<Stock>(a => a.Product.ID == currentProduct.ID).SingleOrDefault();
        stock.Price = stockVM.Price;
        stock.Quantity = stockVM.Quantity;
        currentProduct.Stock = stock;
        _writeEntities.Update(currentProduct);

        //Stock stock = _readEntities.Get<Stock>(a => a.Product.ID == currentProduct.ID).SingleOrDefault();
        //stock.Price = stockVM.Price;
        //stock.Quantity = stockVM.Quantity;
        //_writeEntities.Update(stock);
        _writeEntities.Save();
```


bigrixin@hotmail.com

```
}
```

DropDownList select and view value

```
//ViewModel
public IEnumerable<SelectListItem> CategoryDropDownList { get; set; }

//Controler
public SelectList GetCategorySelectList()
{
        return new SelectList(
         _readEntities.Get<Category>().ToList(),
        "ID",
        "Classification");
}

var CategoryDropDownList=GetCategorySelectList();


//Create, Update

<div class="form-group">
        @Html.LabelFor(model => model.BrandID, htmlAttributes: new { @class = "control-label col-md-2" })
        <div class="col-md-8">
                @Html.DropDownListFor(model => model.BrandID, Model.BrandDropDownList, new { @class = "form-
control" })
                @Html.ValidationMessageFor(model => model.BrandID, "", new { @class = "text-danger" })
        </div>
</div>

//View
@Model.BrandDropDownList.Single(a => a.Value == Model.BrandID.ToString()).Text
```

Change migration from Auto to manual

```
1. Move Migrations folder from SmartRP to SmartRP.Infrastructure.Data and .Auth
2. Change AutomaticMigrationsEnabled from true to false
3. namespace SmartRP.Infrastructure.Auth.Migrations
  add ContextKey = "AspNetIdentityDbContext";
4. At SmartRP.Infrastructure.Data.Migrations
  Change AspNetIdentityDbContext to SmartRPDbContext
```

Generate first migration and tables:

```
Open Package Manager Console in VS

select SmartRP.Infrastructure.Auth as Default project
PM>
```

add-migration InitialCreate ==> (generate 201802080021006_InitialCreate.cs)
update-database ==> (create asp.net tables)

select SmartRP.Infrastructure.Data as Default project

add-migration InitialCreate
update-database ==> (create tables)

(-force) is used to Re-scaffolding migration 'InitialCreate'.
A previous migration called 'InitialCreate' was already applied to the target database. If you meant to re-scaffold 'InitialCreate', revert it by running 'Update-Database -TargetMigration $InitialDatabase', then delete '201802060819006_InitialCreate1.cs' and run 'Add-Migration InitialCreate' again.

If want to rebuild InitialCreate,please follow below steps:
1. delete database in SQL Server
2. delect *******_InitialCreate in VS
3. check .csproj which in project folder:
SmartRP.Infrastructure.Auth.csproj
SmartRP.Infrastructure.Data.csproj

Visual Studio Team Services

https://thefirstteam.visualstudio.com/_projects

VS ->Publish to Visual Studio Team Services->(login)->Publish repository

Use continuous integration (CI):
https://thefirstteam.visualstudio.com/SmartRP2.02
->Set up Build -> VSTS Git, Team project, Repository ->Continue->ASP.NET->Apply
->select Build solution->Save & queue
-->Triggers->Enable continuous integration
->Schduled->Add

Azure add manage role for a subscription

https://docs.microsoft.com/en-us/azure/billing/billing-add-change-azure-subscription-administrator

IEnumerable and List

IEnumerable to List
```
IEnumerable<int> enumerable = Enumerable.Range(1, 300);
List<int> asList = enumerable.ToList();
```

List to IEnumerable
```
public IEnumerable<Book> GetBooks()
{
   List<Book> books = FetchEmFromSomewhere();
```

```
    return books;
}
```

Using DynamicViewBag

```
// -- Controller
public class MyAccountController : Controller
{
        public ActionResult _StudentMenu()
        {
                string userType = this._userService.GetCurrentRole(User.Identity.Name);
                DynamicViewBag model = this._projectService.GetCurrentMyRequestsCountModel(User.Identity.Name,
userType);
                return PartialView("_StudentMenu", model);
        }
}

public DynamicViewBag GetCurrentMyRequestsCountModel(string currentUserID, string userType)
{
        int myRequestProjects = 0;
        int myRequestProjectsRespondCount = 0;
        if (userType == "Student")
        {
                var student = this._readEntities.Get<Student>(a => a.Email == currentUserID).FirstOrDefault();
                var now = DateTime.Now;
                var currentTerm = this._readEntities.Get<Term>().Where(a => a.StartAt <= now && now <=
a.EndAt).SingleOrDefault();
                var myRequests = this._readEntities.Get<JoinProjectGroup>(a => a.TermId == currentTerm.ID &&
a.StudentId == student.ID);
                myRequestProjectsRespondCount = myRequests.Where(a => a.RequestStatus ==
RequestStatus.Accepted || a.RequestStatus == RequestStatus.Rejected).Count();
                myRequestProjects = myRequests.Count();

        }
        DynamicViewBag model = new DynamicViewBag();
        model.AddValue("MyRequestProjects", myRequestProjects);
        model.AddValue("MyRequestProjectsRespondCount", myRequestProjectsRespondCount);
        return model;
}

//MyCurrentProject.cshtml  --View
@Html.Action("_StudentMenu", "MyAccount")

//_StudentMenu.cshtml     --Menu
<div class="nav nav-sidebar">
        <a href="@Url.Action( "MyRequestedList", "Projects", null)" title="Request to join project group list" class="btn
btn-block btn-success btn-lg fa fa-comment">

        My Request Status
```


bigrixin@hotmail.com

```
        @if (Model.MyRequestProjects != 0)
        {
        <span class="badge" style="background-color:#ccfef8"> @Model.MyRequestProjects</span>
        }
        else if (Model.MyRequestProjectsRespondCount != 0)
        {
        <span style="color:#ff6a00">
                <i class="fa fa-circle" aria-hidden="true"></i>
        </span>
        }
  </a>
</div>
```

Open a page in new tab

```
@Html.ActionLink(item.Title, "Details", "projects", new { projectID = item.ID }, htmlAttributes: new
{ title = @Html.DisplayFor(modelItem => item.Description), target = "_blank" })
```

Close current tab (window)

```
<button onclick="self.close()" class="btn btn-default">Close this window</button>
```

Go back to history page

- It does not work with opened new tab

```
<button onclick="goBack()" class="btn btn-default">Go Back</button>
<script>
        function goBack() {
                if (document.referrer == "") {
                        window.close()
                } else {
                        window.history.go(-1);  //.back();
                        return false;
                }
        }
</script>
```

JQuery date issues-The field Start Date must be a date

```
<script>
   $(document).ready(function () {
        jQuery.validator.methods.date = function (value, element) {
        var isfirefox = navigator.userAgent.indexOf('Firefox') > -1;
        var isChrome = /Chrome/.test(navigator.userAgent) && /Google Inc/.test(navigator.vendor);
        var isSafari = /Safari/.test(navigator.userAgent) && /Apple Computer/.test(navigator.vendor);
        if (isfirefox||isSafari || isChrome) {
                var d = new Date();
                return this.optional(element) || !/Invalid|NaN/.test(new Date(d.toLocaleDateString(value)));
        }
         else {
                return this.optional(element) || !/Invalid|NaN/.test(new Date(value));
        }
        };
```


bigrixin@hotmail.com

```
        });
</script>
```

UserManager always null

Add code is below red color, get a confirmation Email. If still cannot login , reset password.

```
var user = await UserManager.FindAsync(model.Email, model.Password);
if (user == null)
        user = await UserManager.FindByEmailAsync(model.Email);
if (user != null)
{
        if (!await UserManager.IsEmailConfirmedAsync(user.Id))
                return RedirectToAction("ResendEmail", "Account", new { Email = model.Email });
        AddNewUser(user.Id);
}
…….

default:

        if (user != null)
        {
                bool passhash = false;
                passhash = await UserManager.CheckPasswordAsync(user, model.Password);
                if (passhash)
                        return RedirectToLocal(returnUrl);
        }
        ModelState.AddModelError("", "Invalid login attempt.");
        return View(model);
```

Linq samples

https://code.msdn.microsoft.com/101-LINQ-Samples-3fb9811b

Open Elmah

<<Web.config.>> Add below:

```
 <elmah>
        <errorLog type="Elmah.SqlErrorLog, Elmah" connectionStringName="MyAbilityFirstDbContext" />
         <!-- allowRemoteAccess - Allow remotely connected users to access the elmah page. -->
        <security allowRemoteAccess="false" />
 </elmah>
```

<<app_settings.config>> Change below:

bigrixin@hotmail.com

```
<!-- Turn on/off ELMAH.MVC handler. -->
<add key="elmah.mvc.disableHandler" value="true" />

<!-- Secure the elmah route with authentication. -->
<add key="elmah.mvc.requiresAuthentication" value="true" />

<!-- In case of authentication is turned on, you can specify exact roles of user that have access (e.g. "Administrator"). -->
<add key="elmah.mvc.allowedRoles" value="admin" />
```

```
//in factor, not need following code
namespace MyAbilityFirst.Infrastructure.Data.Migrations
{
using System.Data.Entity.Migrations;
using System.IO;
using System.Linq;
using System.Reflection;

public partial class AddElmahObjects : DbMigration
{
        public override void Up()
        {
                var assembly = Assembly.GetExecutingAssembly();
                var elmahScript =
                        assembly
                                .GetManifestResourceNames()
                                .Where(x => x.StartsWith("MyAbilityFirst.Infrastructure.Data.Sql.Elmah.SqlServer.sql"));

                if (elmahScript.Any())
                {
                        var resourceName = elmahScript.FirstOrDefault();
                        using (Stream stream = assembly.GetManifestResourceStream(resourceName))
                        using (StreamReader reader = new StreamReader(stream))
                        {
                                string scriptText = reader.ReadToEnd();
                                Sql(scriptText);
                        }
                }
        }

        public override void Down()
        {
                DropStoredProcedure("dbo.ELMAH_GetErrorsXml");
                DropStoredProcedure("dbo.ELMAH_GetErrorXml");
                DropStoredProcedure("dbo.ELMAH_LogError");
                DropIndex("dbo.ELMAH_Error", "dbo.IX_ELMAH_Error_App_Time_Seq");
                DropTable("dbo.ELMAH_Error");
        }
}
}
```

PagedList with customer ID

```
@using (Html.BeginForm("EnrolledStudents", "Terms", FormMethod.Get))
{
        <input type="hidden" name="id" value="@ViewBag.SubjectID" />
        @Html.TextBox("SearchString", ViewBag.CurrentFilter as string, new { @placeholder = "Search for Firstname or Lastname or Email or Student ID", @style = "width:450px" })
        <button type="submit" value="Search" class="btn btn-normal btn-sm"> <span class="fa fa-search" aria-hidden="true"></span> Search </button>
}

Page @(Model.PageCount < Model.PageNumber ? 0 : Model.PageNumber) of @Model.PageCount

@Html.PagedListPager(Model, page => Url.Action("EnrolledStudents",
        new { ViewBag.SubjectID, page, currentFilter = ViewBag.CurrentFilter }))


[HttpGet, Route("terms/enrolled-students")]
public ActionResult EnrolledStudents(int? id, string currentFilter, string searchString, int? page)
{

}
```

MS-SQL Server Management Studio - Export scripts

Select database->right mouse->Generate scripts for database objects->

select table->Advanced->Types of data to script **[Data only]**

Azure SQL database import and export

https://mikhail.io/2016/10/azure-sql-databases-backups-disaster-recovery-import-export/

SQL Server Management Studio, connect to Azure SQL Database,

A. Right-click and select **Generate and Publish Scripts** -> Next ->select Tables->Next->Advanced->
Types of data to script [Schema and data]->OK->Next=>finish

B. Right-click and select Tasks -> Export Data tier application in the menu.
   ====>Advanced-->Select all table
   >>>>>Save to local disk [.backpac]
   >>>>>Save to Windows Azure [Storage]

Fix error “The field Start Date must be a date.”

```
@Convert.ToString(string.Format("{0:dd/MM/yyyy}", Model.StartAt)),
```

http://www.itorian.com/2015/04/the-field-date-must-be-date-error-in.html

Step 1: Create a new jquery.validate.date.js file.

Step 2: Use following code inside that file.

```
$(function () {
"use strict"

   $.validator.methods.date = function (value, element) {
      if ($.browser.webkit) {
         var d = new Date();
         return this.optional(element) || !/Invalid|NaN/.test(new Date(d.toLocaleDateString(value)));
      }
      else {
         return this.optional(element) || !/Invalid|NaN/.test(new Date(value));
      }
   };
});
```

Step 3: Now ensure that it is loaded after jquery.validate.js file, look at this:

```
<script src="@Url.Content("~/Scripts/jquery.validate.date.js")" type="text/javascript"></script>
```

>>> browser.webkit may can not find
or

Other solution is to modify jquery.validate.js by finding the function date: function (value, element) and put this code in it:

```
if ($.browser.webkit) {
   //ES - Chrome does not use the locale when new Date objects instantiated:
   var d = new Date();
   return this.optional(element) || !/Invalid|NaN/.test(new Date(d.toLocaleDateString(value)));
}
else {
   return this.optional(element) || !/Invalid|NaN/.test(new Date(value));
}
```

Check brower

```
$(document).ready(function(){

   /* Get browser */
   $.browser.chrome = /chrome/.test(navigator.userAgent.toLowerCase());

   /* Detect Chrome */
   if($.browser.chrome){
      /* Do something for Chrome at this point */
      /* Finally, if it is Chrome then jQuery thinks it's
```

```
        Safari so we have to tell it isn't */
      $.browser.safari = false;
   }

   /* Detect Safari */
   if($.browser.safari){
      /* Do something for Safari */
   }

});
```

Visual Studio Slow

1. Go to **Tools -> Options -> Text Editor -> C# -> Advanced** and uncheck **Enable full solution analysis**
2. Go to **Tools -> Options -> XAML Designer -> General** and uncheck **Enable XAML Designer (requires restart of Visual Studio)**
3. Go to **Tools -> Options -> Text Editor -> All Languages** and uncheck **Enable CodeLens**
4. Go to **Tools -> Options -> Debugging -> General** and disable **Enable Diagnostic Tools while debugging**

- Set Current source control ⋯ to None under Tools → Options → Source Control.
- Uncheck Synchronized settings across ... option under Tools → Options → Environment → Synchronized Settings.
- Disable CodeLens (Optional): Uncheck Enable CodeLens option under Tools → Options → Text Editor → All Languages.
- Disable Diagnostic Tools (Optional): Uncheck Enable Diagnostic Tools while debugging option under Tools → Options → Debugging → General.
- Clean the content in WebSiteCache folder
  - C:\Users\%USERNAME%\AppData\Local\Microsoft\WebSiteCache
- Clean the content in Temporary ASP.NET Files folder
  - C:\Users\%USERNAME%\AppData\Local\Temp\Temporary ASP.NET Files
- Tools → Options → Environment → General.
- Clear the Automatically adjust visual experience based on client performance check box. (Refer to the following screen shot for this step.)
- Select or clear the Enable rich client visual experience

Controller Extension

```
public static class ControllerExtensions
{

        #region Extensions

        public static User GetLoggedInUser(this Controller controller)
        {
                if (!controller.HttpContext.Request.IsAuthenticated)
                        return null;

                var entities = DependencyResolver.Current.GetService<IReadEntities>();
                var loginIdentityId = controller.HttpContext.User.Identity.GetUserId();
                User user = null;
```


bigrixin@hotmail.com

```
            if (controller.HttpContext.User.IsInRole("Coordinator"))
                  user = entities.Single<Coordinator>(u => u.LoginIdentityID == loginIdentityId);
            else if (controller.HttpContext.User.IsInRole("Supervisor"))
                  user = entities.Single<Supervisor>(u => u.LoginIdentityID == loginIdentityId);
            return user;
      }

      #endregion

}
```

Coordinator currentCoordinator = this.GetLoggedInUser() as Coordinator;

'Microsoft.SqlServer.Types' Error

```
Spatial types and functions are not available for this provider because the assembly
'Microsoft.SqlServer.Types' version 10 or higher could not be found.
```

https://www.koskila.net/2016/06/07/solving-the-spatial-types-and-functions-are-not-available-problem/

```
Update-Package -reinstall Microsoft.SqlServer.Types
```

```
Global.asax.cs
protected void Application_Start()
{
    // Enables use of spatial data types
    SqlServerTypes.Utilities.LoadNativeAssemblies(Server.MapPath("~/bin"));

    // Other registrations...
}
```

Install "Microsoft System CLR Types for SQL Server 2012" from:

X86 - http://go.microsoft.com/fwlink/?LinkID=239643&clcid=0x409
X64 - http://go.microsoft.com/fwlink/?LinkID=239644&clcid=0x409

VS References issues

Before backup, delete all bin and obj folders in solution

When using, open solution, right mouse at Solution-> Restore Nuget Pageage

Waiting for all references reinstall. Then clean solution and rebuild

AutoMapper and Autofac

//register

```
public class AutoMapperModule : Autofac.Module
{
 protected override void Load(ContainerBuilder builder)
 {
        base.Load(builder);
```


bigrixin@hotmail.com

```
        //register all profile classes in the calling assembly
        builder.RegisterAssemblyTypes(typeof(AutoMapperModule).Assembly).As<Profile>();

        builder.Register(context => new MapperConfiguration(cfg =>
        {
                foreach (var profile in context.Resolve<IEnumerable<Profile>>())
                {
                        cfg.AddProfile(profile);
                }
        })).AsSelf().SingleInstance();

        builder.Register(c => c.Resolve<MapperConfiguration>().CreateMapper(c.Resolve))
                        .As<IMapper>()
                        .InstancePerLifetimeScope();
 }
}


// setting mapper
public class CoordinatorMappingProfile : Profile
{
        private readonly ICommonService _commonServices;

        public CoordinatorMappingProfile(ICommonService commonServices)
        {
                this._commonServices = commonServices;
                Mappers();
        }

        private void Mappers()
        {
                CreateMap<Coordinator, CoordinatorViewModel>()
                  .ForMember(dest => dest.ID, opt => opt.MapFrom(src => src.ID))
                        .ForMember(dest => dest.SelectedKeyWords, opt => opt.MapFrom(src =>
this._commonServices.GetSelectedKeywords(src.ID)));
        }
}


//Using it in Contraller

private readonly IMapper _mapper;

public CoordinatorController(IMapper mapper)
{
  this._mapper = mapper;
}

[Authorize(Roles = "Coordinator")]
[HttpGet, Route("coordinator/updateprofile")]
public ActionResult UpdateProfile()
{
```


bigrixin@hotmail.com

```
        Coordinator currentCoordinator = this.GetLoggedInUser() as Coordinator;
        CoordinatorProfileEditModel model = _mapper.Map<Coordinator,
CoordinatorProfileEditModel>(currentCoordinator);
        ViewBag.TypeOfUser = this._userServices.GetCurrentRole(model.Email);
        return View(model);
}

[Authorize(Roles = "Coordinator")]
[HttpPost, Route("coordinator/updateprofile")]
public ActionResult UpdateProfile(CoordinatorProfileEditModel model)
{
        Coordinator currentCoordinator = this.GetLoggedInUser() as Coordinator;
        currentCoordinator = _mapper.Map(model, currentCoordinator);
        if (ModelState.IsValid)
        {
                string userType = this._userServices.GetCurrentRole(User.Identity.Name);
                this._coordinatorServices.UpdateProfile(currentCoordinator);
                //      this._commonServices.UpdateSelectedInterest(model.SelectedInterests,
currentCoordinator.ID,userType);
        }
        return RedirectToAction("updateprofilekeywords");
}
```

At App_Start/Startup.Container.cs

```
  builder.RegisterModule<AutoMapperModule>();
```

Show Crow's foot notation in Visio

Database->Display Options->select Crow's feet

Aside bar

```
<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="X-UA-Compatible" content="IE=edge" />
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Dynamic Printing - GWA DocuSearch</title>
  @Styles.Render("~/Bundle/css")
  @RenderSection("PageHeader", required: false)
</head>
<body>
  <header>
    <div id="header-menu"><i id="header-menu-show-drawer" class="material-icons">menu</i></div>
    <div id="header-title">@ViewBag.Title</div>
```


bigrixin@hotmail.com

```
      <div id="header-toolbar">@RenderSection("PageToolbar", required: false)<i id="header-menu-show-account"
class="material-icons" data-menu="header-toolbar-menu-account">account_box</i></div>
   </header>

   <main>
      @RenderBody()
   </main>

   <div id="overlay"></div>

   <aside id="drawer">
      <section id="drawer-about">
         <div id="drawer-about-logo"><img src="~/Content/logo.jpg" width="250" height="94" alt="GWA Logo" /></div>
         <div id="drawer-about-version">DocuSearch Dynamic Printing
v<span>@ViewBag.ApplicationVersion</span></div>
      </section>
      <nav id="drawer-menu">
         @Html.ActionLink("Delivery Docket Printer", "Index", "DynamicPrinting", null, new { @class = "navbar-brand" })
         @if (GWA.DocuSearch.UI2.Authentication.AuthenticatedUser.Create(User).IsAdministrator())
         {
            @Html.ActionLink("Printer Setup", "Index", "Administration", null, new { @class = "navbar-brand" })
         }
      </nav>
   </aside>

   <nav id="header-toolbar-menu-account" class="header-toolbar-menu">
      <div class="header-toolbar-menu-triangle"></div>
      <p><b>Authenticated User:</b> @User.Identity.Name</p>
      @if (GWA.DocuSearch.UI2.Authentication.AuthenticatedUser.Create(User).IsAdministrator())
      {
      <p><b>Access Level:</b> Administrator</p>
      } else  {
      <p><b>Access Level:</b> Standard</p>
      }
   </nav>

   @RenderSection("PageDialogs", required: false)

   @Scripts.Render("~/Bundle/js")
   @RenderSection("PageFooter", required: false)
</body>
</html>


---
Page.cshtml

@{
   ViewBag.Title = "Administration - Printer Setup";
}
@section PageToolbar {
```

```
  <i id="main:administration:printer:create" class="material-icons" title="Create Printer Config">add</i>
}
@section PageDialogs {
  <section id="main:administration:printer:dialog" class="dialog">
    @using (Html.BeginForm("Printer", "Administration", FormMethod.Get, new { @id =
"main:administration:printer:dialog:form", @onsubmit = "return false" }))
    {
      <header><span id="main:administration:printer:dialog:header:verb"></span> Printer Config</header>
      <aside class="layout:vertical-form">
        @Html.Hidden("Id", "", new { @id = "main:administration:printer:dialog:id:input" })
        @Html.AntiForgeryToken()
        <div id="main:administration:printer:dialog:username">
          @Html.Label("Username", "Username", new { @for = "main:administration:printer:dialog:username:input" })
          <span>@Html.TextBox("Username", null, new { @id =
"main:administration:printer:dialog:username:input" })</span>
        </div>
        <div id="main:administration:printer:dialog:hostname">
          @Html.Label("Hostname", "Hostname", new { @for = "main:administration:printer:dialog:hostname:input" })
          <span>@Html.TextBox("Hostname", null, new { @id =
"main:administration:printer:dialog:hostname:input" })</span>
        </div>
        <div id="main:administration:printer:dialog:printer">
          @Html.Label("Printer", "Printer", new { @for = "main:administration:printer:dialog:printer:input" })
          <span>@Html.TextBox("Printer", null, new { @id =
"main:administration:printer:dialog:printer:input" })</span>
        </div>
        <div id="main:administration:printer:dialog:printersettings">
          @Html.Label("PrinterSettings", "Printer Settings", new { @for =
"main:administration:printer:dialog:printersettings:input" })
          <span>@Html.DropDownList("PrinterSettings", (List<SelectListItem>)ViewBag.PrinterSettingOptions, new
{ @id = "main:administration:printer:dialog:printersettings:input" })</span>
        </div>
        <div id="main:administration:printer:dialog:labelprinter">
          @Html.Label("LabelPrinter", "Label Printer", new { @for =
"main:administration:printer:dialog:labelprinter:input" })
          <span>@Html.TextBox("LabelPrinter", null, new { @id =
"main:administration:printer:dialog:labelprinter:input" })</span>
        </div>
        <div id="main:administration:printer:dialog:labelprintersettings">
          @Html.Label("LabelPrinterSettings", "Lable Printer Settings", new { @for =
"main:administration:printer:dialog:lableprintersettings:input" })
          <span>@Html.DropDownList("LabelPrinterSettings",
(List<SelectListItem>)ViewBag.LabelPrinterSettingOptions, new { @id =
"main:administration:printer:dialog:labelprintersettings:input" })</span>
        </div>
      </aside>
      <footer>
        <button type="button" id="main:administration:printer:dialog:cancel:button">Cancel</button>
        <button type="submit" id="main:administration:printer:dialog:submit:button">Save</button>
      </footer>
    }
```

```
    </section>
    <section id="main:error:dialog" class="dialog">
      <header></header>
      <aside></aside>
      <footer><button type="button" id="main:error:dialog:cancel:button">OK</button></footer>
    </section>
}
@section PageFooter {
    @Scripts.Render("~/Scripts/page-administration-index.js")
}

<div class="layout:fill-and-scroll-y">
    <table class="layout:list" id="main:administration:printer:listing" data-data-
source="@Href("~/Administration/Printer")">
      <thead>
        <tr>
          <th>Id</th>
          <th>Username</th>
          <th>Hostname</th>
          <th>Printer</th>
          <th>Printer Settings</th>
          <th>Label Printer</th>
          <th>Label Printer Settings</th>
          <th></th>
        </tr>
      </thead>
      <tbody id="main:administration:printer:listing:body"></tbody>
    </table>
</div>
```


bigrixin@hotmail.com