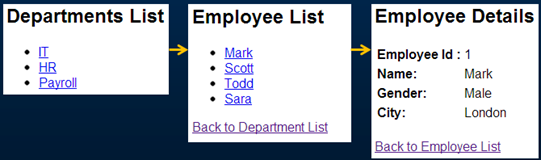
Please [watch Part 9](http://csharp-video-tutorials.blogspot.com/2013/05/part-9-generate-hyperlinks-using.html), before proceeding.  
  
**In this video we will discuss working with 2 related tables in MVC**  
**1.** tblDepartment  
**2.** tblEmployee

**Use the sql script to create and populate these 2 tables**  
Create table tblDepartment  
(  
 Id int primary key identity,  
 Name nvarchar(50)  
)  
  
Insert into tblDepartment values('IT')  
Insert into tblDepartment values('HR')  
Insert into tblDepartment values('Payroll')  
  
Create table tblEmployee  
(  
 EmployeeId int Primary Key Identity(1,1),  
 Name nvarchar(50),  
 Gender nvarchar(10),  
 City nvarchar(50),  
 DepartmentId int  
)  
  
Alter table tblEmployee  
add foreign key (DepartmentId)  
references tblDepartment(Id)  
  
Insert into tblEmployee values('Mark','Male','London',1)  
Insert into tblEmployee values('John','Male','Chennai',3)  
Insert into tblEmployee values('Mary','Female','New York',3)  
Insert into tblEmployee values('Mike','Male','Sydeny',2)

Insert into tblEmployee values('Scott','Male','London',1)  
Insert into tblEmployee values('Pam','Female','Falls Church',2)  
Insert into tblEmployee values('Todd','Male','Sydney',1)  
Insert into tblEmployee values('Ben','Male','New Delhi',2)  
Insert into tblEmployee values('Sara','Female','London',1)  
  
**This is what we want to achieve**  
**1.** Display all the departments from **tblDepartments** table. The Department names should be rendered as hyperlinks.  
**2.** On clicking the **department name link**, all the employees in the department should be displayed. The employee names should be rendered as hyperlinks.  
**3.** On clicking the **employee name link**, the full details of the employee should be displayed.   
**4.** A link should also be provided on the employee full details page to navigate back to **Employee list page**. Along the same lines, a link should also be provided on the employee list page to navigate back to Departments list page.   
  
**The screen shots of the above workflow is shown below for your reference**   
   
  
**Implementing Departments List:**  
**Step 1:** Right click on the **"Models"** folder and add a class file with name=**Department.cs**. Copy and paste the following code.  
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Web;  
using System.ComponentModel.DataAnnotations.Schema;  
  
namespace MVCDemo.Models  
{  
    [Table("tblDepartment")]  
    public class Department  
    {  
        public int ID { get; set; }  
        public string Name { get; set; }  
        public List<Employee> Employees { get; set; }  
    }  
}  
  
**Step 2:** Add **"Departments"** property to **"EmployeeContext"** class that is present in **"EmployeeContext.cs"** file in **"Models"** folder.  
public class EmployeeContext : DbContext  
{  
    public DbSet<Department> Departments { get; set; }  
    public DbSet<Employee> Employees { get; set; }  
}  
  
**Step 3:** Right click on the **"Controllers"** folder and add a Controller, with name=**DepartmentController**. Copy and paste the following code.  
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Web;  
using System.Web.Mvc;  
using MVCDemo.Models;  
  
namespace MVCDemo.Controllers  
{  
    public class DepartmentController : Controller  
    {  
        public ActionResult Index()  
        {  
            EmployeeContext employeeContext = new EmployeeContext();  
            List<Department> departments = employeeContext.Departments.ToList();  
            return View(departments);  
        }  
    }  
}  
  
**Step 4:** Right click on the **Index**() action method in **DepartmentController** class and select **"Add View"** from the context menu. Set  
View name = **Index**  
View engine = **Razor**  
Select **"Create Strongly-typed view"**checkbox  
Select **Department** class, from **"Model class"** dropdownlist  
Click **"Add"** button  
  
**Copy and paste the following code in Index.cshtml view file in Department folder**  
@using MVCDemo.Models;  
  
@model IEnumerable<Department>  
  
<div style="font-family:Arial">  
@{  
    ViewBag.Title = "Departments List";  
}  
  
<h2>Departments List</h2>  
<ul>  
@foreach (Department department in @Model)  
{  
    <li>@Html.ActionLink(department.Name, "Index", "Employee",   
    new { departmentId = department.ID }, null)</li>  
}  
</ul>  
</div>  
  
**Changes to Employee List and Detail pages**  
Add **"DepartmentId"** property to **"Employee"** model class that is present in **Employee.cs** file in **"Models"** folder.  
[Table("tblEmployee")]  
public class Employee  
{  
    public int EmployeeId { get; set; }  
    public string Name { get; set; }  
    public string Gender { get; set; }  
    public string City { get; set; }  
    public int DepartmentId { get; set; }  
}  
  
Add **"departmentId"** parameter to **Index**() action method in **"EmployeeController"**class that is present in **"EmployeeController.cs"** file in **"Controllers"** folder. Use the **"departmentId"** parameter to filter the list of employees as shown below.  
public ActionResult Index(int departmentId)  
{  
    EmployeeContext employeeContext = new EmployeeContext();  
    List<Employee> employees = employeeContext.Employees.Where(emp => emp.DepartmentId == departmentId).ToList();  
  
    return View(employees);  
}  
  
Copy and paste the following line in **"Index.cshtml"** that is present in **"Employee"**folder in **"Views"** folder. With this change we are able to generate an action link to redirect the user to a different controller action method.  
@Html.ActionLink("Back to Department List", "Index", "Department")

Change the following line in **"Details.cshtml"** that is present in **"Employee"** folder in **"Views"** folder.   
CHANGE THIS LINE @Html.ActionLink("Back to List", "Index")    
TO   
@Html.ActionLink("Back to Employee List", "Index", new { departmentId = @Model.DepartmentId })