**3**

|  |  |  |
| --- | --- | --- |
| [Model](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#Model) |  |  |
| CodeTalk.**Domain** | Models\ | Talk.cs |
|  |  | Comment.cs |
| [Interfaces](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#Interfaces) |  |  |
| CodeTalk.**Domain** | Contracts\Repositories\ | ITalkRepository.cs |
|  |  |  |
| [Context](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#Context) |  |  |
| CodeTalk.**DataSource** | \ | CodeTalkContext.cs |
|  |  |  |
| [Repository](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#Repository) |  |  |
| CodeTalk.**DataSource** | \Repositories\ | TalkRepository.cs |
|  |  |  |
| [ServiceLayer](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#ServiceLayer) |  |  |
| CodeTalk.**Domain** | Contracts\Services\ | ITalkService.cs |
|  |  |  |
| CodeTalk.**ServiceLayer** | \ | TalkService.cs |
|  |  |  |
| [Controller](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#Controller) |  |  |
| CodeTalk.**Web** | \Controllers | TalkController.cs |
|  |  |  |
| [Views](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#Views) |  |  |
| CodeTalk.**Web** | \Views\Shared | \_Layout.cshtml |
|  | \Views\Talk | Index.cshtml  Insert.cshtml |

**Model**

CodeTalk.Domain.Models

References:

CodeTalk.**Domain** (none)

**POCO Models**

no project references

domain objects

vehicles for data

properties that define data

EF will populate them

POCO

* Plain Old Class Object
* Independent and stand alone objects
* Easy to pass around

Web API > Unit of Work > Repositories > EF > db

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace CodeTalk.Domain.Models

{

    public class Talk

    {

        public Talk()

        {

            DateCreated = DateTime.Now;

            DateModified = DateTime.Now;

        }

        public int Id { get; set; }

        public string Title { get; set; }

        public string Description { get; set; }

        ICollection<Comment> Comments { get; set; }

        public DateTime DateCreated { get; set; }

        public DateTime DateModified { get; set; }

    }

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace CodeTalk.Domain.Models

{

    public class Comment

    {

        public Comment()

        {

            DateCreated = DateTime.Now;

            DateModified = DateTime.Now;

        }

        public int Id { get; set; }

        public string Body { get; set; }

        public int TalkId { get; set; }

        public DateTime DateCreated { get; set; }

        public DateTime DateModified { get; set; }

    }

}

DataSource

Domain

ServiceLayer

Web

**3a**

|  |  |  |
| --- | --- | --- |
| [Model](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#Model) |  |  |
| CodeTalk.**Domain** | Models\ | Talk.cs |
|  |  | Comment.cs |
| [Interfaces](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#Interfaces) |  |  |
| CodeTalk.**Domain** | Contracts\Repositories\ | ITalkRepository.cs |
|  |  |  |
| [Context](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#Context) |  |  |
| CodeTalk.**DataSource** | \ | CodeTalkContext.cs |
|  |  |  |
| [Repository](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#Repository) |  |  |
| CodeTalk.**DataSource** | \Repositories\ | TalkRepository.cs |
|  |  |  |
| [ServiceLayer](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#ServiceLayer) |  |  |
| CodeTalk.**Domain** | Contracts\Services\ | ITalkService.cs |
|  |  |  |
| CodeTalk.**ServiceLayer** | \ | TalkService.cs |
|  |  |  |
| [Controller](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#Controller) |  |  |
| CodeTalk.**Web** | \Controllers | TalkController.cs |
|  |  |  |
| [Views](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#Views) |  |  |
| CodeTalk.**Web** | \Views\Shared | \_Layout.cshtml |
|  | \Views\Talk | Index.cshtml  Insert.cshtml |

**Interfaces**

CodeTalk.Domain.Contracts.Repositories

References:

CodeTalk.**Domain** (none)

Just methods we need to get at data

//paging, ordering, grouping, sorting, filtering…

so IQueryable (rather than IEnumerable)

Just start with   
-GetTalks()   
-AddTalk(Talk newTalk)

Allows us to mock up a repo for testing later

using CodeTalk.Domain.Models;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace CodeTalk.Domain.Contracts.Repositories

{

    public interface ITalkRepository

    {

        IQueryable<Talk> GetTalks();

        bool AddTalk(Talk newTalk);

        // 3a

        Talk GetTalkById(int talkId);

        void EditTalk(Talk updatedTalk);

        void DeleteTalk(int deleteTalkWithThisId);

    }

}

DataSource

Domain

ServiceLayer

Web

3-

|  |  |  |
| --- | --- | --- |
| [Model](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#Model) |  |  |
| CodeTalk.**Domain** | Models\ | Talk.cs |
|  |  | Comment.cs |
| [Interfaces](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#Interfaces) |  |  |
| CodeTalk.**Domain** | Contracts\Repositories\ | ITalkRepository.cs |
|  |  |  |
| [Context](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#Context) |  |  |
| CodeTalk.**DataSource** | \ | CodeTalkContext.cs |
|  |  |  |
| [Repository](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#Repository) |  |  |
| CodeTalk.**DataSource** | \Repositories\ | TalkRepository.cs |
|  |  |  |
| [ServiceLayer](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#ServiceLayer) |  |  |
| CodeTalk.**Domain** | Contracts\Services\ | ITalkService.cs |
|  |  |  |
| CodeTalk.**ServiceLayer** | \ | TalkService.cs |
|  |  |  |
| [Controller](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#Controller) |  |  |
| CodeTalk.**Web** | \Controllers | TalkController.cs |
|  |  |  |
| [Views](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#Views) |  |  |
| CodeTalk.**Web** | \Views\Shared | \_Layout.cshtml |
|  | \Views\Talk | Index.cshtml  Insert.cshtml |

**Context**

CodeTalk.DataSource

References:

CodeTalk.**DataSource**

* CodeTalk.Domain
* EntityFramework

- communication between code and db

- conduit for crud

Entity Framework - Nuget package or Add reference

"DefaultConnection"- from security video?

using CodeTalk.Domain.Models;

using System;

using System.Collections.Generic;

using System.Data.Entity;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace CodeTalk.DataSource

{

    public class CodeTalkContext : DbContext

    {

        public CodeTalkContext()

            : base("DefaultConnection")

        {

        }

        public DbSet<Talk> Talks { get; set; }

    }

}

DataSource

Domain

ServiceLayer

Web

3b, 3c

|  |  |  |
| --- | --- | --- |
| [Model](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#Model) |  |  |
| CodeTalk.**Domain** | Models\ | Talk.cs |
|  |  | Comment.cs |
| [Interfaces](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#Interfaces) |  |  |
| CodeTalk.**Domain** | Contracts\Repositories\ | ITalkRepository.cs |
|  |  |  |
| [Context](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#Context) |  |  |
| CodeTalk.**DataSource** | \ | CodeTalkContext.cs |
|  |  |  |
| [Repository](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#Repository) |  |  |
| CodeTalk.**DataSource** | \Repositories\ | TalkRepository.cs |
|  |  |  |
| [ServiceLayer](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#ServiceLayer) |  |  |
| CodeTalk.**Domain** | Contracts\Services\ | ITalkService.cs |
|  |  |  |
| CodeTalk.**ServiceLayer** | \ | TalkService.cs |
|  |  |  |
| [Controller](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#Controller) |  |  |
| CodeTalk.**Web** | \Controllers | TalkController.cs |
|  |  |  |
| [Views](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#Views) |  |  |
| CodeTalk.**Web** | \Views\Shared | \_Layout.cshtml |
|  | \Views\Talk | Index.cshtml  Insert.cshtml |

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using CodeTalk.Domain.Contracts;

using CodeTalk.Domain.Contracts.Repositories;

using CodeTalk.Domain.Models;

namespace CodeTalk.DataSource.Repositories

{

    public class TalkRepository: ITalkRepository

    {

        //3b

        //private CodeTalkContext ctx;

        private CodeTalkContext codeTalkContext;

        public TalkRepository()

        {

            codeTalkContext = new CodeTalkContext();

        }

        public IQueryable<Talk> GetTalks()

        {

            //3b var ctx = new CodeTalkContext();

            return codeTalkContext.Talks;

        }

        public bool AddTalk(Talk newTalk)

        {

            //3b var ctx = new CodeTalkContext();

            codeTalkContext.Talks.Add(newTalk);

            codeTalkContext.SaveChanges();

            return true;

        }

        //3c

        public Talk GetTalkById(int talkId)

        {

            return codeTalkContext.Talks

.FirstOrDefault(t => t.Id == talkId);

        }

        public void EditTalk(Talk updatedTalk)

        {

            Talk existingTalk = codeTalkContext.Talks

.FirstOrDefault(t => t.Id == updatedTalk.Id);

            existingTalk.Title = updatedTalk.Title;

            existingTalk.Description = updatedTalk.Description;

            existingTalk.DateCreated = updatedTalk.DateCreated;

            existingTalk.DateModified = DateTime.Now;

            codeTalkContext.SaveChanges();

        }

        public void DeleteTalk(int deleteTalkWithThisId)

        {

            Talk deleteThisTalk = GetTalkById(deleteTalkWithThisId);

            codeTalkContext.Talks.Remove(deleteThisTalk);

            codeTalkContext.SaveChanges();

        }

    }

}

DataSource

Domain

ServiceLayer

Web

**Repository**

CodeTalk.DataSource.Repositories

References:

CodeTalk.**DataSource**

* CodeTalk.Domain
* EntityFramework

**ServiceLayer**

|  |  |  |
| --- | --- | --- |
| [Model](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#Model) |  |  |
| CodeTalk.**Domain** | Models\ | Talk.cs |
|  |  | Comment.cs |
| [Interfaces](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#Interfaces) |  |  |
| CodeTalk.**Domain** | Contracts\Repositories\ | ITalkRepository.cs |
|  |  |  |
| [Context](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#Context) |  |  |
| CodeTalk.**DataSource** | \ | CodeTalkContext.cs |
|  |  |  |
| [Repository](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#Repository) |  |  |
| CodeTalk.**DataSource** | \Repositories\ | TalkRepository.cs |
|  |  |  |
| [ServiceLayer](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#ServiceLayer) |  |  |
| CodeTalk.**Domain** | Contracts\Services\ | ITalkService.cs |
|  |  |  |
| CodeTalk.**ServiceLayer** | \ | TalkService.cs |
|  |  |  |
| [Controller](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#Controller) |  |  |
| CodeTalk.**Web** | \Controllers | TalkController.cs |
|  |  |  |
| [Views](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#Views) |  |  |
| CodeTalk.**Web** | \Views\Shared | \_Layout.cshtml |
|  | \Views\Talk | Index.cshtml  Insert.cshtml |

3d.

**ITalkService**

CodeTalk.Domain.Contracts.Services

References:

CodeTalk.**Domain** (none)

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using CodeTalk.Domain.Models;

namespace CodeTalk.Domain.Contracts.Services

{

    public interface ITalkService

    {

        IList<Talk> GetTalks();

        bool AddTalk(Talk newTalk);

        // 3d

        Talk GetTalkById(int talkId);

        void EditTalk(Talk updatedTalk);

        void DeleteTalk(int deleteTalkWithThisId);

    }

}

DataSource

Domain

ServiceLayer

Web

3e, 3f, 3g

|  |  |  |
| --- | --- | --- |
| [Model](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#Model) |  |  |
| CodeTalk.**Domain** | Models\ | Talk.cs |
|  |  | Comment.cs |
| [Interfaces](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#Interfaces) |  |  |
| CodeTalk.**Domain** | Contracts\Repositories\ | ITalkRepository.cs |
|  |  |  |
| [Context](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#Context) |  |  |
| CodeTalk.**DataSource** | \ | CodeTalkContext.cs |
|  |  |  |
| [Repository](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#Repository) |  |  |
| CodeTalk.**DataSource** | \Repositories\ | TalkRepository.cs |
|  |  |  |
| [ServiceLayer](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#ServiceLayer) |  |  |
| CodeTalk.**Domain** | Contracts\Services\ | ITalkService.cs |
|  |  |  |
| CodeTalk.**ServiceLayer** | \ | TalkService.cs |
|  |  |  |
| [Controller](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#Controller) |  |  |
| CodeTalk.**Web** | \Controllers | TalkController.cs |
|  |  |  |
| [Views](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#Views) |  |  |
| CodeTalk.**Web** | \Views\Shared | \_Layout.cshtml |
|  | \Views\Talk | Index.cshtml  Insert.cshtml |

**TalkService**

CodeTalk.ServiceLayer

References:

CodeTalk.**ServiceLayer**

* CodeTalk.DataSource
* CodeTalk.Domain

using CodeTalk.Domain.Models;

using CodeTalk.Domain.Contracts.Repositories;

using CodeTalk.Domain.Contracts.Services;

using CodeTalk.DataSource;

using CodeTalk.DataSource.Repositories;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace CodeTalk.ServiceLayer

{

    public class TalkService : ITalkService

    {

        //3f - rename repo //private ITalkRepository repo;

        //3e -private var and constructor

        private ITalkRepository talkRepository;

        public TalkService()

        {

            //3e  //repo = new TalkRepository();

            talkRepository = new TalkRepository();

        }

        public IList<Talk> GetTalks()

        {

            //3e  //var repo = new TalkRepository();

            return talkRepository.GetTalks().ToList();

        }

        public bool AddTalk(Domain.Models.Talk newTalk)

        {

            //3e  //var repo = new TalkRepository();

            return talkRepository.AddTalk(newTalk);

        }

        //3g

        public Talk GetTalkById(int talkId)

        {

            return talkRepository.GetTalkById(talkId);

        }

        public void EditTalk(Talk updatedTalk)

        {

            talkRepository.EditTalk(updatedTalk);

        }

        public void DeleteTalk(int deleteTalkWithThisId)

        {

            talkRepository.DeleteTalk(deleteTalkWithThisId);

        }

    }

}

DataSource

Domain

ServiceLayer

Web

3h

**Controller**

CodeTalk.Web.Controllers

References:

CodeTalk.**Web**

* CodeTalk.Domain
* CodeTalk.ServiceLayer
* EntityFramework

DataSource

Domain

ServiceLayer

Web

|  |  |  |
| --- | --- | --- |
| [Model](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#Model) |  |  |
| CodeTalk.**Domain** | Models\ | Talk.cs |
|  |  | Comment.cs |
| [Interfaces](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#Interfaces) |  |  |
| CodeTalk.**Domain** | Contracts\Repositories\ | ITalkRepository.cs |
|  |  |  |
| [Context](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#Context) |  |  |
| CodeTalk.**DataSource** | \ | CodeTalkContext.cs |
|  |  |  |
| [Repository](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#Repository) |  |  |
| CodeTalk.**DataSource** | \Repositories\ | TalkRepository.cs |
|  |  |  |
| [ServiceLayer](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#ServiceLayer) |  |  |
| CodeTalk.**Domain** | Contracts\Services\ | ITalkService.cs |
|  |  |  |
| CodeTalk.**ServiceLayer** | \ | TalkService.cs |
|  |  |  |
| [Controller](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#Controller) |  |  |
| CodeTalk.**Web** | \Controllers | TalkController.cs |
|  |  |  |
| [Views](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#Views) |  |  |
| CodeTalk.**Web** | \Views\Shared | \_Layout.cshtml |
|  | \Views\Talk | Index.cshtml  Insert.cshtml |

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

using CodeTalk.Domain.Models;

using CodeTalk.ServiceLayer;

namespace CodeTalk.Controllers

{

    public class TalkController : Controller

    {

        //3h

        // Constructor

        private TalkService talkService;

        public TalkController()

        {

            talkService = new TalkService();

        }

        public ActionResult Index()

        {

            //3h -//var talkService = new TalkService();

            var talks = talkService.GetTalks();

            return View(talks);

        }

        public ActionResult Insert()

        {

            //var talk = new Talk();

            //return View(talk);

            return View();

        }

        [HttpPost]

        public ActionResult Insert(Talk newTalk)

        {

            //3h - //var talkService = new TalkService();

            talkService.AddTalk(newTalk);

            //return View();

            return RedirectToAction("Index");

        }

        //3i

        public ActionResult Edit(int id)

        {

            Talk talk = talkService.GetTalkById(id);

            return View(talk);

        }

        [HttpPost]

        public ActionResult Edit(Talk updatedTalk)

        {

            talkService.EditTalk(updatedTalk);

            return RedirectToAction("Index");

        }

        public ActionResult Details(int id)

        {

            Talk talk = talkService.GetTalkById(id);

            return View(talk);

        }

        public ActionResult Delete(int id)

        {

            int deleteTalkWithThisId = id;

            talkService.DeleteTalk(deleteTalkWithThisId);

            return RedirectToAction("Index");

        }

    }

}

3h.views

**Views**

|  |  |  |
| --- | --- | --- |
| [Model](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#Model) |  |  |
| CodeTalk.**Domain** | Models\ | Talk.cs |
|  |  | Comment.cs |
| [Interfaces](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#Interfaces) |  |  |
| CodeTalk.**Domain** | Contracts\Repositories\ | ITalkRepository.cs |
|  |  |  |
| [Context](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#Context) |  |  |
| CodeTalk.**DataSource** | \ | CodeTalkContext.cs |
|  |  |  |
| [Repository](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#Repository) |  |  |
| CodeTalk.**DataSource** | \Repositories\ | TalkRepository.cs |
|  |  |  |
| [ServiceLayer](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#ServiceLayer) |  |  |
| CodeTalk.**Domain** | Contracts\Services\ | ITalkService.cs |
|  |  |  |
| CodeTalk.**ServiceLayer** | \ | TalkService.cs |
|  |  |  |
| [Controller](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#Controller) |  |  |
| CodeTalk.**Web** | \Controllers | TalkController.cs |
|  |  |  |
| [Views](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#Views) |  |  |
| CodeTalk.**Web** | \Views\Shared | \_Layout.cshtml |
|  | \Views\Talk | **Index**.cshtml  Insert.cshtml  Details.cshtml  Edit.cshtml |

CodeTalk.Web.Views.Talk.**Index**

CodeTalk.Web.Views.Talk.**Insert**

@model IEnumerable<CodeTalk.Domain.Models.Talk>

@{

    ViewBag.Title = "Index";

}

<h2>Index</h2>

<p>

    @Html.ActionLink("Create New", "Insert")

</p>

<table>

    <tr>

        <th>

            @Html.DisplayNameFor(model => model.Title)

        </th>

        <th>

            @Html.DisplayNameFor(model => model.Description)

        </th>

        <th>

            @Html.DisplayNameFor(model => model.DateCreated)

        </th>

        <th>

            @Html.DisplayNameFor(model => model.DateModified)

        </th>

        <th></th>

    </tr>

@foreach (var item in Model) {

    <tr>

        <td>

            @Html.DisplayFor(modelItem => item.Title)

        </td>

        <td>

            @Html.DisplayFor(modelItem => item.Description)

        </td>

        <td>

            @Html.DisplayFor(modelItem => item.DateCreated)

        </td>

        <td>

            @Html.DisplayFor(modelItem => item.DateModified)

        </td>

        <td>

            @Html.ActionLink("Edit", "Edit", new { id=item.Id }) |

            @Html.ActionLink("Details", "Details", new { id=item.Id }) |

            @Html.ActionLink("Delete", "Delete", new { id=item.Id })

        </td>

    </tr>

}

</table>

@model CodeTalk.Domain.Models.Talk

@{

    ViewBag.Title = "Insert";

}

<h2>Insert</h2>

@using (Html.BeginForm()) {

    @Html.AntiForgeryToken()

    @Html.ValidationSummary(true)

    <fieldset>

        <legend>Talk</legend>

        <div class="editor-label">

            @Html.LabelFor(model => model.Title)

        </div>

        <div class="editor-field">

            @Html.EditorFor(model => model.Title)

            @Html.ValidationMessageFor(model => model.Title)

        </div>

        <div class="editor-label">

            @Html.LabelFor(model => model.Description)

        </div>

        <div class="editor-field">

            @Html.EditorFor(model => model.Description)

            @Html.ValidationMessageFor(model => model.Description)

        </div>

        @\*

        <div class="editor-label">

            @Html.LabelFor(model => model.DateCreated)

        </div>

        <div class="editor-field">

            @Html.EditorFor(model => model.DateCreated)

            @Html.ValidationMessageFor(model => model.DateCreated)

        </div>

        <div class="editor-label">

            @Html.LabelFor(model => model.DateModified)

        </div>

        <div class="editor-field">

            @Html.EditorFor(model => model.DateModified)

            @Html.ValidationMessageFor(model => model.DateModified)

        </div>\*@

        <p>

            <input type="submit" value="Create" />

        </p>

    </fieldset>

}

<div>

    @Html.ActionLink("Back to List", "Index")

</div>

@section Scripts {

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

}

|  |  |  |
| --- | --- | --- |
| [Model](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#Model) |  |  |
| CodeTalk.**Domain** | Models\ | Talk.cs |
|  |  | Comment.cs |
| [Interfaces](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#Interfaces) |  |  |
| CodeTalk.**Domain** | Contracts\Repositories\ | ITalkRepository.cs |
|  |  |  |
| [Context](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#Context) |  |  |
| CodeTalk.**DataSource** | \ | CodeTalkContext.cs |
|  |  |  |
| [Repository](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#Repository) |  |  |
| CodeTalk.**DataSource** | \Repositories\ | TalkRepository.cs |
|  |  |  |
| [ServiceLayer](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#ServiceLayer) |  |  |
| CodeTalk.**Domain** | Contracts\Services\ | ITalkService.cs |
|  |  |  |
| CodeTalk.**ServiceLayer** | \ | TalkService.cs |
|  |  |  |
| [Controller](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#Controller) |  |  |
| CodeTalk.**Web** | \Controllers | TalkController.cs |
|  |  |  |
| [Views](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#Views) |  |  |
| CodeTalk.**Web** | \Views\Shared | \_Layout.cshtml |
|  | \Views\Talk | Index.cshtml  **Insert**.cshtml  Details.cshtml  Edit.cshtml |

CodeTalk.Web.Views.Talk.**Details**

CodeTalk.Web.Views.Talk.**Edit**

@model CodeTalk.Domain.Models.Talk

@{

    ViewBag.Title = "Details";

}

<h2>Details</h2>

<fieldset>

    <legend>Talk</legend>

    <div class="display-label">

         @Html.DisplayNameFor(model => model.Title)

    </div>

    <div class="display-field">

        @Html.DisplayFor(model => model.Title)

    </div>

    <div class="display-label">

         @Html.DisplayNameFor(model => model.Description)

    </div>

    <div class="display-field">

        @Html.DisplayFor(model => model.Description)

    </div>

    <div class="display-label">

         @Html.DisplayNameFor(model => model.DateCreated)

    </div>

    <div class="display-field">

        @Html.DisplayFor(model => model.DateCreated)

    </div>

    <div class="display-label">

         @Html.DisplayNameFor(model => model.DateModified)

    </div>

    <div class="display-field">

        @Html.DisplayFor(model => model.DateModified)

    </div>

</fieldset>

<p>

    @Html.ActionLink("Edit", "Edit", new { id=Model.Id }) |

    @Html.ActionLink("Back to List", "Index")

</p>

|  |  |  |
| --- | --- | --- |
| [Model](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#Model) |  |  |
| CodeTalk.**Domain** | Models\ | Talk.cs |
|  |  | Comment.cs |
| [Interfaces](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#Interfaces) |  |  |
| CodeTalk.**Domain** | Contracts\Repositories\ | ITalkRepository.cs |
|  |  |  |
| [Context](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#Context) |  |  |
| CodeTalk.**DataSource** | \ | CodeTalkContext.cs |
|  |  |  |
| [Repository](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#Repository) |  |  |
| CodeTalk.**DataSource** | \Repositories\ | TalkRepository.cs |
|  |  |  |
| [ServiceLayer](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#ServiceLayer) |  |  |
| CodeTalk.**Domain** | Contracts\Services\ | ITalkService.cs |
|  |  |  |
| CodeTalk.**ServiceLayer** | \ | TalkService.cs |
|  |  |  |
| [Controller](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#Controller) |  |  |
| CodeTalk.**Web** | \Controllers | TalkController.cs |
|  |  |  |
| [Views](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#Views) |  |  |
| CodeTalk.**Web** | \Views\Shared | \_Layout.cshtml |
|  | \Views\Talk | Index.cshtml  Insert.cshtml  **Details**.cshtml  Edit.cshtml |

@model CodeTalk.Domain.Models.Talk

@{

    ViewBag.Title = "Edit";

}

<h2>Edit</h2>

@using (Html.BeginForm()) {

    @Html.AntiForgeryToken()

    @Html.ValidationSummary(true)

    <fieldset>

        <legend>Talk</legend>

        @Html.HiddenFor(model => model.Id)

        <div class="editor-label">

            @Html.LabelFor(model => model.Title)

        </div>

        <div class="editor-field">

            @Html.EditorFor(model => model.Title)

            @Html.ValidationMessageFor(model => model.Title)

        </div>

        <div class="editor-label">

            @Html.LabelFor(model => model.Description)

        </div>

        <div class="editor-field">

            @Html.EditorFor(model => model.Description)

            @Html.ValidationMessageFor(model => model.Description)

        </div>

        <div class="editor-label">

            @Html.LabelFor(model => model.DateCreated)

        </div>

        <div class="editor-field">

            @Html.EditorFor(model => model.DateCreated)

            @Html.ValidationMessageFor(model => model.DateCreated)

        </div>

        <div class="editor-label">

            @Html.LabelFor(model => model.DateModified)

        </div>

        <div class="editor-field">

            @Html.EditorFor(model => model.DateModified)

            @Html.ValidationMessageFor(model => model.DateModified)

        </div>

        <p>

            <input type="submit" value="Save" />

        </p>

    </fieldset>

}

<div>

    @Html.ActionLink("Back to List", "Index")

</div>

@section Scripts {

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

}

|  |  |  |
| --- | --- | --- |
| [Model](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#Model) |  |  |
| CodeTalk.**Domain** | Models\ | Talk.cs |
|  |  | Comment.cs |
| [Interfaces](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#Interfaces) |  |  |
| CodeTalk.**Domain** | Contracts\Repositories\ | ITalkRepository.cs |
|  |  |  |
| [Context](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#Context) |  |  |
| CodeTalk.**DataSource** | \ | CodeTalkContext.cs |
|  |  |  |
| [Repository](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#Repository) |  |  |
| CodeTalk.**DataSource** | \Repositories\ | TalkRepository.cs |
|  |  |  |
| [ServiceLayer](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#ServiceLayer) |  |  |
| CodeTalk.**Domain** | Contracts\Services\ | ITalkService.cs |
|  |  |  |
| CodeTalk.**ServiceLayer** | \ | TalkService.cs |
|  |  |  |
| [Controller](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#Controller) |  |  |
| CodeTalk.**Web** | \Controllers | TalkController.cs |
|  |  |  |
| [Views](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#Views) |  |  |
| CodeTalk.**Web** | \Views\Shared | \_Layout.cshtml |
|  | \Views\Talk | Index.cshtml  Insert.cshtml  Details.cshtml  **Edit**.cshtml  Delete.cshtml |

ICommentRepository

using CodeTalk.Domain.Models;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace CodeTalk.Domain.Contracts.Repositories

{

    public interface ICommentRepository

    {

        IQueryable<Comment> GetComments();

        Comment GetCommentById(int commentId);

        bool AddComment(Comment newComment);

        void EditComment(Comment updatedComment);

        void DeleteComment(int deleteCommentWithThisId);

    }

}

CommentRepository

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using CodeTalk.Domain.Contracts;

using CodeTalk.Domain.Contracts.Repositories;

using CodeTalk.Domain.Models;

namespace CodeTalk.DataSource.Repositories

{

    public class CommentRepository : ICommentRepository

    {

        // Constructor

        private CodeTalkContext codeTalkContext;

        public CommentRepository()

        {

            codeTalkContext = new CodeTalkContext();

        }

        public IQueryable<Comment> GetComments()

        {

            return codeTalkContext.Comments;

        }

       public Comment GetCommentById(int commentId)

        {

            return codeTalkContext.Comments.FirstOrDefault(c => c.Id == commentId);

        }

        public bool AddComment(Comment newComment)

        {

            codeTalkContext.Comments.Add(newComment);

            codeTalkContext.SaveChanges();

            return true;

        }

        public void EditComment(Comment updatedComment)

        {

            Comment existingComment = codeTalkContext.Comments.FirstOrDefault(c => c.Id == updatedComment.Id);

            existingComment.Body = updatedComment.Body;

            existingComment.DateCreated = updatedComment.DateCreated;

            existingComment.DateModified = DateTime.Now;

            codeTalkContext.SaveChanges();

        }

        public void DeleteComment(int deleteCommentWithThisId)

        {

            Comment deleteThisComment = GetCommentById(deleteCommentWithThisId);

            codeTalkContext.Comments.Remove(deleteThisComment);

            codeTalkContext.SaveChanges();

        }

    }

}

ICommentService

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using CodeTalk.Domain.Models;

namespace CodeTalk.Domain.Contracts.Services

{

    public interface ICommentService

    {

        IList<Comment> GetComments();

        Comment GetCommentById(int commentId);

        bool AddComment(Comment newComment);

        void EditComment(Comment updatedComment);

        void DeleteComment(int deleteCommentWithThisId);

    }

}

using CodeTalk.Domain.Models;

using CodeTalk.Domain.Contracts.Repositories;

using CodeTalk.Domain.Contracts.Services;

using CodeTalk.DataSource;

using CodeTalk.DataSource.Repositories;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace CodeTalk.ServiceLayer

{

    public class CommentService : ICommentService

    {

        private ICommentRepository commentRepository;

        public CommentService()

        {

            commentRepository = new CommentRepository();

        }

        public IList<Comment> GetComments()

        {

            return commentRepository.GetComments().ToList();

        }

        public Comment GetCommentById(int commentId)

        {

            return commentRepository.GetCommentById(commentId);

        }

        public bool AddComment(Comment newComment)

        {

            return commentRepository.AddComment(newComment);

        }

        public void EditComment(Comment updatedComment)

        {

            commentRepository.EditComment(updatedComment);

        }

        public void DeleteComment(int deleteCommentWithThisId)

        {

            commentRepository.DeleteComment(deleteCommentWithThisId);

        }

    }

}

CommentService

CommentController

using CodeTalk.Domain.Models;

using CodeTalk.ServiceLayer;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

namespace CodeTalk.Controllers

{

    public class CommentController : Controller

    {

        //Constructor

        private CommentService commentService;

        public CommentController()

        {

            commentService = new CommentService();

        }

        public ActionResult Index()

        {

            var comments = commentService.GetComments();

            return View(comments);

        }

        public ActionResult Insert(int id)

        {

            Comment comment = new Comment() { TalkId = id };

            return View(comment);

        }

        [HttpPost]

        public ActionResult Insert(Comment newComment)

        {

            commentService.AddComment(newComment);

            return RedirectToAction("Index");

        }

        public ActionResult Edit(int id)

        {

            Comment comment = commentService.GetCommentById(id);

            return View(comment);

        }

        [HttpPost]

        public ActionResult Edit(Comment updatedComment)

        {

            commentService.EditComment(updatedComment);

            return RedirectToAction("Index");

        }

        public ActionResult Details(int id)

        {

            Comment comment = commentService.GetCommentById(id);

            return View(comment);

        }

    }

}

|  |  |  |
| --- | --- | --- |
| [Model](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#Model) |  |  |
| CodeTalk.**Domain** | Models\ | Talk.cs |
|  |  | Comment.cs |
| [Interfaces](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#Interfaces) |  |  |
| CodeTalk.**Domain** | Contracts\Repositories\ | ITalkRepository.cs |
|  |  | ICommentRepository.cs |
| [Context](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#Context) |  |  |
| CodeTalk.**DataSource** | \ | CodeTalkContext.cs |
|  |  |  |
| [Repository](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#Repository) |  |  |
| CodeTalk.**DataSource** | \Repositories\ | TalkRepository.cs |
|  |  | CommentRepository.cs |
| [ServiceLayer](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#ServiceLayer) |  |  |
| CodeTalk.**Domain** | Contracts\Services\ | ITalkService.cs |
|  |  | ICommentService.cs |
| CodeTalk.**ServiceLayer** | \ | TalkService.cs |
|  |  | CommentService.cs |
| [Controller](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#Controller) |  |  |
| CodeTalk.**Web** | \Controllers | TalkController.cs |
|  |  | CommentController.cs |
| [Views](file:///C:\Users\Greg\Documents\Visual%20Studio%202012\Projects\RepoDemos\CodeTalk2%20-%20Steps.docx#Views) |  |  |
| CodeTalk.**Web** | \Views\Shared | \_Layout.cshtml |
|  | \Views\Talk | Index.cshtml  Insert.cshtml  Details.cshtml  Edit.cshtml |
|  | \Views\Comment | Index.cshtml  Insert.cshtml  Details.cshtml  Edit.cshtml |

CodeTalk.Web.Views.Comment.**Index** CodeTalk.Web.Views.Comment.**Insert**

@model CodeTalk.Domain.Models.Comment

@{

    ViewBag.Title = "Insert";

}

<h2>Insert</h2>

@using (Html.BeginForm()) {

    @Html.AntiForgeryToken()

    @Html.ValidationSummary(true)

    <fieldset>

<legend>Comment</legend>

<div class="editor-label">

@Html.LabelFor(model => model.Body)

</div>

<div class="editor-field">

@Html.EditorFor(model => model.Body)

@Html.ValidationMessageFor(model => model.Body)

</div>

<div class="editor-label">

@Html.LabelFor(model => model.TalkId)

</div>

<div class="editor-field">

@Html.EditorFor(model => model.TalkId)

@Html.ValidationMessageFor(model => model.TalkId)

</div>

<div class="editor-label">

@Html.LabelFor(model => model.DateCreated)

</div>

<div class="editor-field">

@Html.EditorFor(model => model.DateCreated)

@Html.ValidationMessageFor(model=>model.DateCreated)

</div>

<div class="editor-label">

@Html.LabelFor(model => model.DateModified)

</div>

<div class="editor-field">

@Html.EditorFor(model => model.DateModified)

@Html.ValidationMessageFor(model=>model.DateModified)

</div>

<p>

<input type="submit" value="Create" />

</p>

    </fieldset>

}

<div>

    @Html.ActionLink("Back to List", "Index")

</div>

@section Scripts {

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

}

@model IEnumerable<CodeTalk.Domain.Models.Comment>

@{

    ViewBag.Title = "Index";

}

<h2>Index</h2>

<p>

    @Html.ActionLink("Create New", "Insert")

</p>

<table>

    <tr>

<th>

  @Html.DisplayNameFor(model => model.Body)

</th>

<th>

  @Html.DisplayNameFor(model => model.TalkId)

</th>

<th>

  @Html.DisplayNameFor(model => model.DateCreated)

</th>

<th>

  @Html.DisplayNameFor(model => model.DateModified)

</th>

<th></th>

    </tr>

@foreach (var item in Model) {

    <tr>

<td>

  @Html.DisplayFor(modelItem => item.Body)

</td>

<td>

  @Html.DisplayFor(modelItem => item.TalkId)

</td>

<td>

  @Html.DisplayFor(modelItem => item.DateCreated)

</td>

<td>

  @Html.DisplayFor(modelItem => item.DateModified)

</td>

<td>

  @Html.ActionLink("Edit", "Edit",new{id=item.Id}) |

@Html.ActionLink("Details","Details",new{id=item.Id}) |

@Html.ActionLink("Delete","Delete",new{id=item.Id})

</td>

    </tr>

}

</table>

CodeTalk.Web.Views.Comment.**Details** CodeTalk.Web.Views.Comment.**Edit**

@model CodeTalk.Domain.Models.Comment

@{

    ViewBag.Title = "Details";

}

<h2>Details</h2>

<fieldset>

    <legend>Comment</legend>

    <div class="display-label">

        @Html.DisplayNameFor(model => model.Body)

    </div>

    <div class="display-field">

        @Html.DisplayFor(model => model.Body)

    </div>

    <div class="display-label">

        @Html.DisplayNameFor(model => model.TalkId)

    </div>

    <div class="display-field">

        @Html.DisplayFor(model => model.TalkId)

    </div>

    <div class="display-label">

        @Html.DisplayNameFor(model => model.DateCreated)

    </div>

    <div class="display-field">

        @Html.DisplayFor(model => model.DateCreated)

    </div>

    <div class="display-label">

        @Html.DisplayNameFor(model => model.DateModified)

    </div>

    <div class="display-field">

        @Html.DisplayFor(model => model.DateModified)

    </div>

</fieldset>

<p>

    @Html.ActionLink("Edit", "Edit", new { id=Model.Id }) |

    @Html.ActionLink("Back to List", "Index")

</p>

@model CodeTalk.Domain.Models.Comment

@{

    ViewBag.Title = "Edit";

}

<h2>Edit</h2>

@using (Html.BeginForm()) {

    @Html.AntiForgeryToken()

    @Html.ValidationSummary(true)

    <fieldset>

        <legend>Comment</legend>

        @Html.HiddenFor(model => model.Id)

        <div class="editor-label">

          @Html.LabelFor(model => model.Body)

        </div>

        <div class="editor-field">

          @Html.EditorFor(model => model.Body)

          @Html.ValidationMessageFor(model => model.Body)

        </div>

        <div class="editor-label">

          @Html.LabelFor(model => model.TalkId)

        </div>

        <div class="editor-field">

          @Html.EditorFor(model => model.TalkId)

          @Html.ValidationMessageFor(model => model.TalkId)

        </div>

        <div class="editor-label">

          @Html.LabelFor(model => model.DateCreated)

        </div>

        <div class="editor-field">

          @Html.EditorFor(model => model.DateCreated)

          @Html.ValidationMessageFor(model => model.DateCreated)

        </div>

        <div class="editor-label">

          @Html.LabelFor(model => model.DateModified)

        </div>

        <div class="editor-field">

          @Html.EditorFor(model => model.DateModified)

          @Html.ValidationMessageFor(model => model.DateModified)

        </div>

        <p>

            <input type="submit" value="Save" />

        </p>

    </fieldset>

}

<div>

    @Html.ActionLink("Back to List", "Index")

</div>

@section Scripts {

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

}

