**4j**

|  |  |  |
| --- | --- | --- |
| [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;

using System.ComponentModel.DataAnnotations;

namespace CodeTalk.Domain.Models

{

    public class Comment

    {

        public Comment()

        {

            DateCreated = DateTime.Now;

            DateModified = DateTime.Now;

        }

        public int Id { get; set; }

        [Required]

        public string Body { get; set; }

        [Required]

        public string Commentor { get; set; } // 4j

        public int TalkId { get; set; }

        public DateTime DateCreated { get; set; }

        public DateTime DateModified { get; set; }

    }

}

DataSource

Domain

ServiceLayer

Web

**4g.1**

|  |  |  |
| --- | --- | --- |
| [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

    {

        //Constructor

        private CodeTalkContext codeTalkContext;

        public TalkRepository()

        {

            codeTalkContext = new CodeTalkContext();

        }

        public IQueryable<Talk> GetTalks()

        {

            return codeTalkContext.Talks.Include("Comments"); //4g.1

        }

        public Talk GetTalkById(int talkId)

        {

            return codeTalkContext.Talks.FirstOrDefault(t => t.Id == talkId);

        }

        public bool AddTalk(Talk newTalk)

        {

            codeTalkContext.Talks.Add(newTalk);

            codeTalkContext.SaveChanges();

            return true;

        }

        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

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();

        }

    }

}

HomeController 4a, 4b, 4f, 4i \Views\Home\Index 4c, 4d, 4e, 4g, 4h, 4j

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 HomeController : Controller

    {

        //Constructor

        private TalkService talkService;  //4a

        private CommentService commentService; // 4i

        public HomeController()

        {

            talkService = new TalkService();        //4a

            commentService = new CommentService();  //4i

        }

        //4b

        public ActionResult Index()

        {

            var talks = talkService.GetTalks()

.OrderByDescending(t => t.DateCreated);

            return View(talks);

        }

        //4f

        public ActionResult SubmitTalk()

        {

            ViewBag.Message = "Describe your talk";

            return View();

        }

        [HttpPost]

        public ActionResult SubmitTalk(Talk newTalk)

        {

            talkService.AddTalk(newTalk);

            return RedirectToAction("Index");

        }

        //4i

        public ActionResult AddComment(int id)

        {

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

            return View(comment);

        }

        [HttpPost]

        public ActionResult AddComment(Comment newComment)

        {

            commentService.AddComment(newComment);

            return RedirectToAction("Index");

        }

        //

        //

        public ActionResult About()

        {

            ViewBag.Message = "Your app description.";

            return View();

        }

        public ActionResult Contact()

        {

            ViewBag.Message = "Your contact page.";

            return View();

        }

    }

}

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

@{

ViewBag.Title = "Index";

}

@section featured{

<section class="featured">

<div class="content-wrapper">

<hgroup class="title">

<h1>Code Talks</h1>

</hgroup>

<ul>

@foreach (CodeTalk.Domain.Models.Talk talk in Model)

{

<li>

@\* 4c @talk.Title\*@

<h2>@Html.ActionLink(talk.Title, "Edit", "Talk",

new { id = talk.Id }, null)</h2>   @\* 4d \*@

 @\* 4g \*@

<ul>

@foreach (CodeTalk.Domain.Models.Comment

comment in talk.Comments)

{

<li>

<p>@comment.Body</p>

<p><i>@comment.Commentor</i></p> @\* 4j \*@

</li>

}

</ul>

<p>@Html.ActionLink("Add Comment", "AddComment",

"Home", new { id=talk.Id }, null)</p>  @\* 4h \*@

</li>

}

</ul>

<p>@Html.ActionLink("Submit Talk", "SubmitTalk",

"Home")</p> @\* 4e \*@

</div>

</section>

}

CodeTalk.Web.Views.Home.**SubmitTalk** CodeTalk.Web.Views.Home.**AddComment**

@model CodeTalk.Domain.Models.Comment

@{

    ViewBag.Title = "Add your Comment";

}

<h2>AddComment</h2>

@using (Html.BeginForm()) {

    @Html.AntiForgeryToken()

    @Html.ValidationSummary(true)

    <fieldset>

        <legend>Comment</legend>

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

        <div class="editor-label">

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

        </div>

        <div class="editor-field">

            @Html.TextAreaFor(model => model.Body, new {rows ="6", cols = "100"  })

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

        </div>

        @\* 4j \*@

        <div class="editor-label">

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

        </div>

        <div class="editor-field">

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

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

        </div>

        <p>

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

        </p>

    </fieldset>

}

<div>

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

</div>

@section Scripts {

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

}

@model CodeTalk.Domain.Models.Talk

@{

    ViewBag.Title = "SubmitTalk";

}

<h2>SubmitTalk</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>

        <p>

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

        </p>

    </fieldset>

}

<div>

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

</div>

@section Scripts {

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

}

