CodeTalk6 – WebApi

using CodeTalk.Domain.Models;

using CodeTalk.ServiceLayer;

using CodeTalk.Domain.Contracts.Services;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Net;

using System.Net.Http;

using System.Web.Http;

namespace CodeTalk.Controllers.Api

{

    public class TalksController : ApiController

    {

        //6a

        private ITalkService \_talkService;

        public TalksController(ITalkService talkService)

        {

            \_talkService = talkService;

        }

        //6b

        public IEnumerable<Talk> Get()

        {

           var talks =  \_talkService.GetTalks()

                                    .OrderByDescending(t => t.DateCreated)

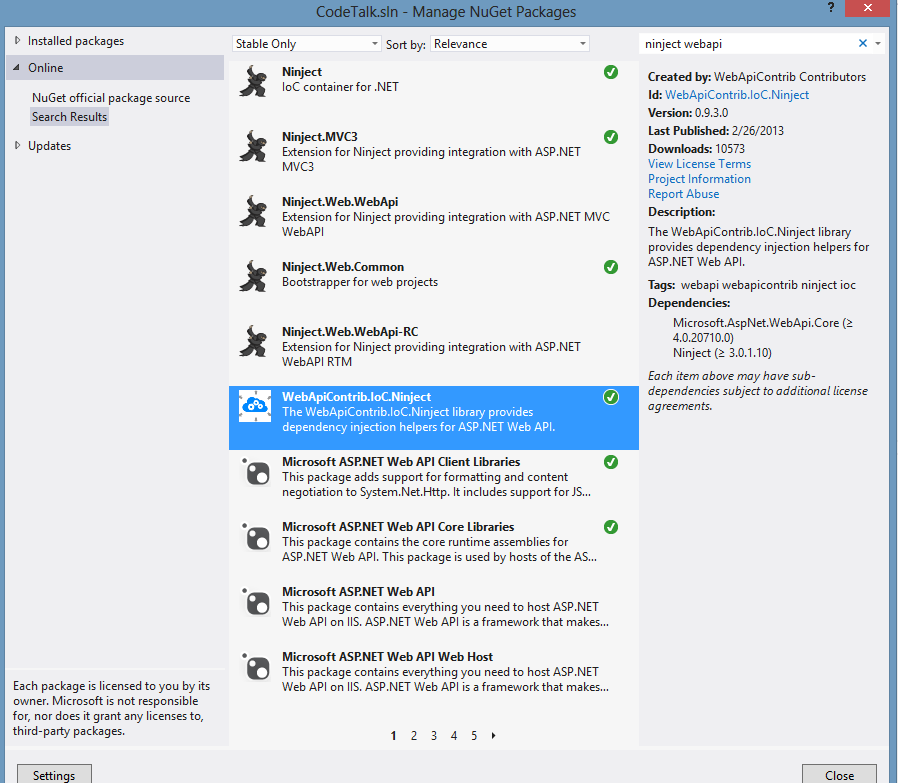
                                    .Take(50);

           return talks;

        }

    }

}



[assembly: WebActivator.PreApplicationStartMethod(typeof(CodeTalk.App\_Start.NinjectWebCommon), "Start")]

[assembly: WebActivator.ApplicationShutdownMethodAttribute(typeof(CodeTalk.App\_Start.NinjectWebCommon), "Stop")]

namespace CodeTalk.App\_Start

{

    using System;

    using System.Web;

    using Microsoft.Web.Infrastructure.DynamicModuleHelper;

    using Ninject;

    using Ninject.Web.Common;

    using CodeTalk.Domain.Contracts.Services;

    using CodeTalk.ServiceLayer;

    using CodeTalk.Domain.Contracts.Repositories;

    using CodeTalk.DataSource;

    using CodeTalk.DataSource.Repositories;

    //6c

    using System.Web.Http;

    using WebApiContrib.IoC.Ninject;

    public static class NinjectWebCommon

    {

        private static readonly Bootstrapper bootstrapper = new Bootstrapper();

        /// <summary>

        /// Starts the application

        /// </summary>

        public static void Start()

        {

            DynamicModuleUtility.RegisterModule(typeof(OnePerRequestHttpModule));

            DynamicModuleUtility.RegisterModule(typeof(NinjectHttpModule));

            bootstrapper.Initialize(CreateKernel);

        }

        /// <summary>

        /// Stops the application.

        /// </summary>

        public static void Stop()

        {

            bootstrapper.ShutDown();

        }

        /// <summary>

        /// Creates the kernel that will manage your application.

        /// </summary>

        /// <returns>The created kernel.</returns>

        private static IKernel CreateKernel()

        {

            var kernel = new StandardKernel();

            kernel.Bind<Func<IKernel>>().ToMethod(ctx => () => new Bootstrapper().Kernel);

            kernel.Bind<IHttpModule>().To<HttpApplicationInitializationHttpModule>();

            RegisterServices(kernel);

            //6c

            GlobalConfiguration.Configuration.DependencyResolver = new NinjectResolver(kernel);

            return kernel;

        }

        /// <summary>

        /// Load your modules or register your services here!

        /// </summary>

        /// <param name="kernel">The kernel.</param>

        private static void RegisterServices(IKernel kernel)

        {

            kernel.Bind<ITalkService>().To<TalkService>().InRequestScope();

            kernel.Bind<ICommentService>().To<CommentService>().InRequestScope();

            kernel.Bind<ITalkRepository>().To<TalkRepository>().InRequestScope();

            kernel.Bind<ICommentRepository>().To<CommentRepository>().InRequestScope();

            kernel.Bind<CodeTalkContext>().To<CodeTalkContext>().InRequestScope();

            //5i

            //kernel.Bind<CodeTalkContext>().To<FakeCodeTalkContext>().InRequestScope();

        }

    }

}

using Newtonsoft.Json.Serialization; //6d

using System;

using System.Collections.Generic;

using System.Linq;

using System.Net.Http.Formatting; //6d

using System.Web.Http;

namespace CodeTalk

{

    public static class WebApiConfig

    {

        public static void Register(HttpConfiguration config)

        {

            //6d

            var jsonFormatter = config.Formatters.OfType<JsonMediaTypeFormatter>().First();

            jsonFormatter.SerializerSettings.ContractResolver = new CamelCasePropertyNamesContractResolver();

            config.Routes.MapHttpRoute(

                name: "DefaultApi",

                routeTemplate: "api/{controller}/{id}",

                defaults: new { id = RouteParameter.Optional }

            );

            // Uncomment the following line of code to enable query support

// for actions with an IQueryable or IQueryable<T> return type.

            // To avoid processing unexpected or malicious queries, use the

// validation settings on QueryableAttribute to validate incoming queries.

            // For more information, visit http://go.microsoft.com/fwlink/?LinkId=279712.

            // config.EnableQuerySupport();

        }

    }

}

<http://localhost:5436/api/talks>

 //6e

        public HttpResponseMessage Post([FromBody]Talk newTalk)

        {

            if (\_talkService.AddTalk(newTalk))

            {

                return Request.CreateResponse(HttpStatusCode.Created, newTalk);

            }

            return Request.CreateResponse(HttpStatusCode.BadRequest);

        }

Request Header:

ContentType: application/json   
Content-Type: application/json

Request Body:

{  
“title”: “Submit WebApi”,  
“description”: “This talk was submitted from the WebApi”  
}

{  
"Title": "Submit WebApi",  
"Description": "This talk was submitted from the WebApi"  
}