Web API handsons

1

using System;

using System.Collections.Generic;

using System.Linq;

using System.Net;

using System.Net.Http;

using System.Web.Http;

namespace WebApi.Controllers

{

public class MoviesController : ApiController

{

public string Get()

{

return "Hello from Web API";

}

public List<string> GetMovies()

{

return new List<string> { "titanic", "Mission Impossible", "Matrix" };

}

}

}

2

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

namespace WebApplication2.Models

{

public class Course

{

public int CourseId { get; set; }

public string CourseName { get; set; }

public string Trainer { get; set; }

public int Fees { get; set; }

public string CourseDescription { get; set; }

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Net;

using System.Net.Http;

using System.Web.Http;

using WebApplication2.Models;

namespace WebApplication2.Controllers

{

public class CoursesController : ApiController

{

static List<Course> courses = new List<Course>()

{

new Course(){CourseId=1,CourseName="Android",Trainer="Shawn",Fees=12000,

CourseDescription="Android is a mobile operating system development"},

new Course(){CourseId=2,CourseName="Asp.net",Trainer="Kevin",Fees=10000,

CourseDescription="Asp.net is a open source web development framework"},

new Course(){CourseId=3,CourseName="JSP",Trainer="Debaratha",Fees=10000,

CourseDescription="Java server pages is a technology used for web page creation" },

new Course(){CourseId=4,CourseName="Xamarin.forms",Trainer="Mark John",Fees=15000,

CourseDescription="Xamarin forms are cross platform UI tools"}

};

public HttpResponseMessage Get(string coursename)

{

Course result = new Course();

result = courses.Where(x => x.CourseName == coursename).FirstOrDefault();

if (result != null)

{

return Request.CreateResponse<Course>(HttpStatusCode.OK, result);

}

else

{

return Request.CreateErrorResponse(HttpStatusCode.NotFound, "Course Name not found");

}

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web.Http;

namespace WebApplication2

{

public static class WebApiConfig

{

public static void Register(HttpConfiguration config)

{

// Web API configuration and services

// Web API routes

config.MapHttpAttributeRoutes();

config.Routes.MapHttpRoute(

name: "DefaultApi",

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

defaults: new { id = RouteParameter.Optional }

);

config.Formatters.Remove(config.Formatters.XmlFormatter);

}

}

}

3

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

namespace WebApplication2.Models

{

public class Course

{

public int CourseId { get; set; }

public string CourseName { get; set; }

public string Trainer { get; set; }

public int Fees { get; set; }

public string CourseDescription { get; set; }

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Net;

using System.Net.Http;

using System.Web.Http;

using WebApplication2.Models;

namespace WebApplication2.Controllers

{

public class CoursesController : ApiController

{

static List<Course> courses = new List<Course>()

{

new Course(){CourseId=1,CourseName="Android",Trainer="Shawn",Fees=12000,

CourseDescription="Android is a mobile operating system development"},

new Course(){CourseId=2,CourseName="Asp.net",Trainer="Kevin",Fees=10000,

CourseDescription="Asp.net is a open source web development framework"},

new Course(){CourseId=3,CourseName="JSP",Trainer="Debaratha",Fees=10000,

CourseDescription="Java server pages is a technology used for web page creation" },

new Course(){CourseId=4,CourseName="Xamarin.forms",Trainer="Mark John",Fees=15000,

CourseDescription="Xamarin forms are cross platform UI tools"}

};

[HttpGet]

public HttpResponseMessage Get()

{

return Request.CreateResponse(HttpStatusCode.OK, courses);

}

[HttpPost]

public HttpResponseMessage Post([FromBody] Course course)

{

courses.Add(course);

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

}

}

}

4

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

namespace WebApplication2.Models

{

public class Brand

{

public string BrandId { get; set; }

public string Name { get; set; }

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Net;

using System.Net.Http;

using System.Web.Http;

using WebApplication2.Models;

namespace WebApplication2.Controllers

{

public class BrandsController : ApiController

{

static List<Brand> brands = new List<Brand>()

{

new Brand(){BrandId="b001",Name="haro"},

new Brand(){BrandId="b002",Name="Electra"},

new Brand(){BrandId="b003",Name="hello"},

new Brand(){BrandId="b004",Name="trek"},

};

[HttpPost]

public HttpResponseMessage Post([FromUri]Brand brand)

{

brands.Add(brand);

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

}

}

}

5

namespace StudentService.Models

{

using System;

using System.Collections.Generic;

public partial class Student

{

public int Id { get; set; }

public string Firstname { get; set; }

public string Lastname { get; set; }

public string Gender { get; set; }

public string Address { get; set; }

}

}

namespace StudentService.Models

{

using System;

using System.Data.Entity;

using System.Data.Entity.Infrastructure;

public partial class StudentDBContext : DbContext

{

public StudentDBContext()

: base("name=StudentDBContext")

{

}

protected override void OnModelCreating(DbModelBuilder modelBuilder)

{

throw new UnintentionalCodeFirstException();

}

public virtual DbSet<Student> Students { get; set; }

}

}

using StudentService.Models;

using System;

using System.Collections.Generic;

using System.Data.Entity.Migrations;

using System.Linq;

using System.Net;

using System.Net.Http;

using System.Web.Http;

namespace StudentService.Controllers

{

public class StudentsController : ApiController

{

StudentDBContext db = new StudentDBContext();

public HttpResponseMessage Get()

{

return Request.CreateResponse(HttpStatusCode.OK, db.Students.ToList());

}

public HttpResponseMessage Get(int id)

{

var user = db.Students.FirstOrDefault(e => e.Id == id);

if (user != null)

{

return Request.CreateResponse(HttpStatusCode.OK, user);

}

else

{

return Request.CreateResponse(HttpStatusCode.NotFound,"student with Id{id} not found");

}

}

public HttpResponseMessage Put(int id,[FromBody]Student student)

{

var user = db.Students.FirstOrDefault(e => e.Id == id);

if (user != null)

{

user.Firstname = student.Firstname;

user.Lastname = student.Lastname;

user.Gender = student.Gender;

user.Address = student.Address;

db.SaveChanges();

return Request.CreateResponse(HttpStatusCode.OK, user);

}

else

{

return Request.CreateResponse(HttpStatusCode.NotFound,

"student with Id"+id.ToString() +"not found to update");

}

}

public HttpResponseMessage Post( [FromBody]Student student)

{

try

{

db.Students.Add(student);

db.SaveChanges();

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

}

catch(Exception ex)

{

return Request.CreateResponse(HttpStatusCode.BadRequest, ex);

}

}

public HttpResponseMessage Delete(int id)

{

var user=db.Students.FirstOrDefault(e => e.Id == id);

if (user != null) {

db.Students.Remove(user);

db.SaveChanges();

return Request.CreateResponse(HttpStatusCode.OK, user);

}

else

{

return Request.CreateResponse(HttpStatusCode.NotFound, "student with Id{id} not found");

}

}

}

}