

SOURCECODE

Controller: EmpController.cs

```
using BLL;
using DAL;
using School_WebAPI.Models;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Net.Http;
using System.Web.Http;

namespace School_WebAPI.Controllers
{
    public class EmpController : ApiController
    {
        // GET api/emp
        static Helper helper;
        static EmpController()
        {
            helper = new Helper();
        }
        public List<Employee> Get()
        {
            var emp = helper.ShowAllEmp();

            List<Employee> emplist = new List<Employee>();
```

```

        foreach (var item in emp)
        {
            emplist.Add(new Employee { EmpCode = item.EmpCode, EmpName = item.EmpName,
DateOfBirth = item.DateOfBirth, Email = item.Email, DeptCode = item.DeptCode });
        }
        return emplist;
    }

// GET api/emp/5
public Employee Get(int id)
{
    var found = helper.ShowEmpById(id);

    Employee got = new Employee { EmpName=found.EmpName, EmpCode=found.EmpCode,
DateOfBirth=found.DateOfBirth, Email=found.Email,DeptCode=found.DeptCode};

    return got;
}

// POST api/emp
public void Post([FromBody] Employee emp)
{
    EmpProfile e = new EmpProfile();
    e.EmpName = emp.EmpName;
    e.EmpCode = emp.EmpCode;
    e.DateOfBirth = emp.DateOfBirth;
    e.Email = emp.Email;
    e.DeptCode = emp.DeptCode;

    helper.CreateEmp(e);
}

// PUT api/emp/5

```

```

public void Put(int id, [FromBody] Employee upd)
{
    EmpProfile e = new EmpProfile();
    e.EmpName = upd.EmpName;
    e.EmpCode = upd.EmpCode;
    e.DateOfBirth = upd.DateOfBirth;
    e.Email = upd.Email;
    e.DeptCode = upd.DeptCode;

    helper.UpdateEmp(id, e);

}

// DELETE api/emp/5
public void Delete(int id)
{
    helper.RemoveEmp(id);
}
}

```

DeptController.cs

```

using BLL;
using DAL;
using School_WebAPI.Models;
using System;
using System.Collections.Generic;
using System.Linq;

```

```
using System.Net;

using System.Net.Http;

using System.Web.Http;


namespace School_WebAPI.Controllers
{
    public class DeptController : ApiController
    {
        static Helper helper;

        static DeptController()
        {

            helper = new Helper();
        }

        public List<Department> Get()
        {
            var dept = helper.ShowAllDept();

            List<Department> deptlist = new List<Department>();

            foreach (var item in dept)
            {
                deptlist.Add(new Department { DeptCode=item.DeptCode, DeptName= item.DeptName });
            }

            return deptlist;
        }


        // GET api/emp/5
        public Department Get(int id)
        {
            var found = helper.ShowDeptById(id);
```

```
        Department got = new Department { DeptCode = found.DeptCode, DeptName =  
found.DeptName };  
        return got;  
    }  
}
```

```
// POST api/emp
```

```
public void Post([FromBody] Department d)
```

```
{  
    DeptMaster e = new DeptMaster();  
    e.DeptCode = d.DeptCode;  
    e.DeptName = d.DeptName;  
  
    helper.CreateDept(e);  
  
}
```

```
// PUT api/emp/5
```

```
public void Put(int id, [FromBody] Department upd)
```

```
{  
    DeptMaster e = new DeptMaster();  
    e.DeptCode = upd.DeptCode;  
    e.DeptName = upd.DeptName;  
  
    helper.UpdateDept(id, e);  
  
}
```

```
// DELETE api/emp/5
```

```
public void Delete(int id)
```

```
{
```

```

        helper.RemoveDept(id);
    }
}

```

models in webapi:

Department.cs

```

using DAL;
using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations.Schema;
using System.ComponentModel.DataAnnotations;
using System.Linq;
using System.Web;

namespace School_WebAPI.Models
{
    public class Department
    {
        [Required]
        public int DeptCode { get; set; }

        [Required]
        [MaxLength(50, ErrorMessage = "Maximum 50 Characters only")]
        public string DeptName { get; set; }

        public virtual ICollection<Employee> EmpProfiles { get; set; }
    }
}

```

Employee.cs

```
using DAL;
using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations.Schema;
using System.ComponentModel.DataAnnotations;
using System.Linq;
using System.Web;

namespace School_WebAPI.Models
{
    public class Employee
    {
        [Required]

        public int EmpCode { get; set; }

        [DisplayFormat(DataFormatString = "{0:dd-MM-yyyy}",
            ApplyFormatInEditMode = true)]
        public DateTime DateOfBirth { get; set; }

        [Required]
        [MaxLength(50, ErrorMessage = "Maximum 50 Characters only")]
        public string EmpName { get; set; }

        [EmailAddress]
```

```
public string Email { get; set; }
```

```
[Required]
```

```
public int DeptCode { get; set; }
```

```
public virtual Department Dept { get; set; }
```

```
}
```

```
}
```

```
BLL
```

```
Helper.cs
```

```
using DAL;
```

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.Linq;
```

```
using System.Runtime.Remoting.Contexts;
```

```
using System.Text;
```

```
using System.Threading.Tasks;
```

```
namespace BLL
```

```
{
```

```
    public class Helper
```

```
    {
```

```
        static EmpOperations opr;
```

```
        static Helper()
```

```
        {
```



```
        opr = new EmpOperations();
    }
    public List<EmpProfile> ShowAllEmp()
    {

        return opr.GetAllEmp();

    }

    public EmpProfile ShowEmpById(int id)
    {

        var fnd = opr.GetEmpById(id);
        return fnd;

    }

    public void CreateEmp(EmpProfile e)
    {

        opr.PostEmp(e);

    }

    public void UpdateEmp(int id, EmpProfile e)
    {

        opr.PutEmp(id, e);

    }
```

```
public void RemoveEmp(int id)
```

```
{
```

```
    opr.DelEmp(id);
```

```
}
```

```
DEPARTMENT HELPER
```

```
public List<DeptMaster> ShowAllDept()
```

```
{
```

```
    return opr.GetAllDept();
```

```
}
```

```
public DeptMaster ShowDeptById(int id)
```

```
{
```

```
    var fnd = opr.GetDeptById(id);
```

```
    return fnd;
```

```
}
```

```
public void CreateDept(DeptMaster d)
```

```
{
```

```
    opr.PostDept(d);
```

```
}
```

```
public void UpdateDept(int id, DeptMaster e)
```

```
{
```

```

        opr.PutDept(id, e);

    }

    public void RemoveDept(int id)
    {

        opr.DelDept(id);

    }

}

}
}
DAL(CodeFirst)

```

DeptMaster.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations.Schema;
using System.ComponentModel.DataAnnotations;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace DAL
{
    public class DeptMaster
    {
        //Dept table each entry template
    }
}

```

```

[DatabaseGenerated(DatabaseGeneratedOption.None)]

[Key]
public int DeptCode { get; set; }

[MaxLength(50, ErrorMessage = "Maximum 50 Characters only")]
public string DeptName { get; set; }
public virtual ICollection<EmpProfile> EmpProfiles { get; set; }
}
}

```

EmpProfile.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations.Schema;
using System.ComponentModel.DataAnnotations;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

```

```

namespace DAL
{
    public class EmpProfile
    {

        // Employee table each entry template

```

```

[Key]

[DatabaseGenerated(DatabaseGeneratedOption.None)]
public int EmpCode { get; set; }

```

```

        [Column(TypeName = "Date")]
        [DisplayFormat(DataFormatString = "{0:dd-MM-yyyy}",
            ApplyFormatInEditMode = true)]
        public DateTime DateOfBirth { get; set; }

        [Required]
        [MaxLength(50, ErrorMessage = "Maximum 50 Characters only")]
        public string EmpName { get; set; }

        [EmailAddress]
        public string Email { get; set; }

        [ForeignKey("DeptMaster")]
        [Required]
        public int DeptCode { get; set; }

        public virtual DeptMaster DeptMaster { get; set; }

    }
}

```

EmpOperations.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace DAL
{
    public class EmpOperations

```

```
{  
    static EmpManagementContext context;  
  
    static EmpOperations()  
    {  
        context = new EmpManagementContext();  
    }  
  
    public List<EmpProfile> GetAllEmp()  
    {  
  
        return context.EmpProfiles.ToList();  
  
    }  
  
    public EmpProfile GetEmpById(int id)  
    {  
  
        var found = context.EmpProfiles.ToList().Find(e => e.EmpCode == id);  
        return found;  
  
    }  
  
    public void PostEmp(EmpProfile e)  
    {  
  
        context.EmpProfiles.Add(e);  
        context.SaveChanges();  
    }  
    public void PutEmp(int id, EmpProfile e)  
    {
```

```
var found = context.EmpProfiles.ToList().Find(a => a.EmpCode == id);
context.EmpProfiles.Remove(found);
context.EmpProfiles.Add(e);
context.SaveChanges();
}

public void DelEmp(int id)
{
    var found = context.EmpProfiles.ToList().Find(a => a.EmpCode == id);
    context.EmpProfiles.Remove(found);
    context.SaveChanges();
}
```

Dept operations

```
public List<DeptMaster> GetAllDept()
{

    return context.DeptMasters.ToList();

}
```

```
public DeptMaster GetDeptById(int id)
{

    var found = context.DeptMasters.ToList().Find(e => e.DeptCode == id);
    return found;

}
```

```
public void PostDept(DeptMaster d)
{
```

```

        context.DeptMasters.Add(d);
        context.SaveChanges();
    }

    public void PutDept(int id, DeptMaster d)
    {
        var found = context.DeptMasters.ToList().Find(a => a.DeptCode == id);
        context.DeptMasters.Remove(found);
        context.DeptMasters.Add(d);
        context.SaveChanges();
    }

    public void DelDept(int id)
    {
        var found = context.DeptMasters.ToList().Find(a => a.DeptCode == id);
        context.DeptMasters.Remove(found);
        context.SaveChanges();
    }
}
}

```

EmpManagementInitializer.cs : for seeding the database tables with initial data

```

using System;
using System.Collections.Generic;
using System.Data.Entity;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

```



```
namespace DAL
```

```
{
```

```
    public class EmpManagementInitializer :  
        DropCreateDatabaseIfModelChanges<EmpManagementContext>
```

```
    {
```

```
        protected override void Seed(EmpManagementContext context)
```

```
        {
```

```
            // adding values to deptmaster table
```

```
            List<DeptMaster> deptMasterList = new List<DeptMaster>();
```

```
            deptMasterList.Add(new DeptMaster() { DeptCode = 1, DeptName = "HR" });
```

```
            deptMasterList.Add(new DeptMaster() { DeptCode = 2, DeptName = "Marketing" });
```

```
            deptMasterList.Add(new DeptMaster() { DeptCode = 3, DeptName = "Product Management"  
        });
```

```
            context.DeptMasters.AddRange(deptMasterList);
```

```
            List<EmpProfile> empProfileList = new List<EmpProfile>();
```

```
            empProfileList.Add(new EmpProfile() { DeptCode = 1, EmpName = "Ganesh", DateOfBirth =  
            new DateTime(2000, 01, 10), Email = "ganesh@gmail.com", EmpCode = 1000 });
```

```
            empProfileList.Add(new EmpProfile() { DeptCode = 2, EmpName = "Siva", DateOfBirth = new  
            DateTime(1999, 02, 15), Email = "siva21@gmail.com", EmpCode = 1001 });
```

```
            empProfileList.Add(new EmpProfile() { DeptCode = 3, EmpName = "Ram", DateOfBirth = new  
            DateTime(2002, 07, 01), Email = "ram34@gmail.com", EmpCode = 1002 });
```

```
            context.EmpProfiles.AddRange(empProfileList);
```

```
            context.SaveChanges();
```

```

        base.Seed(context);
    }

}

}

```

EmpManagementContext.cs : Database class with codefirst

```

using System;
using System.Data.Entity;
using System.Linq;

namespace DAL
{
    public class EmpManagementContext : DbContext
    {

        public EmpManagementContext()
            : base("name=EmpManagement") => Database.SetInitializer(new
EmpManagementInitializer());

        public virtual DbSet<EmpProfile> EmpProfiles { get; set; }
        public virtual DbSet<DeptMaster> DeptMasters { get; set; }

    }
}

```

