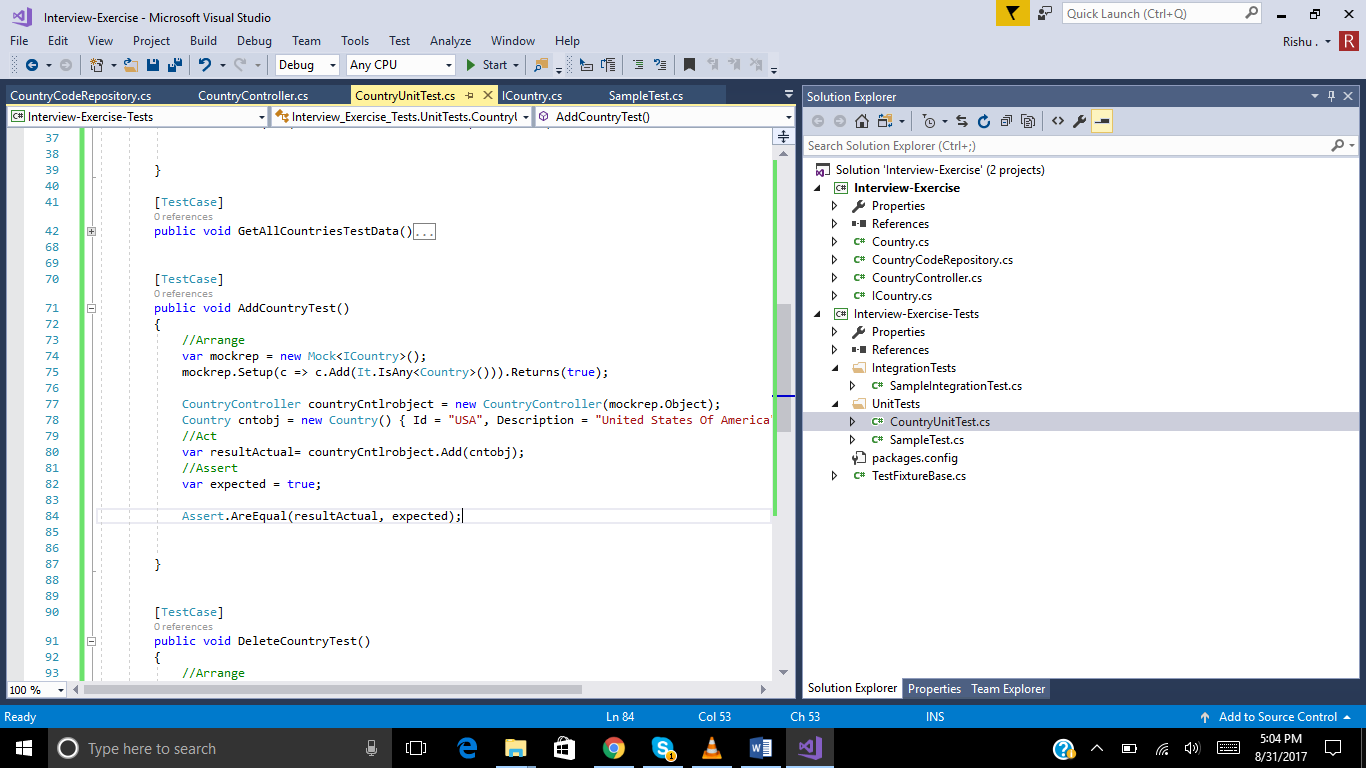


Project structure:



Files

Country.cs.

using System.Collections.Generic;

namespace Interview\_Exercise

{

public class Country

{

public string Id { get; set; }

public string Description { get; set; }

}

}

ICountry.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Interview\_Exercise

{

/// <summary>

/// To Perfom CRUD operation on Country table.

/// </summary>

public interface ICountry

{

List<Country> GetAllCountry();

bool Add(Country countobj);

bool Delete(string cntryId);

}

}

using System;

using System.Collections.Generic;

using System.Linq;

namespace Interview\_Exercise

{

/// <summary>

///

/// </summary>

public class CountryCodeRepository: ICountry

{

public List<Country> lstCountries;

public CountryCodeRepository()

{

lstCountries= new List<Country>()

{

new Country(){ Id="USA", Description="United States" },

new Country(){ Id="UK", Description="United KingDom" },

new Country(){ Id="IND", Description="India" },

new Country(){ Id="AUS", Description="Australia" },

};

}

public List<Country> GetAllCountry()

{

//Create DataContext Class object

//CountryEntities dbcontext=new CountryEntities();

//Linq query to query to database to get country objects

//var countrylist= from country in dbcontext.Countries

//select Country;

//return countryList;

//Hard coded values without database

return lstCountries;

}

public bool Add(Country countobj)

{

//Create DataContext Class object

//CountryEntities dbcontext=new CountryEntities();

//Linq query to query to database to get country objects

//dbcontext.countries.Add(countryobj);

//dbcontext.savechanges();

//return true/false

//Hardcoded values without database

lstCountries.Add(countobj);

return true;

}

public bool Delete(string cntryId)

{

//Create DataContext Class object

//CountryEntities dbcontext=new CountryEntities();

//Linq query to find object from databse based on countryid

//var countryobj=(from country in dbcontext.Countries

//select country).FirstOrDefault();

//dbcontext.countries.Remove(countryobj);

//dbcontext.savechanges();

//Hardcoded values without database

lstCountries.RemoveAll(a => a.Id == cntryId);

return true;

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Interview\_Exercise

{

public class CountryController

{

public ICountry icountryrepobj;

//Construcotr interface injection using UNITY

public CountryController(ICountry icountryrepobj2)

{

icountryrepobj = icountryrepobj2;

}

public List<Country> GetAllCountries()

{

var countryList = icountryrepobj.GetAllCountry();

return countryList;

}

public bool Add(Country cntobj)

{

var result= icountryrepobj.Add(cntobj);

return result;

}

public bool Delete(string countryid)

{

var result = icountryrepobj.Delete(countryid);

return result;

}

}

}

Unit test cases:

using Interview\_Exercise;

using Moq;

using NUnit.Framework;

using System.Collections.Generic;

namespace Interview\_Exercise\_Tests.UnitTests

{

[TestFixture]

public class CountryUnitTest

{

[TestCase]

public void GetAllCountriesCount()

{

//Arrange

var mockrep = new Mock<ICountry>();

mockrep.Setup(c => c.GetAllCountry()).Returns(

new List<Country>()

{

new Country(){ Id="USA", Description="United States" },

new Country(){ Id="UK", Description="United KingDom" },

new Country(){ Id="IND", Description="India" },

new Country(){ Id="AUS", Description="Australia" },

}

);

CountryController countryCntlrobject = new CountryController(mockrep.Object);

//Act

var resultlstActual= countryCntlrobject.GetAllCountries();

var expectedCount = 4;

//Assert

Assert.AreEqual(resultlstActual.Count, expectedCount);

}

[TestCase]

public void GetAllCountriesTestData()

{

//Arrange

var mockrep = new Mock<ICountry>();

mockrep.Setup(c => c.GetAllCountry()).Returns(

new List<Country>()

{

new Country(){ Id="USA", Description="United States" },

new Country(){ Id="UK", Description="United KingDom" }

}

);

CountryController countryCntlrobject = new CountryController(mockrep.Object);

List<Country> lstexpected = new List<Country>()

{

new Country(){ Id="USA", Description="United States" },

new Country(){ Id="UK", Description="United KingDom" }

};

//Act

var resultlstActual = countryCntlrobject.GetAllCountries();

//Assert

CollectionAssert.AreEqual(resultlstActual, lstexpected);

}

[TestCase]

public void AddCountryTest()

{

//Arrange

var mockrep = new Mock<ICountry>();

mockrep.Setup(c => c.Add(It.IsAny<Country>())).Returns(true);

CountryController countryCntlrobject = new CountryController(mockrep.Object);

Country cntobj = new Country() { Id = "USA", Description = "United States Of America" };

//Act

var resultActual= countryCntlrobject.Add(cntobj);

//Assert

var expected = true;

Assert.AreEqual(resultActual, expected);

}

[TestCase]

public void DeleteCountryTest()

{

//Arrange

var mockrep = new Mock<ICountry>();

mockrep.Setup(c => c.Delete(It.IsAny<string>())).Returns(true);

CountryController countryCntlrobject = new CountryController(mockrep.Object);

//Act

var resultActual = countryCntlrobject.Delete("USA");

//Assert

var expected = true;

Assert.AreEqual(resultActual, expected);

}

}

}