

```

using Microsoft.VisualStudio.TestTools.UnitTesting;
using CupPlaner.Controllers;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Web.Mvc;

namespace CupPlaner.Controllers.Tests
{
    [TestClass()]
    public class TeamControllerTests
    {
        TeamController controller = new TeamController();

        [TestMethod()]
        public void CreateTest()
        {
            //Create a new team
            dynamic jsonResult = ((JsonResult)controller.Create("Test Team",
ID.PoolId)).Data;
            ID.TeamId = jsonResult.id;
            Assert.AreEqual("success", jsonResult.status);

            //Create a new team using null values
            jsonResult = ((JsonResult)controller.Create(null, ID.PoolId)).Data;
            Assert.AreEqual("error", jsonResult.status);

            //Create a new team, but to a non-existing pool
            jsonResult = ((JsonResult)controller.Create("Test Team", 999999)).Data;
            Assert.AreEqual("error", jsonResult.status);
        }

        [TestMethod()]
        public void DetailsTest()
        {
            //Find the created team
            dynamic jsonResult = ((JsonResult)controller.Details(ID.TeamId)).Data;
            Assert.AreEqual("success", jsonResult.status);
            Assert.AreEqual(ID.TeamId, jsonResult.Id);
            Assert.AreEqual("Test Team", jsonResult.Name);

            //Find a team that does not exist
            jsonResult = ((JsonResult)controller.Details(999999)).Data;
            Assert.AreEqual("error", jsonResult.status);
        }

        [TestMethod()]
        public void EditTest()
        {
            //Edit the created team
            dynamic jsonResult = ((JsonResult)controller.Edit(ID.TeamId, "Test Team2",
ID.PoolId, new List<DateTime>(), new List<DateTime>())).Data;
            Assert.AreEqual("success", jsonResult.status);

            //Check to see if edits have been saved
            jsonResult = ((JsonResult)controller.Details(ID.TeamId)).Data;

```

```

        Assert.AreEqual(ID.TeamId, jsonResult.Id);
        Assert.AreEqual("Test Team2", jsonResult.Name);

        //Edit a team using null values
        jsonResult = ((JsonResult)controller.Edit(ID.TeamId, null , ID.PoolId, new
List<DateTime>(), new List<DateTime>())).Data;
        Assert.AreEqual("error", jsonResult.status);
        jsonResult = ((JsonResult)controller.Edit(ID.TeamId, "Test Team2", ID.PoolId,
null, new List<DateTime>())).Data;
        Assert.AreEqual("error", jsonResult.status);

        //Edit a team that does not exist
        jsonResult = ((JsonResult)controller.Edit(999999, "Test Team2", ID.PoolId,
new List<DateTime>(), new List<DateTime>())).Data;
        Assert.AreEqual("error", jsonResult.status);
    }

    [TestMethod()]
    public void DeleteTest()
    {
        //Delete the created team
        dynamic jsonResult = ((JsonResult)controller.Delete(ID.TeamId)).Data;
        Assert.AreEqual("success", jsonResult.status);

        //Delete a team that does not exist
        jsonResult = ((JsonResult)controller.Delete(999999)).Data;
        Assert.AreEqual("error", jsonResult.status);
    }
}

```