```
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;
using System.Diagnostics;
using System.Web.Script.Serialization;

namespace CupPlaner.Controllers.Tests
{
    public class ID
    {
        public static int TournamentId = 50;
        public static int DivisionId = 223;
        public static int PoolId = 3975;
        public static int TeamId = 12150;
        public static int FieldId = 134;
        public static int FinalsLinkId = 919;
    }

    [TestClass()]
    public class TournamentControllerTests
    {
        TournamentController controller = new TournamentController();
        String startDates =  "16-11-2015 08:00:00,17-11-2015 10:00:00";
        String endDates = "16-11-2015 20:00:00,17-11-2015 22:00:00";
        List<DateTime> startDates2 = new List<DateTime>() { DateTime.Parse("16-11-2015 
09:00:00"), DateTime.Parse("17-11-2015 11:00:00") };
        List<DateTime> endDates2 = new List<DateTime>() { DateTime.Parse("16-11-2015 
21:00:00"), DateTime.Parse("17-11-2015 22:30:00") };

        [TestMethod()]
        public void CreateTest()
        {
            //Create a new tournament
            dynamic jsonResult = ((JsonResult)controller.Create("TestName", 
"TestPassword", startDates, endDates)).Data;
            ID.TournamentId = jsonResult.id;
            Assert.AreEqual("success", jsonResult.status);

            //Create a new tournament using null values
            jsonResult = ((JsonResult)controller.Create(null, "TestPassword", startDates, 
endDates)).Data;
            Assert.AreEqual("error", jsonResult.status);
            jsonResult = ((JsonResult)controller.Create("TestName", "TestPassword", null, 
endDates)).Data;
            Assert.AreEqual("error", jsonResult.status);

            //Create a new tournament with the same password
            jsonResult = ((JsonResult)controller.Create("TestName", "TestPassword", 
startDates, endDates)).Data;
            Assert.AreEqual("error", jsonResult.status);
            Assert.AreEqual("Password already exists", jsonResult.message);
        }
        [TestMethod()]
        public void IdFromPassTest()
        {
            //Find the created tournament from the password
            dynamic jsonResult =
((JsonResult)controller.IdFromPass("TestPassword")).Data;
            Assert.AreEqual(ID.TournamentId, jsonResult.Id);

            //Find a tournament using null as password
            jsonResult = ((JsonResult)controller.IdFromPass(null)).Data;
            Assert.AreEqual(0, jsonResult.Id);

            //Find a tournament that does not exist
            jsonResult =
((JsonResult)controller.IdFromPass("ATournamentThatDoesNotExist")).Data;
            Assert.AreEqual(0, jsonResult.Id);
        }

        [TestMethod()]
        public void DetailsTest()
        {
            //Find the created tournament
            dynamic jsonResult = ((JsonResult)controller.Details(ID.TournamentId)).Data;
            Assert.AreEqual("success", jsonResult.status);
            Assert.AreEqual("TestName", jsonResult.Name);
            Assert.AreEqual("TestPassword", jsonResult.Password);
            Assert.AreEqual(DateTime.Parse("16-11-2015 08:00:00"),
jsonResult.TimeIntervals[0].StartTime);
            Assert.AreEqual(DateTime.Parse("16-11-2015 20:00:00"),
jsonResult.TimeIntervals[0].EndTime);
            Assert.AreEqual(DateTime.Parse("17-11-2015 10:00:00"),
jsonResult.TimeIntervals[1].StartTime);
            Assert.AreEqual(DateTime.Parse("17-11-2015 22:00:00"),
jsonResult.TimeIntervals[1].EndTime);

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

        [TestMethod()]
        public void EditTest()
        {
            //Edit the created tournament
            dynamic jsonResult = ((JsonResult)controller.Edit(ID.TournamentId,
"TestName2", "TestPassword2", startDates2, endDates2)).Data;
            Assert.AreEqual("success", jsonResult.status);

            //Check to see if edits have been saved
            jsonResult = ((JsonResult)controller.Details(ID.TournamentId)).Data;
            Assert.AreEqual("TestName2", jsonResult.Name);
            Assert.AreEqual("TestPassword2", jsonResult.Password);
            Assert.AreEqual(DateTime.Parse("16-11-2015 09:00:00"),
jsonResult.TimeIntervals[0].StartTime);
            Assert.AreEqual(DateTime.Parse("16-11-2015 21:00:00"),
jsonResult.TimeIntervals[0].EndTime);
            Assert.AreEqual(DateTime.Parse("17-11-2015 11:00:00"),
jsonResult.TimeIntervals[1].StartTime);
```

```
            Assert.AreEqual(DateTime.Parse("17-11-2015 22:30:00"), 
jsonResult.TimeIntervals[1].EndTime);


            //Edit a tournament using null values
            jsonResult = ((JsonResult)controller.Edit(ID.TournamentId, "TestName2", null, 
startDates2, endDates2)).Data;
            Assert.AreEqual("error", jsonResult.status);
            jsonResult = ((JsonResult)controller.Edit(ID.TournamentId, "TestName2", 
"TestPassword2", startDates2, null)).Data;
            Assert.AreEqual("error", jsonResult.status);

            //Edit a tournament that does not exist
            jsonResult = ((JsonResult)controller.Edit(999999, "TestName2", 
"TestPassword2", startDates2, endDates2)).Data;
            Assert.AreEqual("error", jsonResult.status);
        }

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

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