```
using System;
using System.Collections.Generic;
using System.Data.Entity;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using CupPlaner.Helpers;

namespace CupPlaner.Controllers
{
    public class FinalsLinkController : Controller
    {
        CupDBContainer db = new CupDBContainer();
        ScheduleManager sm = new ScheduleManager();

        // GET: FinalsLink/Details/5 - Gets the details of the FinalsLink class with its 
id as a parameter. The id are used to get to the specific FinalsLink object.
        // The Details function returns a JSON object, containing a copy of the 
FinalsLinks varibles.
        // If the function runs successfully, the object are returned with a "success" 
status.
        // If the function do not run successfully, an "error" status is returned 
instead.
        public ActionResult Details(int id)
        {
            try
            {
                FinalsLink fl = db.FinalsLinkSet.Find(id);

                object obj = new { status = "success", id = fl.Id, Finalstage = 
fl.Finalstage, PoolPlacement = fl.PoolPlacement, Division_Id = fl.Division };

                return Json(obj, JsonRequestBehavior.AllowGet);
            }
            catch (Exception ex)
            {
                return Json(new { status = "error", message = "Could not find finals 
link", details = ex.Message }, JsonRequestBehavior.AllowGet);
            }
        }

        // POST: FinalsLink/Create - The Create function will try and create a new 
FinalsLink object with the parameters: finalStage, poolPlacement and divisionId.
        // The function will add the FinalsLink object to the FinalsLinkSet and saves it 
to the database.
        // Returns a Json object with a state, indicating whether it succeeded creating 
the Division object or not.
        [HttpPost]
        public ActionResult Create(int finalStage, int poolPlacement, int divisionID)
        {
            try
            {
                Division d = db.DivisionSet.Find(divisionID);

                sm.DeleteSchedule(d.Tournament.Id, db);

                db.FinalsLinkSet.Add(new FinalsLink { Finalstage = finalStage, 
PoolPlacement = poolPlacement, Division = d });
```

```
                db.SaveChanges();

                return Json(new { status = "success", message = "New finals link added",
id = d.Id }, JsonRequestBehavior.AllowGet);
            }
            catch (Exception ex)
            {
                return Json(new { status = "error", message = "New finals link not
added", details = ex.Message }, JsonRequestBehavior.AllowGet);
            }
        }

        // POST: FinalsLink/Edit/5 - The Edit function will try to change the values of
the specific FinalsLink object.
        // The Edit function can change the finalStage and the poolPlacement.
        // The function will send back a message either if the function succeded or
failed to edit the FinalsLink object.
        [HttpPost]
        public ActionResult Edit(int id, int finalStage, int poolPlacement)
        {
            try
            {
                FinalsLink fl = db.FinalsLinkSet.Find(id);
                fl.Finalstage = finalStage;
                fl.PoolPlacement = poolPlacement;

                //Clear the schedule
                sm.DeleteSchedule(fl.Division.Tournament.Id, db);

                db.Entry(fl).State = EntityState.Modified;
                db.SaveChanges();

                return Json(new { status = "success", message = "Finals link edited" },
JsonRequestBehavior.AllowGet);

            }
            catch (Exception ex)
            {
                return Json(new { status = "error", message = "Finals link not edited",
details = ex.Message }, JsonRequestBehavior.AllowGet);
            }
        }

        // POST: FinalsLink/Delete/5 - The delete function will try to delete a specific
FinalsLink with its corresponding id.
        // It will delete all FinalsLink and save it to the databse if succeeded.
        // The function will return a JSON object which will indicate if the Finalsink
was deleted or not.
        [HttpPost]
        public ActionResult Delete(int id)
        {
            try
            {
                FinalsLink fl = db.FinalsLinkSet.Find(id);

                //Clear the schedule
                sm.DeleteSchedule(fl.Division.Tournament.Id, db);
                db.FinalsLinkSet.Remove(fl);
                db.SaveChanges();

                return Json(new { status = "success", message = "Finals link deleted" },
JsonRequestBehavior.AllowGet);
            }
            catch (Exception ex)
            {
                return Json(new { status = "error", message = "Finals link not deleted",
details = ex.Message }, JsonRequestBehavior.AllowGet);
            }
        }
    }
}
```