

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using CupPlaner.Helpers;

namespace CupPlaner.Controllers
{
    public class MatchGenerationController : Controller
    {
        MatchGeneration mg = new MatchGeneration();

        // These functions call functions from the matchGeneration class and returns a
        // json object to tell the frontend whether or not it was successfull
        public ActionResult GenerateGroupStage(int tournamentID)
        {
            try
            {
                if (mg.GenerateGroupStage(tournamentID))
                {
                    return Json(new { status = "success" },
                        JsonRequestBehavior.AllowGet);
                }
                else
                {
                    return Json(new { status = "error", message = "Generering
mislykkedes" }, JsonRequestBehavior.AllowGet);
                }
            }
            catch (Exception)
            {
                return Json(new { status = "error", message = "fejl i programmet" },
                    JsonRequestBehavior.AllowGet);
            }
        }

        public ActionResult GenerateFinalsTeams(int tournamentID)
        {
            try
            {
                if (mg.GenerateFinalsTeams(tournamentID))
                {
                    return Json(new { status = "success" },
                        JsonRequestBehavior.AllowGet);
                }
                else
                {
                    return Json(new { status = "error", message = "Generering
mislykkedes" }, JsonRequestBehavior.AllowGet);
                }
            }
            catch (Exception)
            {
                return Json(new { status = "error", message = "fejl i programmet" },
                    JsonRequestBehavior.AllowGet);
            }
        }
    }
}

```

```

public ActionResult GenerateFinalsMatches(int tournamentID)
{
    try
    {
        if (mg.GenerateFinalsMatches(tournamentID))
        {
            return Json(new { status = "success" },
                JsonRequestBehavior.AllowGet);
        }
        else
        {
            return Json(new { status = "error", message = "Generering
mislykkedes" }, JsonRequestBehavior.AllowGet);
        }
    }
    catch (Exception)
    {
        return Json(new { status = "error", message = "fejl i programmet" },
            JsonRequestBehavior.AllowGet);
    }
}
}

```