```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Net;
using System.Text;
using System.Threading.Tasks;
using Newtonsoft.Json;
using DAL.apiClases;

namespace DAL
{
    public class APICall
    {
        public static List<Root> GetCall()
        {
            ServicePointManager.Expect100Continue = true;
            ServicePointManager.SecurityProtocol = SecurityProtocolType.Tls12;
            WebRequest request = HttpWebRequest.Create(@"https://fantasy.premierleague.com/api/fixtures/");
            WebResponse responce = request.GetResponse();
            StreamReader reader = new StreamReader(responce.GetResponseStream());
            string football_Jason = reader.ReadToEnd();

            var call = JsonConvert.DeserializeObject<List<Root>>(football_Jason);
            return call;
        }
        public static Dictionary<string, Element> getListOfStats(List<string> ids)
        {

            ServicePointManager.Expect100Continue = true;
            ServicePointManager.SecurityProtocol = SecurityProtocolType.Tls12;
            WebRequest request = HttpWebRequest.Create(@"https://fantasy.premierleague.com/api/bootstrap-static/");
            WebResponse responce = request.GetResponse();
            StreamReader reader = new StreamReader(responce.GetResponseStream());
            string football_Jason = reader.ReadToEnd();

            Dictionary<string, Element> dic = new Dictionary<string, Element>();

            var call = JsonConvert.DeserializeObject<BooStat>(football_Jason);
            foreach (Element eve in call.elements)
            {
                foreach (string id in ids)
                {
                    if (eve.id == Convert.ToInt32(id))
                    {
                        dic.Add(id, eve);
                    }
                }
            }
            return dic;
        }
        public static int getCurrentGameweek(int offset)
        {
            ServicePointManager.Expect100Continue = true;
            ServicePointManager.SecurityProtocol = SecurityProtocolType.Tls12;
            WebRequest request = HttpWebRequest.Create(@"https://fantasy.premierleague.com/api/bootstrap-static/");
            WebResponse responce = request.GetResponse();
            StreamReader reader = new StreamReader(responce.GetResponseStream());
            string football_Jason = reader.ReadToEnd();

            var call = JsonConvert.DeserializeObject<BooStat>(football_Jason);
            foreach (Event eve in call.events)
            {
                if (eve.finished == false)
                {
                    if (eve.id is int)
                    {
                        return eve.id + offset?? 1;
                    }

                }
            }
            return 1;
        }
        public static List<Root> sortbyNextGameWeek(List<Root> lr, int currGW)
        {
            List<Root> ret = new List<Root>();
            foreach (Root fixture in lr)
            {
                if (fixture.@event == currGW)
                {
                    ret.Add(fixture);
                }
            }
            return ret;
        }
        private static bool DateInsideOneWeek(DateTime date1, DateTime date2)
        {
            DayOfWeek firstDayOfWeek = System.Globalization.CultureInfo.CurrentCulture.DateTimeFormat.FirstDayOfWeek;
            DateTime startDateOfWeek = date1.Date;
            while (startDateOfWeek.DayOfWeek != firstDayOfWeek)
            { startDateOfWeek = startDateOfWeek.AddDays(-1d); }
            DateTime endDateOfWeek = startDateOfWeek.AddDays(6d);
            return date2 >= startDateOfWeek && date2 <= endDateOfWeek;
        }
        public static void addGamesToDB()
        {
            #region connecttoDb
            ServicePointManager.Expect100Continue = true;
            ServicePointManager.SecurityProtocol = SecurityProtocolType.Tls12;
            WebRequest request = HttpWebRequest.Create(@"https://
              fantasy.premierleague.com/api/fixtures/");
            WebResponse responce = request.GetResponse();
            StreamReader reader = new StreamReader(responce.GetResponseStream());
            string football_Jason = reader.ReadToEnd();

            List<Root> call = JsonConvert.DeserializeObject<List<Root>>
              (football_Jason);
            #endregion

            foreach (Root game in call)
            {
                string com = "UPDATE game SET [date] = '"+ (game.kickoff_time ??
                  DateTime.Now.AddYears(-1000)).ToString("dddd, dd MMMM h:mm tt")
                   +"', homes = '"+game.team_h_score+"', ascore
                  ='"+game.team_a_score+"' Where gameID = "+game.id;
                oledbhelper.Execute(com);
            }
        }
        public static void addALLGamesToDB()
        {
            //done once to add all games to database
            #region connecttoDb
            ServicePointManager.Expect100Continue = true;
            ServicePointManager.SecurityProtocol = SecurityProtocolType.Tls12;
            WebRequest request = HttpWebRequest.Create(@"https://
              fantasy.premierleague.com/api/fixtures/");
            WebResponse responce = request.GetResponse();
            StreamReader reader = new StreamReader(responce.GetResponseStream());
            string football_Jason = reader.ReadToEnd();

            List<Root> call = JsonConvert.DeserializeObject<List<Root>>
              (football_Jason);
            #endregion

            foreach (Root game in call)
            {
                string com = "INSERT INTO game (gameID,gw,hteam,ateam,
                  [date],homes,ascore) VALUES("+game.id+",'"+game.@event
                  +"','"+game.team_h+"','"+game.team_a+"','" +
                  (game.kickoff_time ?? DateTime.Now.AddYears(-1000)).ToString
                  ("dddd, dd MMMM h:mm tt") + "','"+game.team_h_score
                  +"','"+game.team_a_score+"')";
                oledbhelper.Execute(com);
            }
        }
        public static int getCurrentGw()
        {
            ServicePointManager.Expect100Continue = true;
            ServicePointManager.SecurityProtocol = SecurityProtocolType.Tls12;
            WebRequest request = HttpWebRequest.Create(@"https://fantasy.premierleague.com/api/bootstrap-static/");
            WebResponse responce = request.GetResponse();
            StreamReader reader = new StreamReader(responce.GetResponseStream());
            string football_Jason = reader.ReadToEnd();

            var call = JsonConvert.DeserializeObject<Root2>(football_Jason);


            foreach (Event2 gw in call.events)
            {
                if ( DateTime.Now.CompareTo(gw.deadline_time) < 0)
                {
                    return gw.id;
                }
            }
            return 1;
        }
    }
}

```