```
1  using System;
2  using System.Collections.Generic;
3  using System.IO;
4  using System.Linq;
5  using System.Net;
6  using System.Text;
7  using System.Threading.Tasks;
8  using Newtonsoft.Json;
9  using DAL.apiClases;
10
11 namespace DAL
12 {
13     public class APICall
14     {
15         public static List<Root> GetCall()
16         {
17             ServicePointManager.Expect100Continue = true;
18             ServicePointManager.SecurityProtocol = SecurityProtocolType.Tls12;
19             WebRequest request = HttpWebRequest.Create(@"https://
                 fantasy.premierleague.com/api/fixtures/");
20             WebResponse responce = request.GetResponse();
21             StreamReader reader = new StreamReader(responce.GetResponseStream());
22             string football_Jason = reader.ReadToEnd();
23
24             var call = JsonConvert.DeserializeObject<List<Root>>(football_Jason);
25             return call;
26         }
27         public static Dictionary<string, Element> getListOfStats(List<string>
             ids)
28         {
29
30             ServicePointManager.Expect100Continue = true;
31             ServicePointManager.SecurityProtocol = SecurityProtocolType.Tls12;
32             WebRequest request = HttpWebRequest.Create(@"https://
                 fantasy.premierleague.com/api/bootstrap-static/");
33             WebResponse responce = request.GetResponse();
34             StreamReader reader = new StreamReader(responce.GetResponseStream());
35             string football_Jason = reader.ReadToEnd();
36
37             Dictionary<string, Element> dic = new Dictionary<string, Element>();
38
39             var call = JsonConvert.DeserializeObject<BooStat>(football_Jason);
40             foreach (Element eve in call.elements)
41             {
42                 foreach (string id in ids)
43                 {
44                     if (eve.id == Convert.ToInt32(id))
45                     {
46                         dic.Add(id, eve);
47                     }
48                 }
49             }
```

```
50              return dic;
51          }
52          public static int getCurrentGameweek(int offset)
53          {
54              ServicePointManager.Expect100Continue = true;
55              ServicePointManager.SecurityProtocol = SecurityProtocolType.Tls12;
56              WebRequest request = HttpWebRequest.Create(@"https://
                  fantasy.premierleague.com/api/bootstrap-static/");
57              WebResponse responce = request.GetResponse();
58              StreamReader reader = new StreamReader(responce.GetResponseStream());
59              string football_Jason = reader.ReadToEnd();
60
61              var call = JsonConvert.DeserializeObject<BooStat>(football_Jason);
62              foreach (Event eve in call.events)
63              {
64                  if (eve.finished == false)
65                  {
66                      if (eve.id is int)
67                      {
68                          return eve.id + offset?? 1;
69                      }
70
71                  }
72              }
73              return 1;
74          }
75          public static List<Root> sortbyNextGameWeek(List<Root> lr, int currGW)
76          {
77              List<Root> ret = new List<Root>();
78              foreach (Root fixture in lr)
79              {
80                  if (fixture.@event == currGW)
81                  {
82                      ret.Add(fixture);
83                  }
84              }
85              return ret;
86          }
87          private static bool DateInsideOneWeek(DateTime date1, DateTime date2)
88          {
89              DayOfWeek firstDayOfWeek =
                    System.Globalization.CultureInfo.CurrentCulture.DateTimeFormat.Firs
                    tDayOfWeek;
90              DateTime startDateOfWeek = date1.Date;
91              while (startDateOfWeek.DayOfWeek != firstDayOfWeek)
92              { startDateOfWeek = startDateOfWeek.AddDays(-1d); }
93              DateTime endDateOfWeek = startDateOfWeek.AddDays(6d);
94              return date2 >= startDateOfWeek && date2 <= endDateOfWeek;
95          }
96          public static void addGamesToDB()
97          {
98              #region connecttoDb
```

```
 99              ServicePointManager.Expect100Continue = true;
100              ServicePointManager.SecurityProtocol = SecurityProtocolType.Tls12;
101              WebRequest request = HttpWebRequest.Create(@"https://
                   fantasy.premierleague.com/api/fixtures/");
102              WebResponse responce = request.GetResponse();
103              StreamReader reader = new StreamReader(responce.GetResponseStream());
104              string football_Jason = reader.ReadToEnd();
105
106              List<Root> call = JsonConvert.DeserializeObject<List<Root>>
                   (football_Jason);
107              #endregion
108
109              foreach (Root game in call)
110              {
111                  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;
112                  oledbhelper.Execute(com);
113              }
114          }
115          public static void addALLGamesToDB()
116          {
117              //done once to add all games to database
118              #region connecttoDb
119              ServicePointManager.Expect100Continue = true;
120              ServicePointManager.SecurityProtocol = SecurityProtocolType.Tls12;
121              WebRequest request = HttpWebRequest.Create(@"https://
                   fantasy.premierleague.com/api/fixtures/");
122              WebResponse responce = request.GetResponse();
123              StreamReader reader = new StreamReader(responce.GetResponseStream());
124              string football_Jason = reader.ReadToEnd();
125
126              List<Root> call = JsonConvert.DeserializeObject<List<Root>>
                   (football_Jason);
127              #endregion
128
129              foreach (Root game in call)
130              {
131                  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+"')";
132                  oledbhelper.Execute(com);
133              }
134          }
135          public static int getCurrentGw()
136          {
137              ServicePointManager.Expect100Continue = true;
138              ServicePointManager.SecurityProtocol = SecurityProtocolType.Tls12;
```

```
139             WebRequest request = HttpWebRequest.Create(@"https://
                  fantasy.premierleague.com/api/bootstrap-static/");
140             WebResponse responce = request.GetResponse();
141             StreamReader reader = new StreamReader(responce.GetResponseStream());
142             string football_Jason = reader.ReadToEnd();
143
144             var call = JsonConvert.DeserializeObject<Root2>(football_Jason);
145
146
147             foreach (Event2 gw in call.events)
148             {
149                 if ( DateTime.Now.CompareTo(gw.deadline_time) < 0)
150                 {
151                     return gw.id;
152                 }
153             }
154             return 1;
155         }
156     }
157 }
158
```