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

namespace cymax_con
{
   class cymax
   {
      public decimal amt;

   }
   class Program
   {
      public static void Main(string[] args)
      {
         string sourceAddress, destinationAddress, dimensions, param1, param2, param3, url, type;
         string[] str , str1;
         sourceAddress = destinationAddress = dimensions = param1 = param2 = param3 = url = type =
string.Empty;

         int noOfCompanies = 0; decimal amt = 0 , minAmt = -1;

         MakeHttpRequest makeHttp = new MakeHttpRequest();


         Console.WriteLine("Enter source, destination and dimensions : "); //getting source, destination and
dimension
         str = Console.ReadLine().Split(' ');
         sourceAddress = str[0];
         destinationAddress = str[1];
         dimensions = str[2];

         Console.WriteLine("Enter no. of companies : ");  //getting companies
         noOfCompanies = Int32.Parse(Console.ReadLine());

         for (int i = 0; i < noOfCompanies; i++)
         {
            Console.WriteLine("Enter URL : "); //getting url
            url = Console.ReadLine();

            Console.WriteLine("Enter Type : ");
            type = Console.ReadLine();
```

```
            Console.WriteLine("Enter parameter names : "); //getting different company parameters for
source, destination and dimensions
            str1 = Console.ReadLine().Split(' ');
            param1 = str1[0];
            param2 = str1[1];
            param3 = str1[2];

            amt = makeHttp.RequestAPI(url, type, param1, sourceAddress, param2, destinationAddress,
param3, dimensions);


            if (minAmt < 0)
               minAmt = amt;
               if (amt < minAmt && amt>=0)
               minAmt = amt;
                     }
         Console.WriteLine("Best deal {0}", minAmt);

         Console.ReadLine();
      }
   }

   class MakeHttpRequest
   {
      public decimal RequestAPI(string url, string type, string param1, string param1value, string param2,
string param2value, string param3, string param3value)


 {
         decimal amt=0;
         if (type == "json")
         {
            try
            {
            var httpWebRequest = (HttpWebRequest)WebRequest.Create(url);
            httpWebRequest.ContentType = "application/json";
            httpWebRequest.Method = "POST";
            using (var streamWriter = new

            StreamWriter(httpWebRequest.GetRequestStream()))
            {
              string json = "{\"" + param1 + "\": \"" + param1value + "\",\"" + param2 + "\":\"" + param2value +
"\",\"" + param3 + "\":\"" + param3value + "\"}";

               streamWriter.Write(json);
            }

            var httpResponse = (HttpWebResponse)httpWebRequest.GetResponse();
               if (httpResponse.StatusCode == HttpStatusCode.OK)
```

```
            {
                using (var streamReader = new StreamReader(httpResponse.GetResponseStream()))
                {
                    var result = streamReader.ReadToEnd();
                    amt = JsonConvert.DeserializeObject<cymax>(result).amt;

                }
            }
            else
            {
                amt=-1;
            }
        }
        catch (Exception e)
        {

            amt = -1;

           // Console.WriteLine(e);
        }
            }
      else if (type == "xml")
      {
          try
          {
              string xml = @"<?xml version=""1.0""
encoding=""UTF-8""?><param1>param1value</param1><param2>param2value</param2><param3>para
m3value</param3>";
              byte[] requestBytes = System.Text.Encoding.ASCII.GetBytes(xml);
              var httpWebRequest = (HttpWebRequest)WebRequest.Create(url);
              httpWebRequest.ContentType = "text/xml;charset=utf-8";
              httpWebRequest.Method = "POST";
              httpWebRequest.ContentLength = requestBytes.Length;
              Stream requestStream = httpWebRequest.GetRequestStream();
              requestStream.Write(requestBytes, 0, requestBytes.Length);
              requestStream.Close();



              var httpResponse = (HttpWebResponse)httpWebRequest.GetResponse();
              if (httpResponse.StatusCode == HttpStatusCode.OK)
              {
                  using (var streamReader = new StreamReader(httpResponse.GetResponseStream(),
System.Text.Encoding.Default))
                  {
                      var result = streamReader.ReadToEnd();
                      amt = JsonConvert.DeserializeObject<cymax>(result).amt;
                      //  return Int32.Parse(result);
                  }
```

```
                }
                else
                {
                    amt = -1;
                }
            }
            catch (Exception e)
            {
                amt = -1;
                //Console.WriteLine(e);
            }
        }
        return amt;
    }

  }
}
```