```
 1  using System;
 2  using System.Collections.Generic;
 3  using System.Linq;
 4  using System.Net.Http;
 5  using System.Text;
 6  using System.Threading.Tasks;
 7
 8  namespace HttpClientProject
 9  {
10      class Program
11      {
12          static void Main(string[] args)
13          {
14              ShowMenu();
15          }
16
17          public static void ShowMenu()
18          {
19              int selection;
20              string options = "1.GET" +
21                  "\n2.POST" +
22                  "\n3.PUT" +
23                  "\n4.DELETE" +
24                  "\n5. Quit program";
25              Console.WriteLine(options);
26              Console.Write("\nEnter an option: ");
27              selection = Int32.Parse(Console.ReadLine());
28              while (selection != 5)
29              {
30                  switch (selection)
31                  {
32                      case 1:
33                          GetStudents();
34                          break;
35                      case 2:
36                          PostStudent();
37                          break;
38                      case 3:
39                          PutStudent();
40                          break;
41                      case 4:
42                          DeleteStudent();
43                          break;
44                      default:
45                          break;
46                  }
47                  Console.WriteLine("\n" + options);
48                  Console.Write("\nEnter an option: ");
49                  selection = Int32.Parse(Console.ReadLine());
```

```
50              }
51          }
52
53          public static void GetStudents()
54          {
55              using (var client = new HttpClient())
56              {
57                  client.BaseAddress = new Uri("http://localhost:60260/api/");
58                  //HTTP GET
59                  var responseTask = client.GetAsync("student");
60                  responseTask.Wait();
61
62                  var result = responseTask.Result;
63                  if (result.IsSuccessStatusCode)
64                  {
65
66                      var readTask = result.Content.ReadAsAsync<Student[]>();
67                      readTask.Wait();
68
69                      var students = readTask.Result;
70                      foreach (var student in students)
71                      {
72                          Console.WriteLine("{0} \t {1}", student.Id,
                        student.Name);
73                      }
74                  }
75              }
76          }
77
78          public static void PostStudent()
79          {
80              Console.Write("\nEnter the name for a new student: ");
81              string studentName = Console.ReadLine();
82              var student = new Student() { Name = studentName };
83              using (var client = new HttpClient())
84              {
85                  client.BaseAddress = new Uri("http://localhost:60260/api/");
86                  var postTask = client.PostAsJsonAsync<Student>("student",
                      student);
87                  postTask.Wait();
88                  var result = postTask.Result;
89                  if (result.IsSuccessStatusCode)
90                  {
91                      Student selectedStudent = null;
92                      //HTTP GET
93                      var responseTask = client.GetAsync("student");
94                      responseTask.Wait();
95
96                      var getResult = responseTask.Result;
```

```
97                      if (getResult.IsSuccessStatusCode)
98                      {
99
100                         var readTask = getResult.Content.ReadAsAsync<Student[]>
                        ();
101                         readTask.Wait(TimeSpan.FromSeconds(3));
102
103                         var students = readTask.Result;
104                         selectedStudent = students.Last();
105                     }
106                     Console.WriteLine("Student with id {0} and name {1} has
                        been added", selectedStudent.Id, selectedStudent.Name);
107                 }
108                 else
109                 {
110                     Console.WriteLine(result.StatusCode);
111                 }
112             }
113         }
114
115         public static void PutStudent()
116         {
117             Console.Write("\nEnter the id of the student you would like to
                  edit: ");
118             int selectedID = ValidInt();
119             Student selectedStudent = GetSingleStudent(selectedID);
120             if (selectedStudent == null)
121             {
122                 Console.WriteLine("A student with that ID does not exist");
123             }
124             else
125             {
126                 Console.WriteLine("Current student values: ID={0} \t Name={1}",
                       selectedStudent.Id, selectedStudent.Name);
127                 using (var client = new HttpClient())
128                 {
129                     client.BaseAddress = new Uri("http://localhost:60260/
                        api/");
130                     Console.Write("\nEnter the new name for this student: ");
131                     selectedStudent.Name = Console.ReadLine();
132                     var putTask = client.PutAsJsonAsync("student/",
                        selectedStudent);
133                     var result = putTask.Result;
134                     if (result.IsSuccessStatusCode)
135                     {
136                         Console.WriteLine("New student values: ID={0} \t Name=
                        {1}", selectedStudent.Id, selectedStudent.Name);
137                     }
138                     else
```

```
139                     {
140                         Console.WriteLine("PUT failed");
141                     }
142                 }
143             }
144         }
145
146         public static void DeleteStudent()
147         {
148             Console.Write("\nEnter the id of the student you would like to
                  delete: ");
149             int selectedID = ValidInt();
150             Student selectedStudent = GetSingleStudent(selectedID);
151             if (selectedStudent == null)
152             {
153                 Console.WriteLine("A student with that ID does not exist");
154             }
155             else
156             {
157                 string selectedName = selectedStudent.Name;
158                 using (var client = new HttpClient())
159                 {
160                     client.BaseAddress = new Uri("http://localhost:60260/
                        api/");
161                     var deleteTask = client.DeleteAsync("student/" +
                        selectedStudent.Id);
162                     deleteTask.Wait();
163                     var result = deleteTask.Result;
164                     if (result.IsSuccessStatusCode)
165                     {
166                         Console.WriteLine("Student (ID={0} \t Name={1}) has
                        been deleted.", selectedID, selectedName);
167                     }
168                     else
169                     {
170                         Console.WriteLine("Student with that name does not
                        exist or was not deleted.");
171                     }
172                 }
173             }
174         }
175
176         public static Student GetSingleStudent(int id)
177         {
178             using (var client = new HttpClient())
179             {
180                 client.BaseAddress = new Uri("http://localhost:60260/api/");
181                 //HTTP GET
182                 var responseTask = client.GetAsync("student");
```

```
183                 responseTask.Wait();
184
185                 var result = responseTask.Result;
186                 if (result.IsSuccessStatusCode)
187                 {
188
189                     var readTask = result.Content.ReadAsAsync<Student[]>();
190                     readTask.Wait();
191
192                     var students = readTask.Result;
193                     foreach (var student in students)
194                     {
195                         if (student.Id == id)
196                         {
197                             return student;
198                         }
199                     }
200                 }
201             }
202             return null;
203         }
204
205         public static int ValidInt()
206         {
207             int result;
208             String input = Console.ReadLine();
209             while (Int32.TryParse(input, out result) != true)
210             {
211                 Console.Write("Invalid input. Enter a valid integer: ");
212                 input = Console.ReadLine();
213             }
214             return result;
215         }
216     }
217 }
218
```