

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Web;
5 using System.Web.Mvc;
6 using Assignment7.Models;
7 using System.Net.Http;
8
9 namespace Assignment7.Controllers
10 {
11     public class StudentController : Controller
12     {
13         // GET: Student
14         public ActionResult Index()
15         {
16             IEnumerable<StudentViewModel> students = null;
17
18             using (var client = new HttpClient())
19             {
20                 client.BaseAddress = new Uri("http://localhost:60260/api/");
21                 //HTTP GET
22                 var responseTask = client.GetAsync("student");
23                 responseTask.Wait();
24
25                 var result = responseTask.Result;
26                 if (result.IsSuccessStatusCode)
27                 {
28                     var readTask =
29                         result.Content.ReadAsAsync<IList<StudentViewModel>>();
30                     readTask.Wait();
31
32                     students = readTask.Result;
33                 }
34                 else //web api sent error response
35                 {
36                     //log response status here...
37
38                     students = Enumerable.Empty<StudentViewModel>();
39
40                     ModelState.AddModelError(string.Empty, "Server error.
41                         Please contact administrator.");
42                 }
43             }
44             return View(students);
45         }
46
47         public ActionResult create()
48         {
49             return View();
50         }
51     }
52 }
```

```
48     }
49
50     [HttpPost]
51     public ActionResult create(StudentViewModel student)
52     {
53         using (var client = new HttpClient())
54         {
55             client.BaseAddress = new Uri("http://localhost:60260/api/
56                                     student");
57
58             //HTTP POST
59             var postTask = client.PostAsJsonAsync<StudentViewModel>
60                 ("student", student);
61             postTask.Wait();
62
63             var result = postTask.Result;
64             if (result.IsSuccessStatusCode)
65             {
66                 return RedirectToAction("Index");
67             }
68
69             ModelState.AddModelError(string.Empty, "Server Error. Please
70                                     contact administrator.");
71
72             return View(student);
73         }
74
75     public ActionResult Edit(int id)
76     {
77         StudentViewModel student = null;
78
79         using (var client = new HttpClient())
80         {
81             client.BaseAddress = new Uri("http://localhost:60260/api/");
82             //HTTP GET
83             var responseTask = client.GetAsync("student?id=" + id.ToString
84                 ());
85             responseTask.Wait();
86
87             var result = responseTask.Result;
88             if (result.IsSuccessStatusCode)
89             {
90                 var readTask = result.Content.ReadAsAsync<StudentViewModel>
91                     ();
92                 readTask.Wait();
93
94                 student = readTask.Result;
95             }
96         }
97     }
```

```
92     }
93
94     return View(student);
95 }
96
97 [HttpPost]
98 public ActionResult Edit(StudentViewModel student)
99 {
100     using (var client = new HttpClient())
101     {
102         client.BaseAddress = new Uri("http://localhost:60260/api/
103             student");
104
105         //HTTP POST
106         var putTask = client.PutAsJsonAsync<StudentViewModel>
107             ("student", student);
108         putTask.Wait();
109
110         var result = putTask.Result;
111         if (result.IsSuccessStatusCode)
112         {
113             return RedirectToAction("Index");
114         }
115     }
116     return View(student);
117 }
118
119 public ActionResult Delete(int id)
120 {
121     using (var client = new HttpClient())
122     {
123         client.BaseAddress = new Uri("http://localhost:60260/api/");
124
125         //HTTP DELETE
126         var deleteTask = client.DeleteAsync("student/" + id.ToString
127             ());
128         deleteTask.Wait();
129
130         var result = deleteTask.Result;
131         if (result.IsSuccessStatusCode)
132         {
133             return RedirectToAction("Index");
134         }
135     }
136
137     return RedirectToAction("Index");
```

```
138         }  
139     }  
140 }
```