

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Net;
5 using System.Net.Http;
6 using System.Web.Http;
7 using Assignment7.Models;
8
9 namespace Assignment7.Controllers.API
10 {
11     //[RoutePrefix("api/student")]
12     public class StudentController : ApiController
13     {
14         public StudentController()
15         {
16         }
17
18         [HttpGet]
19         public IHttpActionResult GetAllStudents()
20         {
21             IList<StudentViewModel> students = null;
22             using (var ctx = new SchoolDBEntities())
23             {
24                 students = ctx.Students.Include("StudentAddress")
25                     .Select(s => new StudentViewModel()
26                     {
27                         Id = s.StudentID,
28                         Name = s.StudentName
29                     }).ToList<StudentViewModel>();
30             }
31
32             if (students.Count == 0)
33             {
34                 return NotFound();
35             }
36
37             return Ok(students);
38         }
39
40         [HttpGet]
41         public IHttpActionResult GetAllStudentsWithAddress(bool includeAddress)
42         {
43             IList<StudentViewModel> students = null;
44
45             using (var ctx = new SchoolDBEntities())
46             {
47                 students = ctx.Students.Include("StudentAddress").Select(s =>
48                     new StudentViewModel()
```

```
49         Id = s.StudentID,
50         Name = s.StudentName,
51         Address = s.StudentAddress == null ? null : new AddressViewModel()
52     {
53         StudentId = s.StudentAddress.StudentID,
54         Address1 = s.StudentAddress.Address1,
55         Address2 = s.StudentAddress.Address2,
56         City = s.StudentAddress.City,
57         State = s.StudentAddress.State
58     }
59     }).ToList<StudentViewModel>();
60 }
61
62
63     if (students.Count == 0)
64     {
65         return NotFound();
66     }
67
68     return Ok(students);
69 }
70
71 [HttpGet]
72 public IHttpActionResult GetStudentById(int id)
73 {
74     StudentViewModel student = null;
75
76     using (var ctx = new SchoolDBEntities())
77     {
78         student = ctx.Students.Include("StudentAddress")
79             .Where(s => s.StudentID == id)
80             .Select(s => new StudentViewModel()
81             {
82                 Id = s.StudentID,
83                 Name = s.StudentName
84             }).FirstOrDefault<StudentViewModel>();
85     }
86
87     if (student == null)
88     {
89         return NotFound();
90     }
91
92     return Ok(student);
93 }
94
95 [HttpGet]
96 public IHttpActionResult GetAllStudents(string name)
```

```
97     {
98         IList<StudentViewModel> students = null;
99
100        using (var ctx = new SchoolDBEntities())
101        {
102            students = ctx.Students.Include("StudentAddress")
103                .Where(s => s.StudentName.ToLower() == name.ToLower())
104                .Select(s => new StudentViewModel()
105                {
106                    Id = s.StudentID,
107                    Name = s.StudentName,
108                    Address = s.StudentAddress == null ? null : new AddressViewModel()
109                    {
110                        StudentId = s.StudentAddress.StudentID,
111                        Address1 = s.StudentAddress.Address1,
112                        Address2 = s.StudentAddress.Address2,
113                        City = s.StudentAddress.City,
114                        State = s.StudentAddress.State
115                    }
116                }).ToList<StudentViewModel>();
117        }
118
119
120        if (students.Count == 0)
121        {
122            return NotFound();
123        }
124        return Ok(students);
125    }
126
127    [HttpGet]
128    public IHttpActionResult GetAllStudentsInSameStandard(int standardId)
129    {
130        IList<StudentViewModel> students = null;
131
132        using (var ctx = new SchoolDBEntities())
133        {
134            students = ctx.Students.Include("StudentAddress").Include
135                ("Standard").Where(s => s.StandardId == standardId)
136                .Select(s => new StudentViewModel()
137                {
138                    Id = s.StudentID,
139                    Name = s.StudentName,
140                    Address = s.StudentAddress == null ? null : new AddressViewModel()
141                    {
142                        StudentId = s.StudentAddress.StudentID,
```

```
143         Address2 = s.StudentAddress.Address2,
144         City = s.StudentAddress.City,
145         State = s.StudentAddress.State
146     },
147     Standard = new StandardViewModel()
148     {
149         StandardId = s.Standard.StandardId,
150         Name = s.Standard.StandardName
151     }
152     }).ToList<StudentViewModel>());
153 }
154
155
156     if (students.Count == 0)
157     {
158         return NotFound();
159     }
160
161     return Ok(students);
162 }
163
164 [HttpPost]
165 public IHttpActionResult PostNewStudent(StudentViewModel student)
166 {
167     if (!ModelState.IsValid)
168         return BadRequest("Invalid data.");
169
170     using (var ctx = new SchoolDBEntities())
171     {
172         ctx.Students.Add(new Student()
173         {
174             StudentID = student.Id,
175             StudentName = student.Name
176         });
177
178         ctx.SaveChanges();
179     }
180
181     return Ok();
182 }
183
184 [HttpPut]
185 public IHttpActionResult Put(StudentViewModel student)
186 {
187     if (!ModelState.IsValid)
188         return BadRequest("Not a valid model");
189
190     using (var ctx = new SchoolDBEntities())
191     {
```

```
192         var existingStudent = ctx.Students.Where(s => s.StudentID == student.Id)
193         .FirstOrDefault<Student>();
194
195         if (existingStudent != null)
196         {
197             existingStudent.StudentName = student.Name;
198
199             ctx.SaveChanges();
200         }
201         else
202         {
203             return NotFound();
204         }
205     }
206
207     return Ok();
208 }
209
210 [HttpDelete]
211 public IActionResult Delete(int id)
212 {
213     if (id <= 0)
214         return BadRequest("Not a valid student id");
215
216     using (var ctx = new SchoolDBEntities())
217     {
218         var student = ctx.Students
219             .Where(s => s.StudentID == id)
220             .FirstOrDefault();
221
222         ctx.Entry(student).State =
223             System.Data.Entity.EntityState.Deleted;
224         ctx.SaveChanges();
225     }
226
227     return Ok();
228 }
229
230 }
231 }
232
```