

```
1 using exemplo2.api.Models;
2 using exemplo2.api.Services;
3 using Microsoft.AspNetCore.Http;
4 using Microsoft.AspNetCore.Mvc;
5 using System.Data;
6 using System.Data.SqlClient;
7
8 namespace exemplo2.api.Controllers
9 {
10     [Route("api/[controller]")]
11     [ApiController]
12     public class ClienteController : ControllerBase
13     {
14         private const string _stringConnection = "Server=LAPTOP-
            BLR4SJU9\
            \SQLEXPRESS01;Database=estudodb;Trusted_Connection=True;";
15
16         // GET: ClienteController
17         [HttpGet]
18         public IActionResult Get()
19         {
20             var clientes = new List<Cliente>();
21
22             var sqlConnection = new SqlConnection(_stringConnection);
23
24             var sqlCommand = new SqlCommand("select * from cliente",
                sqlConnection);
25
26             sqlConnection.Open();
27
28             var sqlDataReader = sqlCommand.ExecuteReader();
29
30             while (sqlDataReader.Read())
31             {
32                 var cliente = new Cliente
33                 (
34                     Convert.ToInt64(sqlDataReader["Id"].ToString()),
35                     sqlDataReader["Nome"].ToString(),
36                     sqlDataReader["Cpf"].ToString(),
37                     new Endereco
38                     {
39                         Logradouro = sqlDataReader
39                             ["Logradouro"].ToString(),
40                         Cidade = sqlDataReader["Cidade"].ToString()
41                     },
42                     new Email
43                     {
44                         Descricao = sqlDataReader["Email"].ToString()
45                     }
46                 );
47
48                 clientes.Add(cliente);
49             }
50         }
51     }
52 }
```

```
50
51         sqlConnection.Close();
52         sqlConnection.Dispose();
53
54         return Ok(clientes);
55
56     }
57
58     // GET: ClienteController/Details/5
59     [HttpGet("{id}")]
60     public IActionResult Get(int id)
61     {
62         var clientes = new List<Cliente>();
63
64
65         var sqlConnection = new SqlConnection(_stringConnection);
66
67         var sqlCommand = new SqlCommand("select * from cliente      ↗
68             where id = @id", sqlConnection);
69         sqlCommand.Parameters.Add("@id", SqlDbType.BigInt);
70         sqlCommand.Parameters["@id"].Value = id;
71
72         sqlConnection.Open();
73
74         var sqlDataReader = sqlCommand.ExecuteReader();
75
76         while (sqlDataReader.Read())
77         {
78             var cliente = new Cliente
79             (
80                 Convert.ToInt64(sqlDataReader["Id"].ToString()),
81                 sqlDataReader["Nome"].ToString(),
82                 sqlDataReader["Cpf"].ToString(),
83                 new Endereco
84                 {
85                     Logradouro = sqlDataReader      ↗
86                     ["Logradouro"].ToString(),
87                     Cidade = sqlDataReader["Cidade"].ToString()
88                 },
89                 new Email
90                 {
91                     Descricao = sqlDataReader["Email"].ToString()
92                 }
93             ); clientes.Add(cliente);
94         }
95
96         sqlConnection.Close();
97         sqlConnection.Dispose();
98
99         return Ok(clientes.FirstOrDefault());
100
```

```
101     }
102
103     // GET: ClienteController
104     [HttpGet("nome")]
105     public IActionResult GetNome(string nome)
106     {
107         var clienteService = new ClienteService();
108
109         var clientes = clienteService.ObterClientes();
110
111         return Ok(clientes.FirstOrDefault(x => x.Nome == nome));
112     }
113
114     //POST api/<ClienteController>
115     [HttpPost]
116     public IActionResult Post([FromBody] Cliente cliente)
117     {
118         var sqlConnection = new SqlConnection(_stringConnection);
119
120         var sqlCommand = new SqlCommand(@"INSERT INTO CLIENTE
121                                     (NOME, CPF, EMAIL, LOGRADOURO, NUMERO, BAIRRO, CIDADE, UF, CEP, TIPOPESSOA)
122                                     VALUES
123                                     (@NOME, @CPF, @EMAIL, @LOGRADOURO, @NUMERO, @BAIRRO, @CIDADE, @UF, @CEP, @TIPOPESSOA)", sqlConnection);
124
125         sqlCommand.Parameters.Add("@NOME", SqlDbType.VarChar);
126         sqlCommand.Parameters.Add("@CPF", SqlDbType.VarChar);
127         sqlCommand.Parameters.Add("@EMAIL", SqlDbType.VarChar);
128         sqlCommand.Parameters.Add("@LOGRADOURO", SqlDbType.VarChar);
129         sqlCommand.Parameters.Add("@NUMERO", SqlDbType.VarChar);
130         sqlCommand.Parameters.Add("@BAIRRO", SqlDbType.VarChar);
131         sqlCommand.Parameters.Add("@CIDADE", SqlDbType.VarChar);
132         sqlCommand.Parameters.Add("@UF", SqlDbType.VarChar);
133         sqlCommand.Parameters.Add("@CEP", SqlDbType.VarChar);
134         sqlCommand.Parameters.Add("@TIPOPESSOA", SqlDbType.Int);
135
136         sqlCommand.Parameters["@NOME"].Value = cliente.Nome;
137         sqlCommand.Parameters["@CPF"].Value = cliente.Cpf;
138         sqlCommand.Parameters["@EMAIL"].Value = cliente.Email.Descricao;
139         sqlCommand.Parameters["@LOGRADOURO"].Value = cliente.Endereco.Logradouro;
140         sqlCommand.Parameters["@NUMERO"].Value = cliente.Endereco.Numero;
141         sqlCommand.Parameters["@BAIRRO"].Value = cliente.Endereco.Bairro;
142         sqlCommand.Parameters["@CIDADE"].Value = cliente.Endereco.Cidade;
143         sqlCommand.Parameters["@UF"].Value = cliente.Endereco.Uf;
144         sqlCommand.Parameters["@CEP"].Value = cliente.Endereco.Cep;
145         sqlCommand.Parameters["@TIPOPESSOA"].Value =
```

```
        cliente.TipoPessoa;

144
145
146        sqlConnection.Open();
147
148        var linhaAfetadas = sqlCommand.ExecuteNonQuery();
149
150        sqlConnection.Close();
151        sqlConnection.Dispose();
152
153        if (linhaAfetadas > 0)
154            return Ok("Inserido com sucesso !! ");
155
156        return BadRequest("Erro !!");
157
158    }
159
160    //PUT api/<ClienteController>/5
161    [HttpPut]
162    public IActionResult Put([FromBody] Cliente cliente)
163    {
164        var sqlConnection = new SqlConnection(_stringConnection);
165
166        var sqlCommand = new SqlCommand(@"UPDATE CLIENTE SET NOME = 
            @NOME, CPF = @CPF, EMAIL = @EMAIL, LOGRADOURO = 
            @LOGRADOURO,
167        NUMERO = @NUMERO, BAIRRO = @BAIRRO, CIDADE = @CIDADE,UF = 
            @UF, CEP = @CEP, TIPOPESSOA = @TIPOPESSOA WHERE ID = @ID", 
            sqlConnection);

168        sqlCommand.Parameters.Add("@ID", SqlDbType.BigInt);
169        sqlCommand.Parameters.Add("@NOME", SqlDbType.VarChar);
170        sqlCommand.Parameters.Add("@CPF", SqlDbType.VarChar);
171        sqlCommand.Parameters.Add("@EMAIL", SqlDbType.VarChar);
172        sqlCommand.Parameters.Add("@LOGRADOURO", 
173            SqlDbType.VarChar);
174        sqlCommand.Parameters.Add("@NUMERO", SqlDbType.VarChar);
175        sqlCommand.Parameters.Add("@BAIRRO", SqlDbType.VarChar);
176        sqlCommand.Parameters.Add("@CIDADE", SqlDbType.VarChar);
177        sqlCommand.Parameters.Add("@UF", SqlDbType.VarChar);
178        sqlCommand.Parameters.Add("@CEP", SqlDbType.VarChar);
179        sqlCommand.Parameters.Add("@TIPOPESSOA", SqlDbType.Int);
180
181        sqlCommand.Parameters["@ID"].Value = cliente.Id;
182        sqlCommand.Parameters["@NOME"].Value = cliente.Nome;
183        sqlCommand.Parameters["@CPF"].Value = cliente.Cpf;
184        sqlCommand.Parameters["@EMAIL"].Value = 
            cliente.Email.Descricao;
185        sqlCommand.Parameters["@LOGRADOURO"].Value = 
            cliente.Endereco.Logradouro;
186        sqlCommand.Parameters["@NUMERO"].Value = 
            cliente.Endereco.Numero;
187        sqlCommand.Parameters["@BAIRRO"].Value =
```

```
        cliente.Endereco.Bairro;
188        sqlCommand.Parameters["@CIDADE"].Value = cliente.Endereco.Cidade;
189        sqlCommand.Parameters["@UF"].Value = cliente.Endereco.Uf;
190        sqlCommand.Parameters["@CEP"].Value = cliente.Endereco.Cep;
191        sqlCommand.Parameters["@TIPOPESSOA"].Value = cliente.TipoPessoa;

192
193        sqlConnection.Open();
194
195        var linhasAfetadas = sqlCommand.ExecuteNonQuery();
196
197        sqlConnection.Close();
198        sqlConnection.Dispose();
199
200        if (linhasAfetadas > 0)
201            return Ok("Atualizado com sucesso !!");
202        return BadRequest("Erro: Não atualizado !!");
203
204    }
205
206    // DELETE api/<ClienteController>/5
207    [HttpDelete("{id}")]
208    public IActionResult Delete(int id)
209    {
210        var sqlConnection = new SqlConnection(_stringConnection);
211
212        var sqlCommand = new SqlCommand(@"DELETE FROM CLIENTE WHERE
        ID = @ID", sqlConnection);

213
214        sqlCommand.Parameters.Add("@ID", SqlDbType.BigInt);
215        sqlCommand.Parameters["@ID"].Value = id;
216
217        sqlConnection.Open();
218
219        var linhasAfetadas = sqlCommand.ExecuteNonQuery();
220
221        sqlConnection.Close();
222        sqlConnection.Dispose();
223
224        if (linhasAfetadas > 0)
225            return Ok("Deletada com sucesso !");
226
227        return BadRequest("Erro: Não deletado ! ");
228
229
230    }
231
232    }
233 }
234
```