

```
1 using Microsoft.AspNetCore.Authorization;
2 using Microsoft.AspNetCore.Mvc;
3 using Microsoft.EntityFrameworkCore;
4
5 using api_pim.Entities;
6 using System.Net;
7 using api_pim.Exceptions;
8
9 namespace api_pim.Controllers;
10
11 [ApiController]
12 [Route("api/[controller]")]
13 public class TipoCargoController : ControllerBase
14 {
15     private readonly ApplicationDbContext _context;
16
17     private readonly ILogger<TipoCargoController> _logger;
18
19     public TipoCargoController(ApplicationDbContext context,           ↗
20         ILogger<TipoCargoController> logger)
21     {
22         _context = context;
23         _logger = logger;
24     }
25
26     [HttpGet]
27     [Authorize]
28     public IActionResult Get()
29     {
30         try
31         {
32             var result = _context.TipoCargo.ToList();
33             _logger.LogInformation("TipoCargoController.Get -> [Success]");           ↗
34             return Ok(result);
35         }
36         catch (Exception)
37         {
38             _logger.LogError("TipoCargoController.Get -> [Error]");
39             throw new ApiException((int)           ↗
40                 HttpStatusCode.InternalServerError, $"Erro interno           ↗
41                 [{ErrorCode.GTC}]");
42         }
43     }
44
45     [HttpPost]
46     [Authorize]
47     public IActionResult Create([FromBody] TipoCargo request)
48     {
49         try
50         {
51             if (request == null)
52             {
53                 return BadRequest();
54             }
55
56             if (_context.TipoCargo.Any(c => c.Codigo == request.Codigo))
57             {
58                 return BadRequest();
59             }
60
61             _context.TipoCargo.Add(request);
62             _context.SaveChanges();
63
64             return CreatedAtRoute("api/[controller]", new { id = request.Codigo }, request);
65         }
66         catch (Exception)
67         {
68             _logger.LogError("TipoCargoController.Create -> [Error]");
69             throw new ApiException((int)
70                 HttpStatusCode.InternalServerError, $"Erro interno
71                 [{ErrorCode.GTC}]");
72         }
73     }
74
75     [HttpPut]
76     [Authorize]
77     public IActionResult Update(TipoCargo request)
78     {
79         try
80         {
81             if (request == null)
82             {
83                 return BadRequest();
84             }
85
86             if (!_context.TipoCargo.Any(c => c.Codigo == request.Codigo))
87             {
88                 return BadRequest();
89             }
90
91             _context.TipoCargo.Update(request);
92             _context.SaveChanges();
93
94             return NoContent();
95         }
96         catch (Exception)
97         {
98             _logger.LogError("TipoCargoController.Update -> [Error]");
99             throw new ApiException((int)
100                 HttpStatusCode.InternalServerError, $"Erro interno
101                 [{ErrorCode.GTC}]");
102         }
103     }
104
105     [HttpDelete]
106     [Authorize]
107     public IActionResult Delete(int id)
108     {
109         try
110         {
111             if (id == 0)
112             {
113                 return BadRequest();
114             }
115
116             if (!_context.TipoCargo.Any(c => c.Codigo == id))
117             {
118                 return BadRequest();
119             }
120
121             _context.TipoCargo.Remove(_context.TipoCargo.FirstOrDefault(c => c.Codigo == id));
122             _context.SaveChanges();
123
124             return NoContent();
125         }
126         catch (Exception)
127         {
128             _logger.LogError("TipoCargoController.Delete -> [Error]");
129             throw new ApiException((int)
130                 HttpStatusCode.InternalServerError, $"Erro interno
131                 [{ErrorCode.GTC}]");
132         }
133     }
134 }
```

```
50         return BadRequest(new { message = "Dados do tipo de cargo inválidos." });
51     }
52
53     _context.TipoCargo.Add(request);
54     _context.SaveChanges();
55     _logger.LogInformation("TipoCargoController.Get -> [Success]");
56
57     return Ok($"Tipo de cargo [{request.Valor}] criado com sucesso");
58 }
59 catch (Exception)
60 {
61     _logger.LogError("TipoCargoController.Create -> [Error]");
62     throw new ApiException((int)
        HttpStatusCode.InternalServerError, $"Erro interno [{ErrorCode.CTC}]");
63 }
64 }
65
66 [HttpDelete]
67 [Authorize]
68 public IActionResult Delete(int cod) {
69     try
70     {
71         var tipoCargo = _context.TipoCargo.FirstOrDefault(x => x.Cod == cod);
72         if (tipoCargo == null) {
73             return NotFound(new { message = "Tipo de cargo não encontrado" });
74         }
75
76         _context.TipoCargo.Remove(tipoCargo);
77         _context.SaveChanges();
78         return NoContent();
79     }
80     catch (Exception)
81     {
82         _logger.LogError("TipoCargoController.Create -> [Error]");
83         throw new ApiException((int)
            HttpStatusCode.InternalServerError, $"Erro interno [{ErrorCode.DTC}]");
84     }
85 }
86 }
```