**JSON web token**

**Task:**

There are various modes of authenticating a request. Json Web Token(JWT) authentication is one among them. It is a methodology of passing a token in the Authorization header value in the request so that it can be checked at the WebAPI and validated. If not there, then ‘**Unauthorized**’ status message with status code 401 should be thrown.

**Solution:**

**Program.cs:**

using Microsoft.AspNetCore.Authentication.JwtBearer;

using Microsoft.IdentityModel.Tokens;

using System.Text;

var builder = WebApplication.CreateBuilder(args);

builder.Services.AddControllers();

string securityKey = "mysuperdupersecretmysuperdupersecret!";

var symmetricSecurityKey = new SymmetricSecurityKey(Encoding.UTF8.GetBytes(securityKey));

builder.Services.AddAuthentication(x =>

{

x.DefaultAuthenticateScheme = JwtBearerDefaults.AuthenticationScheme;

x.DefaultChallengeScheme = JwtBearerDefaults.AuthenticationScheme;

x.DefaultSignInScheme = JwtBearerDefaults.AuthenticationScheme;

})

.AddJwtBearer(JwtBearerDefaults.AuthenticationScheme, x =>

{

x.TokenValidationParameters = new TokenValidationParameters

{

ValidateIssuer = true,

ValidateAudience = true,

ValidateLifetime = true,

ValidateIssuerSigningKey = true,

ValidIssuer = "mySystem",

ValidAudience = "myUsers",

IssuerSigningKey = symmetricSecurityKey

};

});

builder.Services.AddEndpointsApiExplorer();

builder.Services.AddSwaggerGen();

var app = builder.Build();

if (app.Environment.IsDevelopment())

{

app.UseSwagger();

app.UseSwaggerUI();

}

app.UseHttpsRedirection();

app.UseAuthentication(); *// Enable JWT Authentication*

app.UseAuthorization();

app.MapControllers();

app.Run();

**AuthController.cs:**

using Microsoft.AspNetCore.Mvc;

using Microsoft.AspNetCore.Authorization;

using Microsoft.IdentityModel.Tokens;

using System.IdentityModel.Tokens.Jwt;

using System.Security.Claims;

using System.Text;

using System;

using System.Collections.Generic;

**namespace** YourNamespace.Controllers

{

[ApiController]

[Route("api/[controller]")]

[AllowAnonymous]

**public** **class** AuthController : ControllerBase

{

[HttpGet("login")]

**public** IActionResult Login()

{

**var** token = GenerateJSONWebToken(1001, "Admin");

return Ok(new { token });

}

**private** string GenerateJSONWebToken(int userId, string userRole)

{

**var** securityKey = new SymmetricSecurityKey(Encoding.UTF8.GetBytes("mysuperdupersecretmysuperdupersecret!"));

**var** credentials = new SigningCredentials(securityKey, SecurityAlgorithms.HmacSha256);

**var** claims = new List<Claim>

{

new Claim(ClaimTypes.Role, userRole),

new Claim("UserId", userId.ToString())

};

**var** token = new JwtSecurityToken(

issuer: "mySystem",

audience: "myUsers",

claims: claims,

expires: DateTime.Now.AddMinutes(2),

signingCredentials: credentials);

return new JwtSecurityTokenHandler().WriteToken(token);

}

}

}

**EmployeeController.cs:**

//Only POC get 200

using Microsoft.AspNetCore.Authorization;

using Microsoft.AspNetCore.Mvc;

using EmployeeApiProject.Models;

using System.Collections.Generic;

namespace EmployeeApiProject.Controllers

{

[Authorize(Roles = "POC")] // Only users with role POC allowed

[ApiController]

[Route("api/[controller]")]

public class EmployeesController : ControllerBase

{

private static List<Employee> employees = new List<Employee>

{

new Employee { Id = 1, Name = "Alice", Department = "HR", Salary = 50000 },

new Employee { Id = 2, Name = "Bob", Department = "IT", Salary = 60000 },

new Employee { Id = 3, Name = "Charlie", Department = "Finance", Salary = 55000 }

};

[HttpGet]

public ActionResult<List<Employee>> GetAllEmployees()

{

return Ok(employees);

}

}

}

//Both POC and Admin get 200

using Microsoft.AspNetCore.Authorization;

using Microsoft.AspNetCore.Mvc;

using EmployeeApiProject.Models;

using System.Collections.Generic;

namespace EmployeeApiProject.Controllers

{

[Authorize(Roles = "Admin,POC")] // Allow users with either Admin or POC roles

[ApiController]

[Route("api/[controller]")]

public class EmployeesController : ControllerBase

{

private static List<Employee> employees = new List<Employee>

{

new Employee { Id = 1, Name = "Alice", Department = "HR", Salary = 50000 },

new Employee { Id = 2, Name = "Bob", Department = "IT", Salary = 60000 },

new Employee { Id = 3, Name = "Charlie", Department = "Finance", Salary = 55000 }

};

[HttpGet]

public ActionResult<List<Employee>> GetAllEmployees()

{

return Ok(employees);

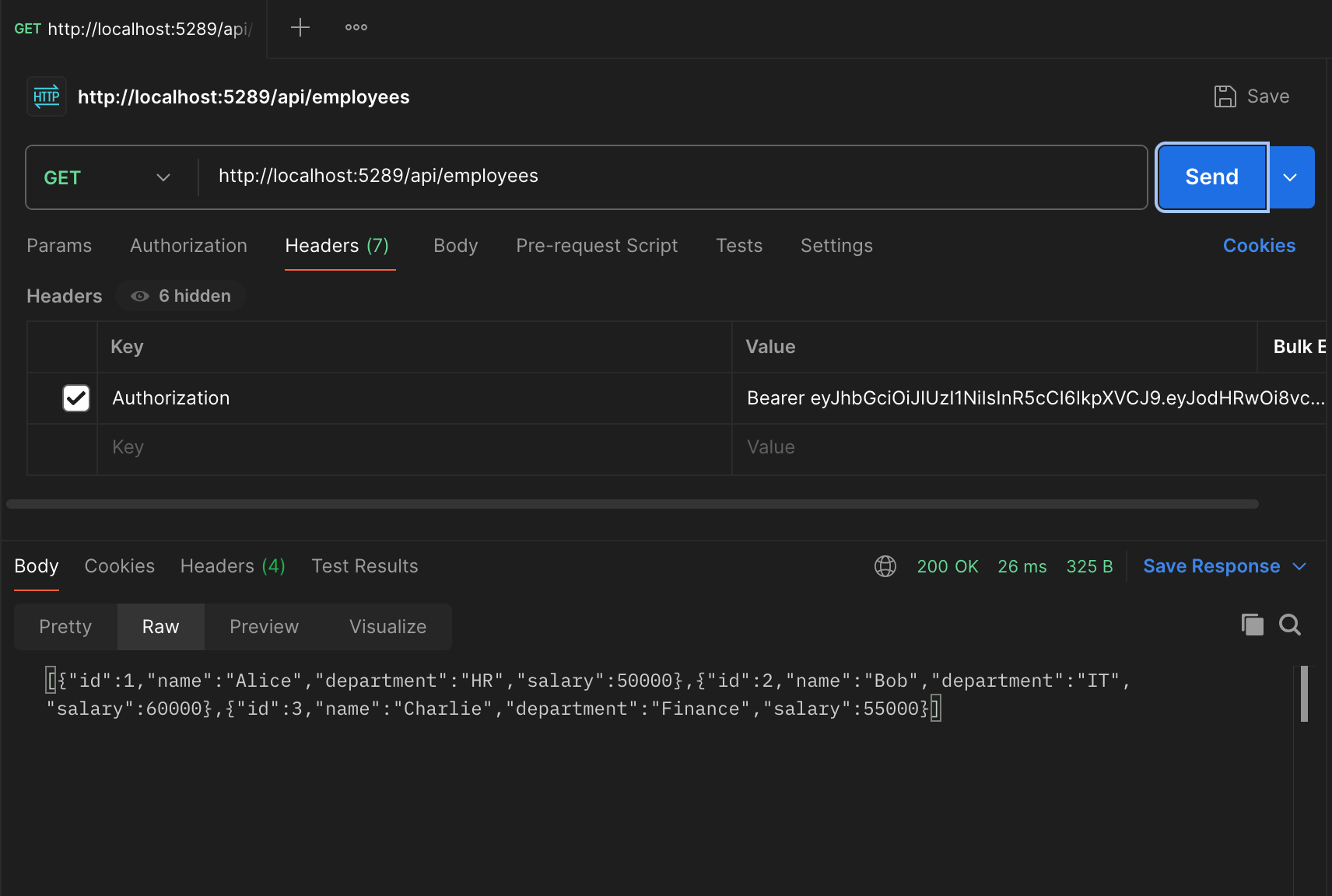
}

}

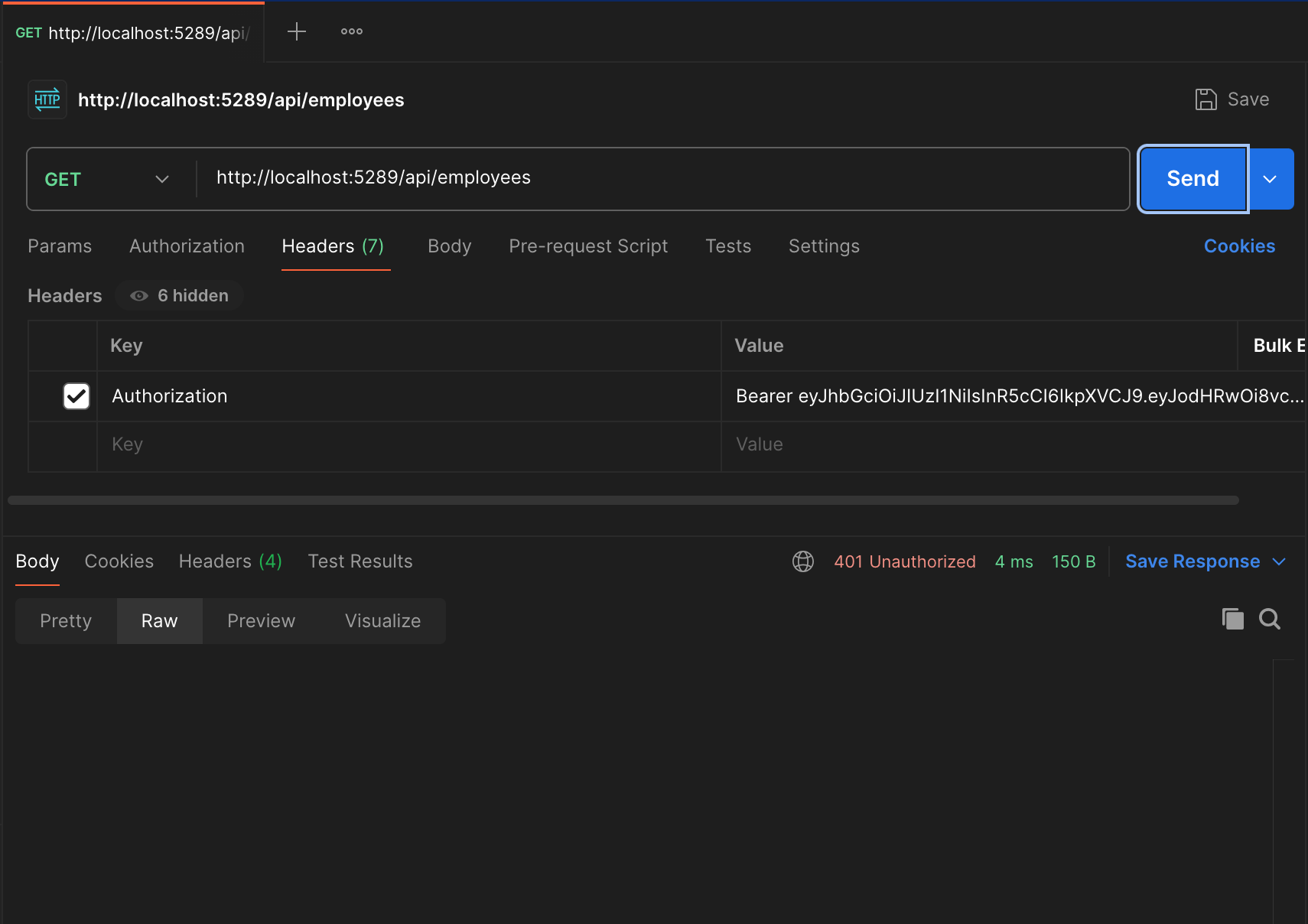
}

**Postman:**

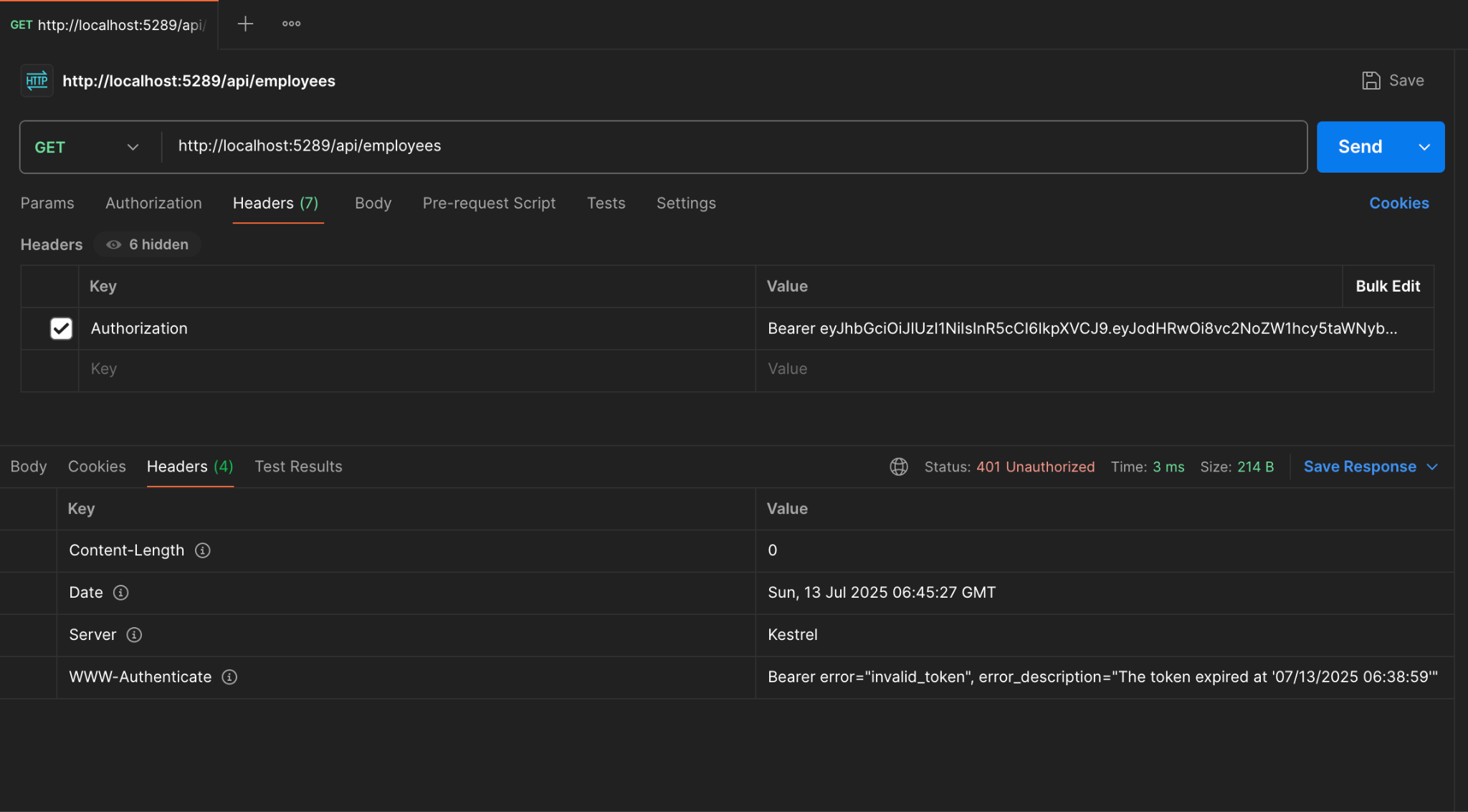
**Valid Token:**

****

**Invalid Token**

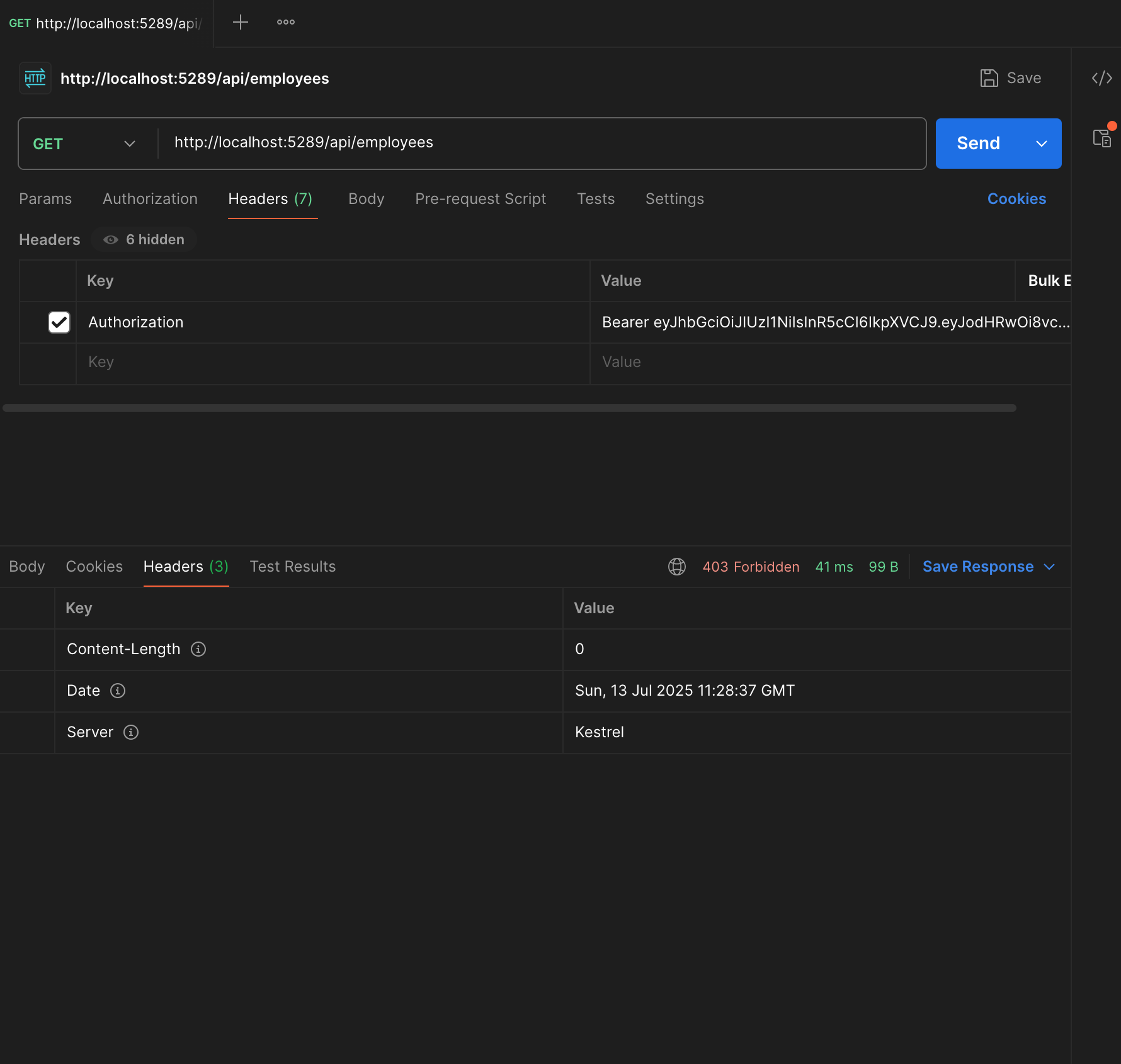
****

**Expired Token**

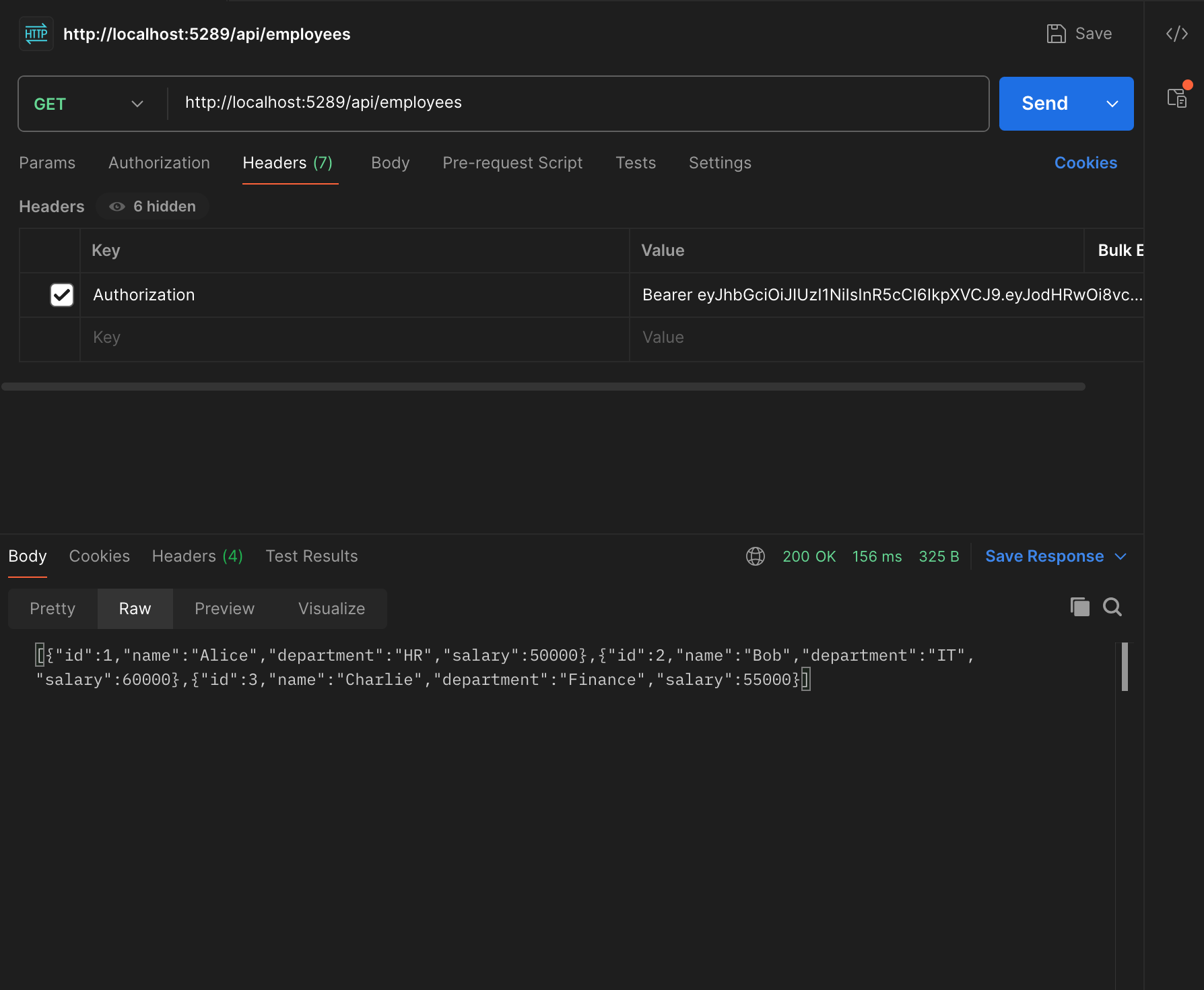
****

**While using Admin Token:**

**Only POC is authorized:**

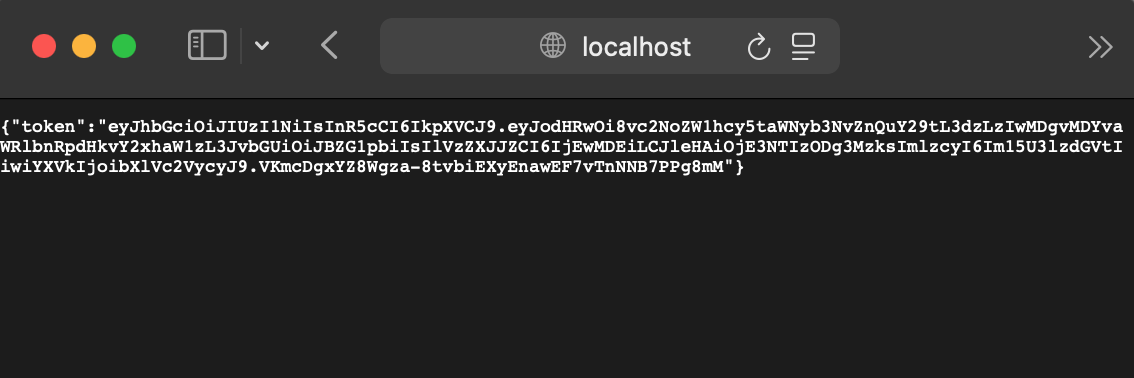
****

**POC and Admin are both authorized**

****

**Local Host:**

**Token**

****