// GET: api/APIProducts

[HttpGet]

public async Task<IActionResult> GetAllProducts(string searchString = "")

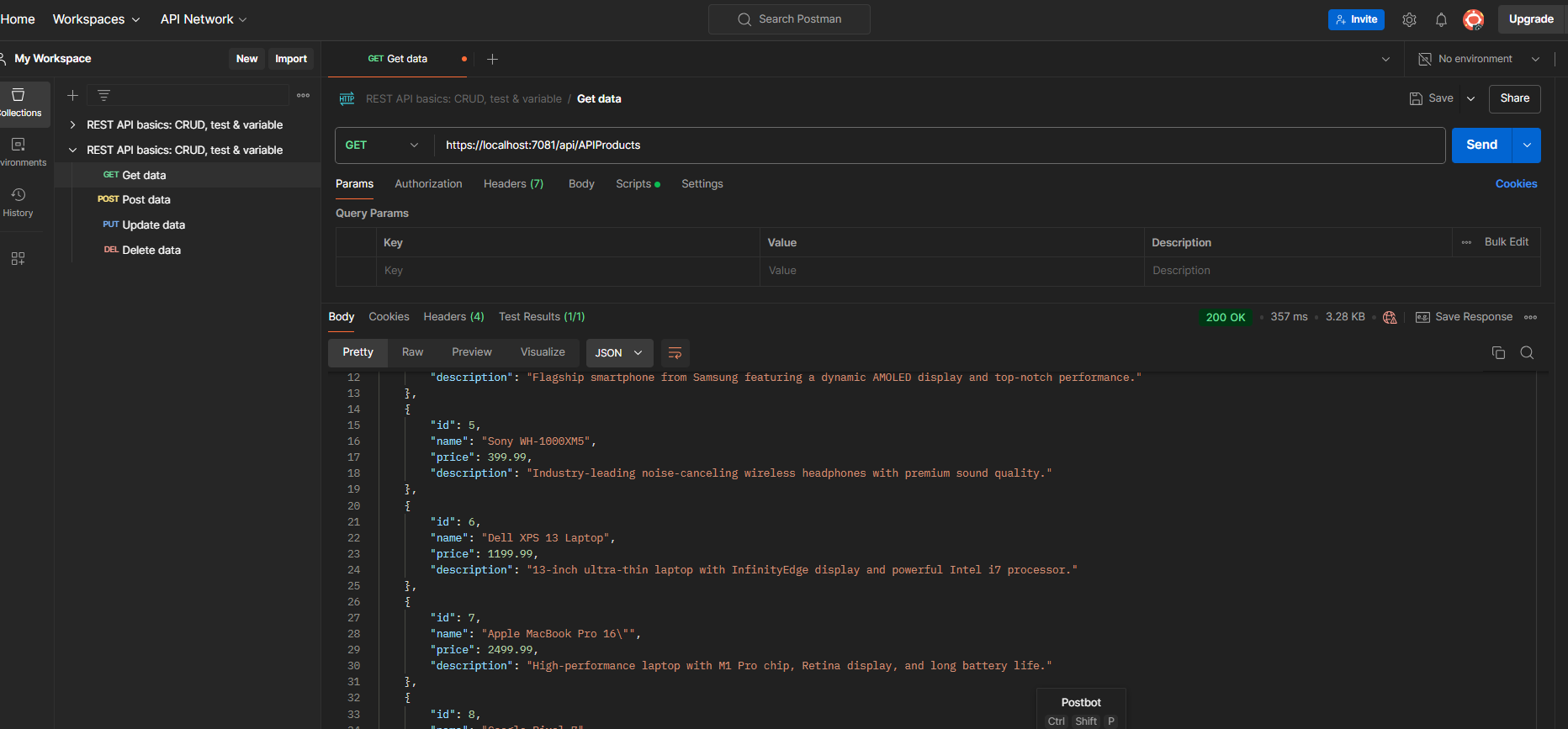
{

var products = await \_serviceProducts.ReadAsync(searchString);

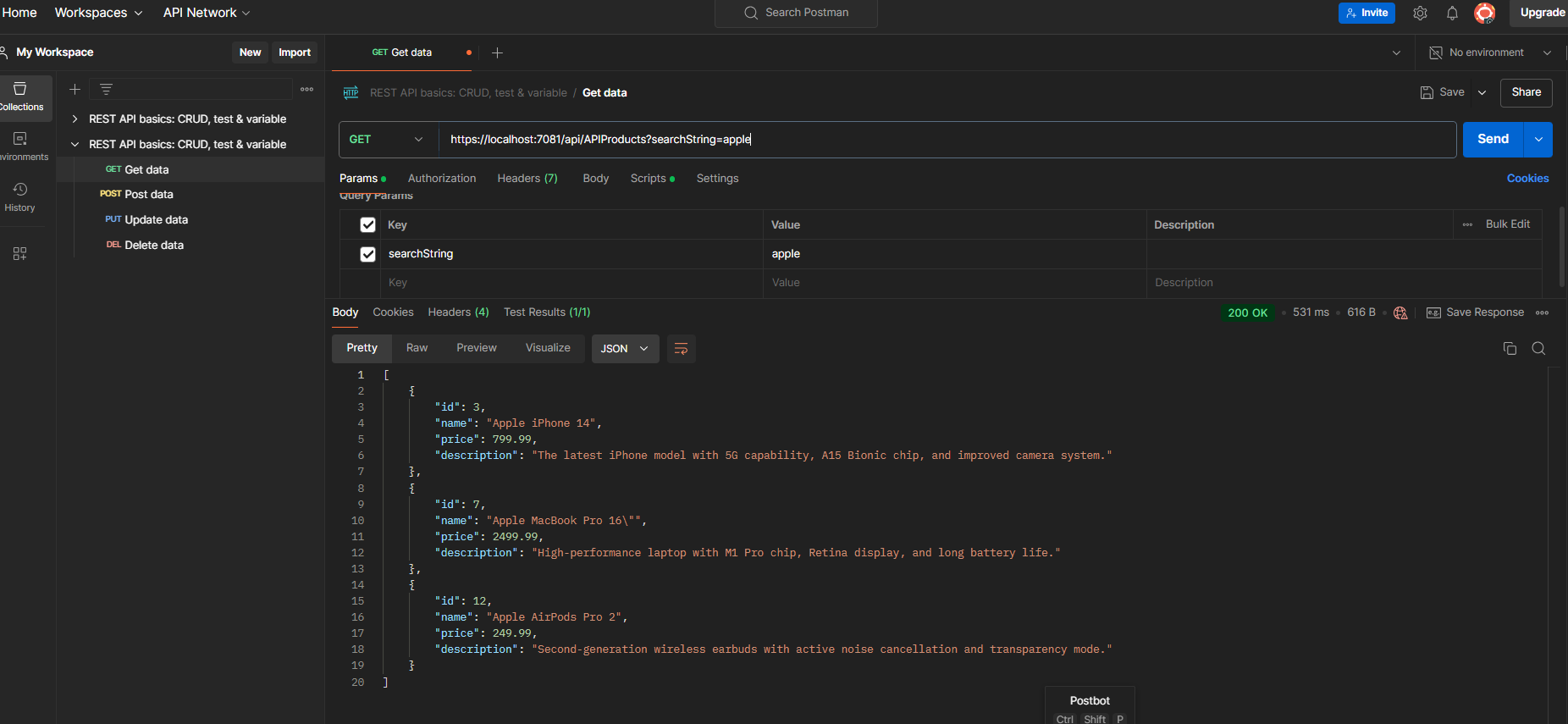
return Ok(products);

}

<https://localhost:7081/api/APIProducts>



<https://localhost:7081/api/APIProducts?searchString=apple>



// GET: api/APIProducts/5

[HttpGet("{id}")]

public async Task<IActionResult> GetProduct(int? id)

{

if (id == null)

{

return NotFound($"Product with id {id} is not found");

}

var product = await \_serviceProducts.GetByIdAsync(id.Value);

if (product == null)

{

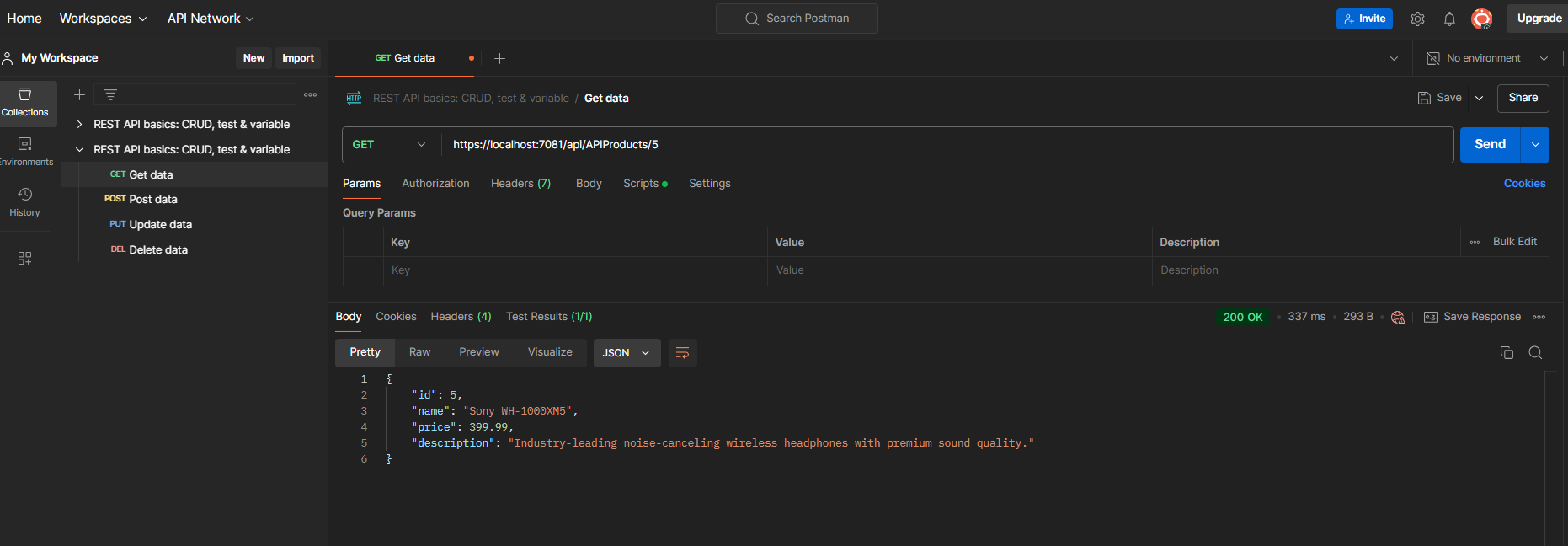
return NotFound();

}

return Ok(product);

}

<https://localhost:7081/api/APIProducts/5>



{

    "id": 5,

    "name": "Sony WH-1000XM5",

    "price": 399.99,

    "description": "Industry-leading noise-canceling wireless headphones with premium sound quality."

}

// POST: api/APIUser/Auth

[HttpPost("auth")]

public async Task<IActionResult> Auth([FromBody] AuthModel model)

{

string email = model.Email;

string password = model.Password;

if (string.IsNullOrEmpty(email) || string.IsNullOrEmpty(password))

{

return BadRequest("Email and password are required.");

}

var result = await \_signInManager.PasswordSignInAsync(

email,

password,

isPersistent: false,

lockoutOnFailure: false

);

if (result.Succeeded)

{

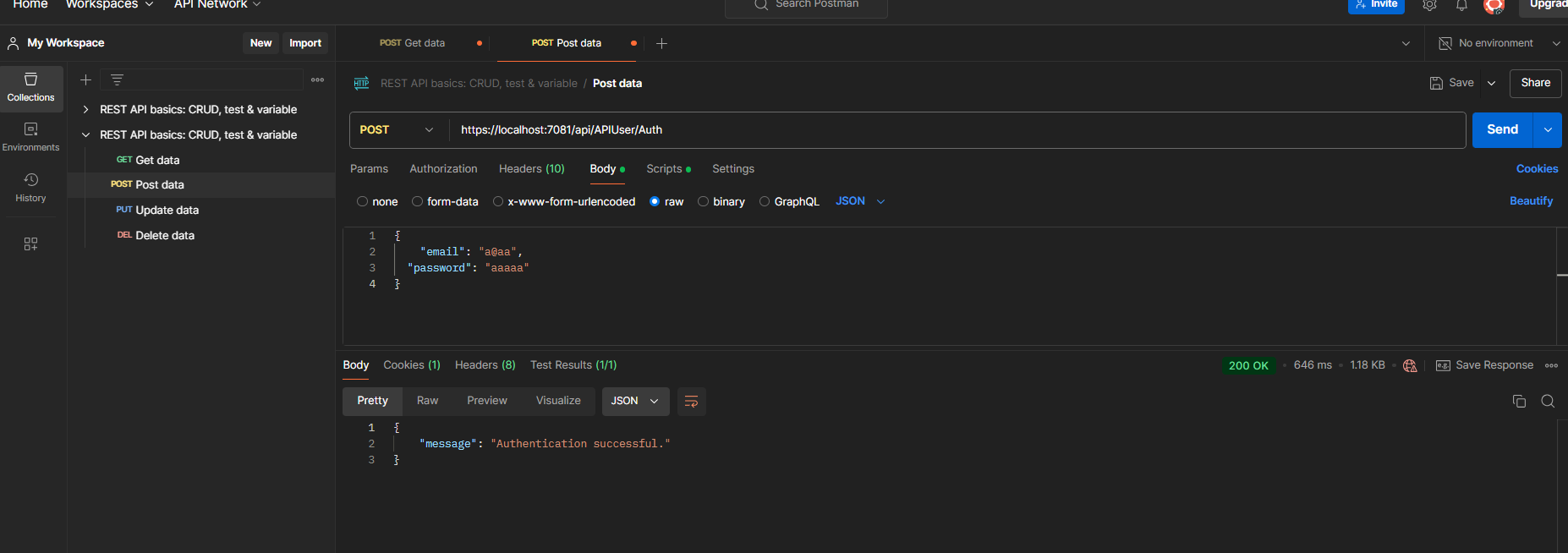
return Ok(new { message = "Authentication successful." });

}

return Unauthorized("Invalid login attempt.");

}

https://localhost:7081/api/APIUser/Auth



// PUT: api/APIProducts/5

[Authorize(Roles = "admin")]

[HttpPut("{id}")]

public async Task<IActionResult> UpdateProduct(int id, Product product)

{

if (id != product.Id)

{

return BadRequest();

}

var updatedProduct = await \_serviceProducts.UpdateAsync(id, product);

if (updatedProduct == null)

{

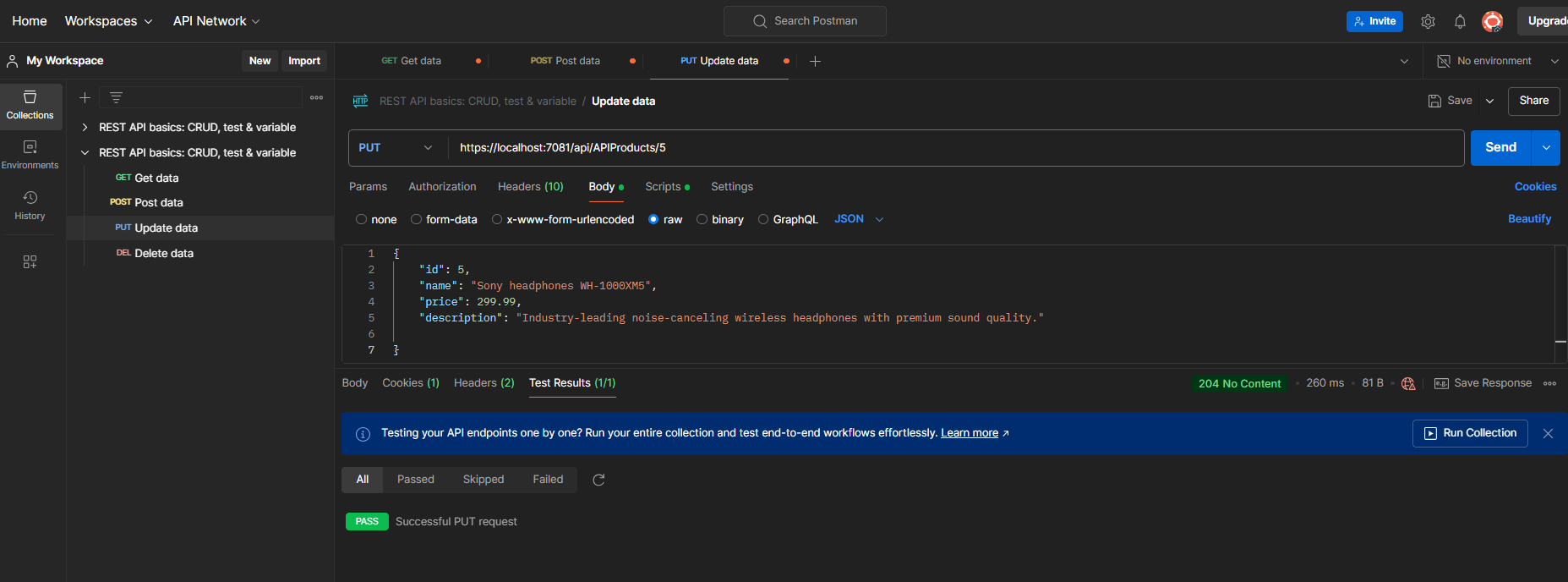
return NotFound();

}

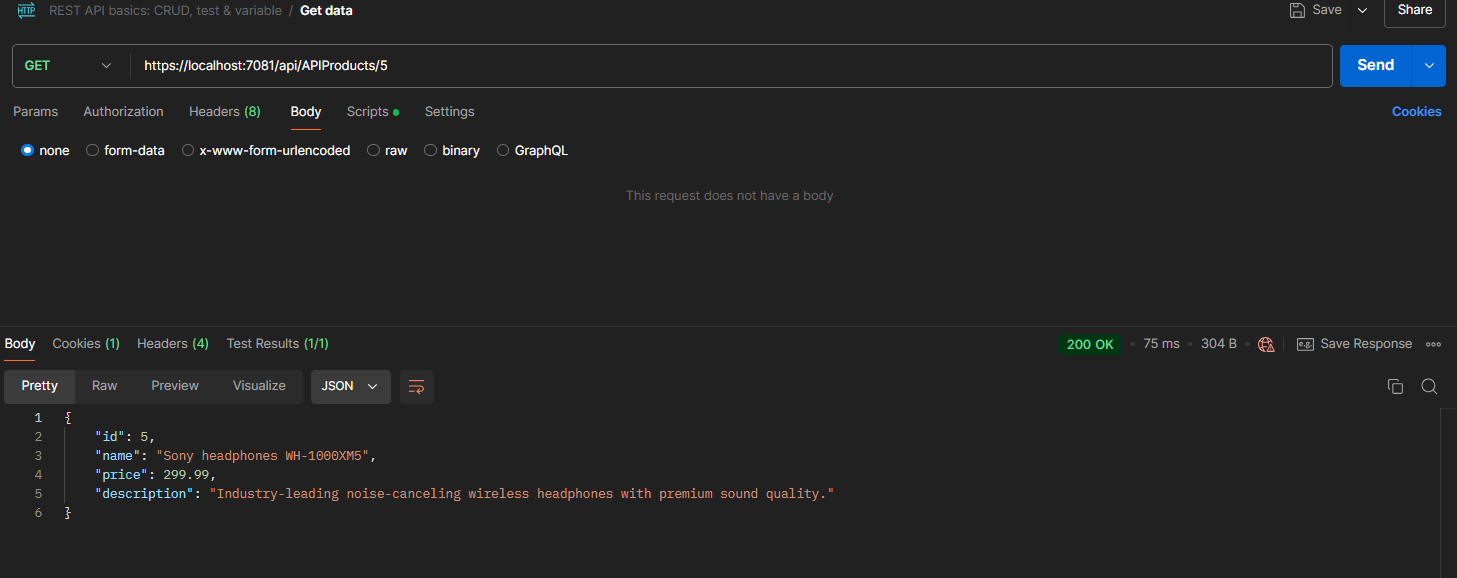
return NoContent();

}

https://localhost:7081/api/APIProducts/5



Перевірка, чи все спрацювало  
GET <https://localhost:7081/api/APIProducts/5>



// POST: api/APIProducts

[Authorize(Roles = "admin")]

[HttpPost]

public async Task<IActionResult> CreateProduct(Product product)

{

if (ModelState.IsValid)

{

return BadRequest(ModelState);

}

await \_serviceProducts.CreateAsync(product);

return CreatedAtAction(nameof(GetProduct), new { id = product.Id }, product);

}

<https://localhost:7081/api/APIProducts>

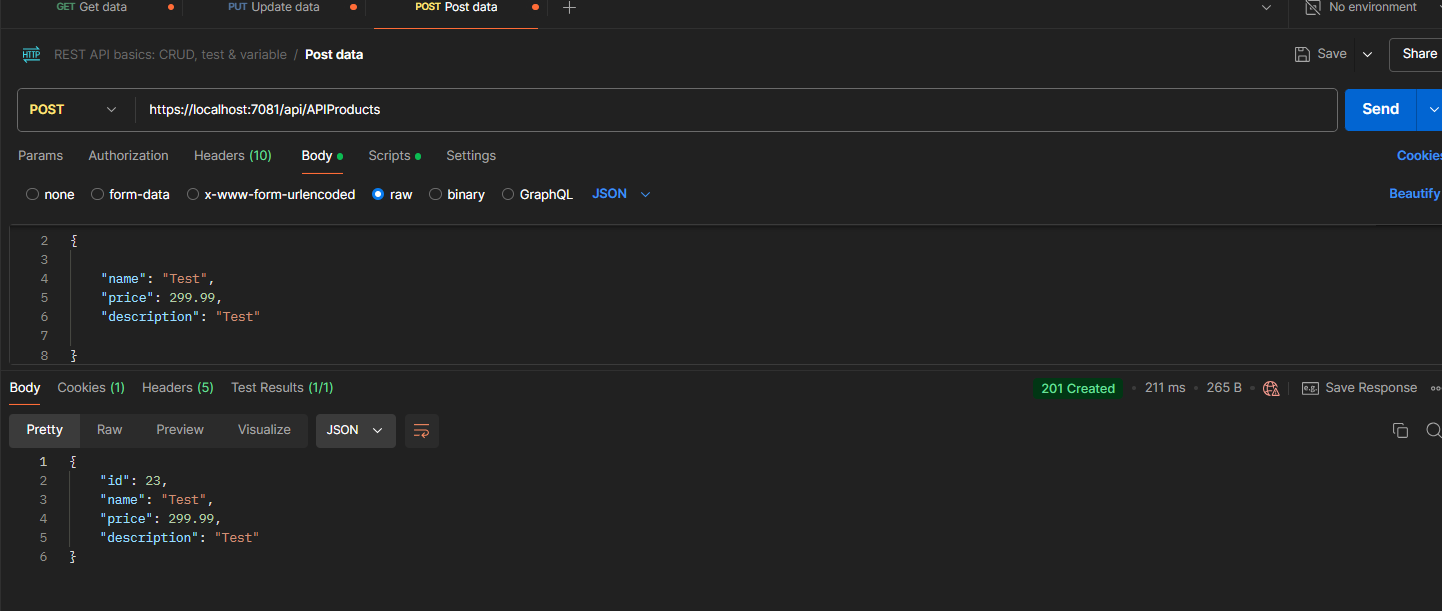
{

    "name": "Test",

    "price": 299.99,

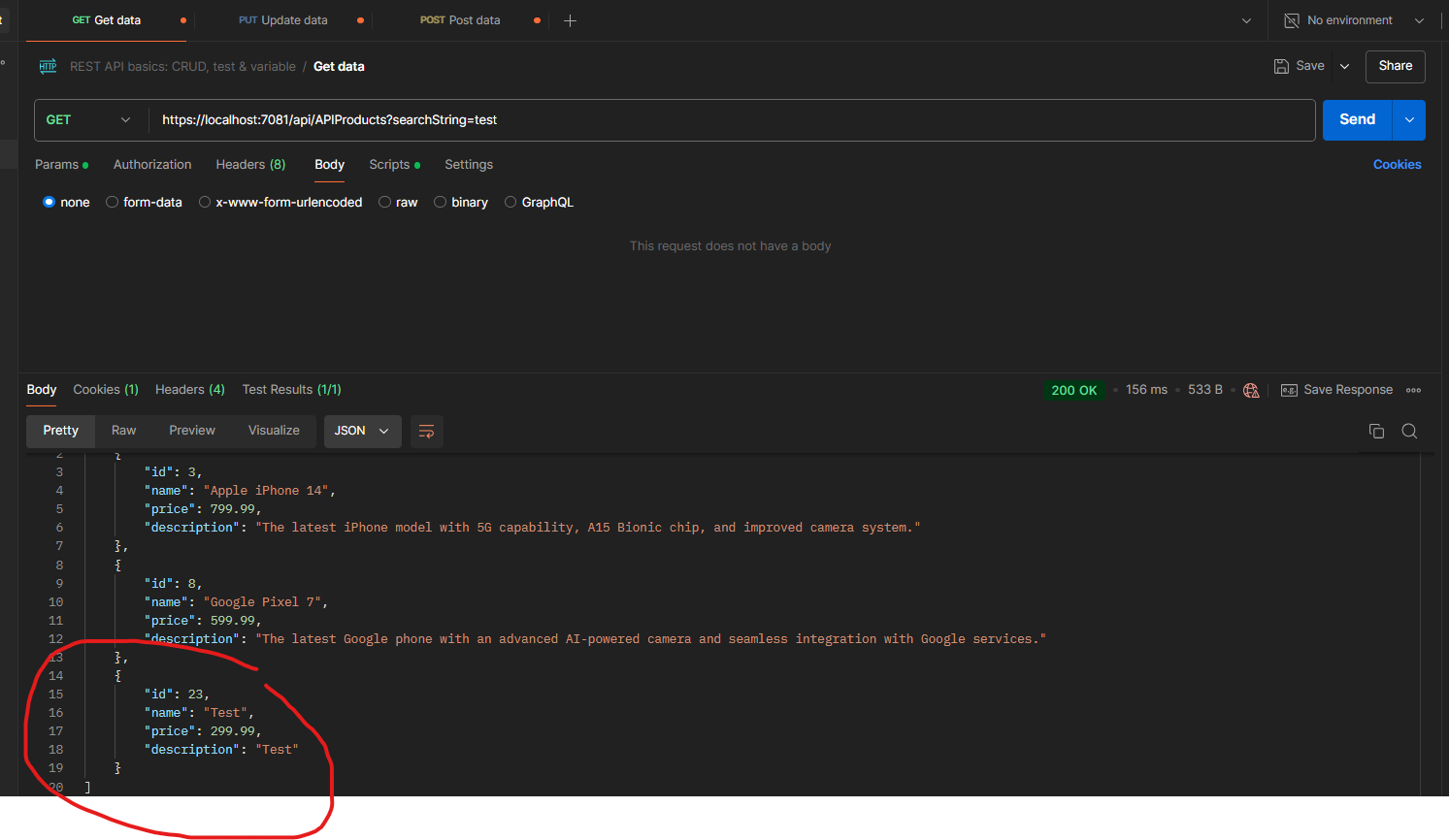
    "description": "Test"

}



Перевірка

https://localhost:7081/api/APIProducts?searchString=test



// DELETE: api/APIProducts/5

[Authorize(Roles = "admin")]

[HttpDelete("{id}")]

public async Task<IActionResult> DeleteProduct(int id)

{

var success = await \_serviceProducts.DeleteAsync(id);

if (!success)

{

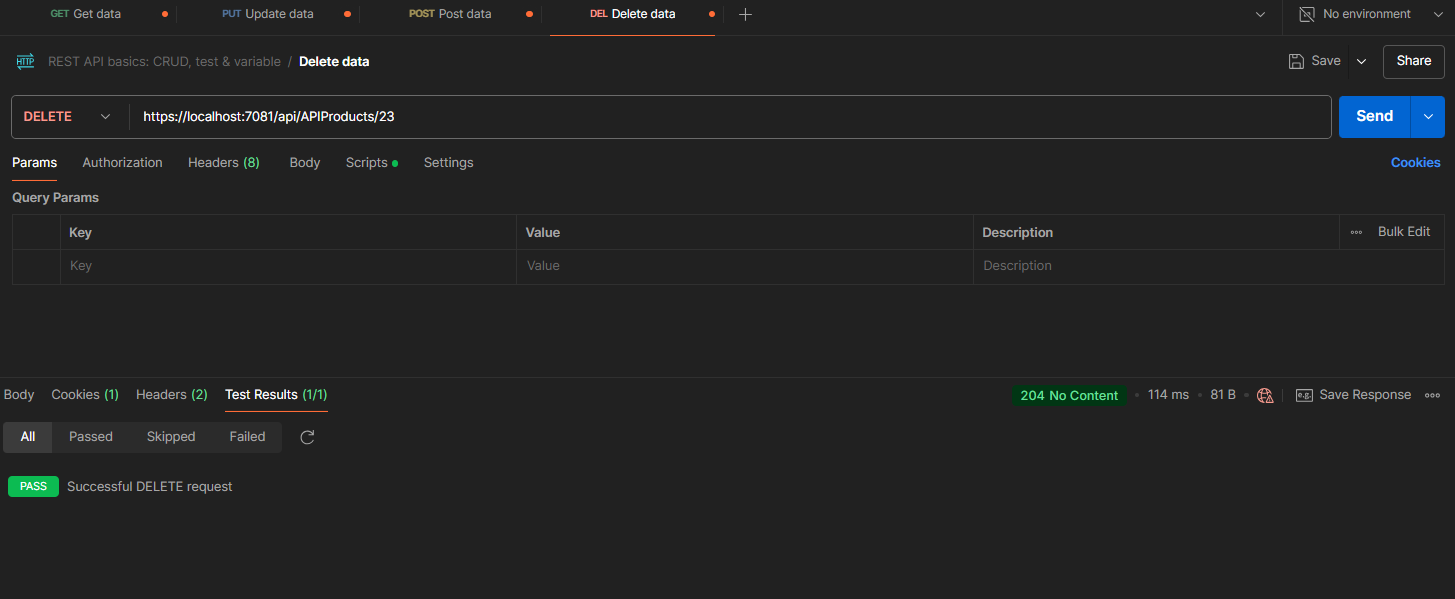
return NotFound();

}

return NoContent();

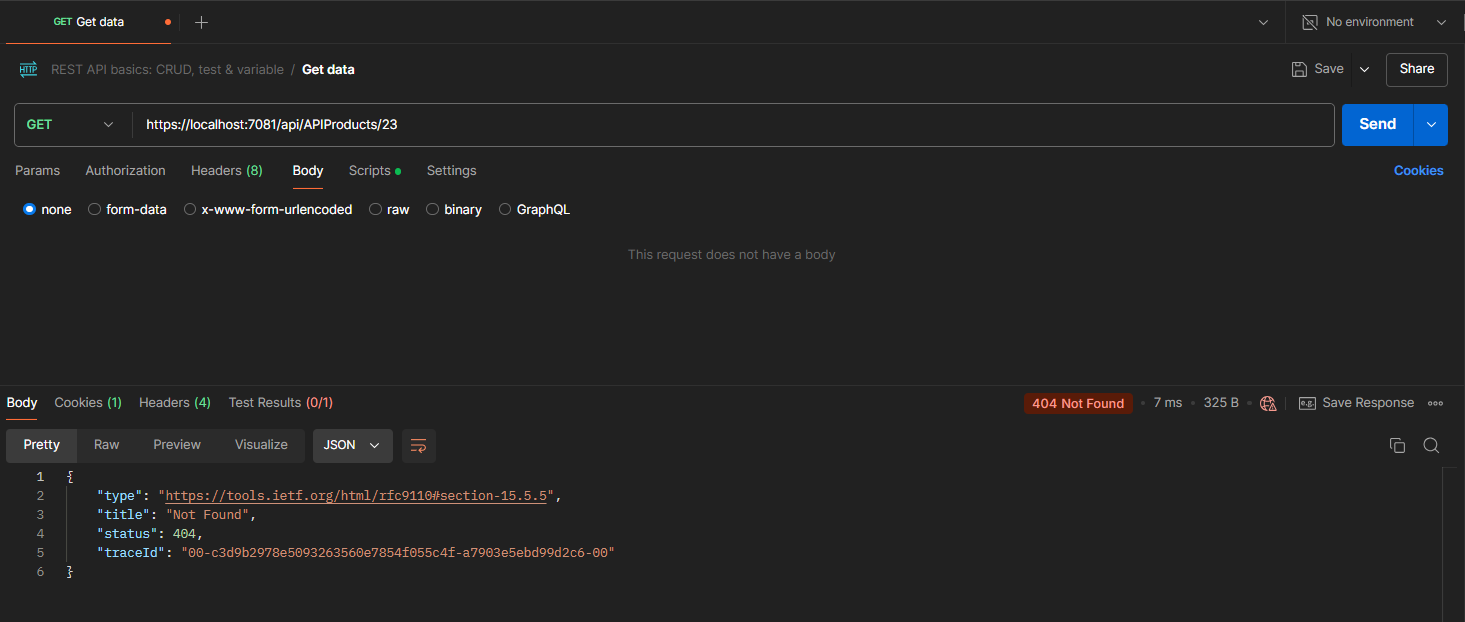
}

https://localhost:7081/api/APIProducts/23



Перевірка

<https://localhost:7081/api/APIProducts/23>



// POST: api/APIUser/Logout

[HttpPost("logout")]

public async Task<IActionResult> Logout()

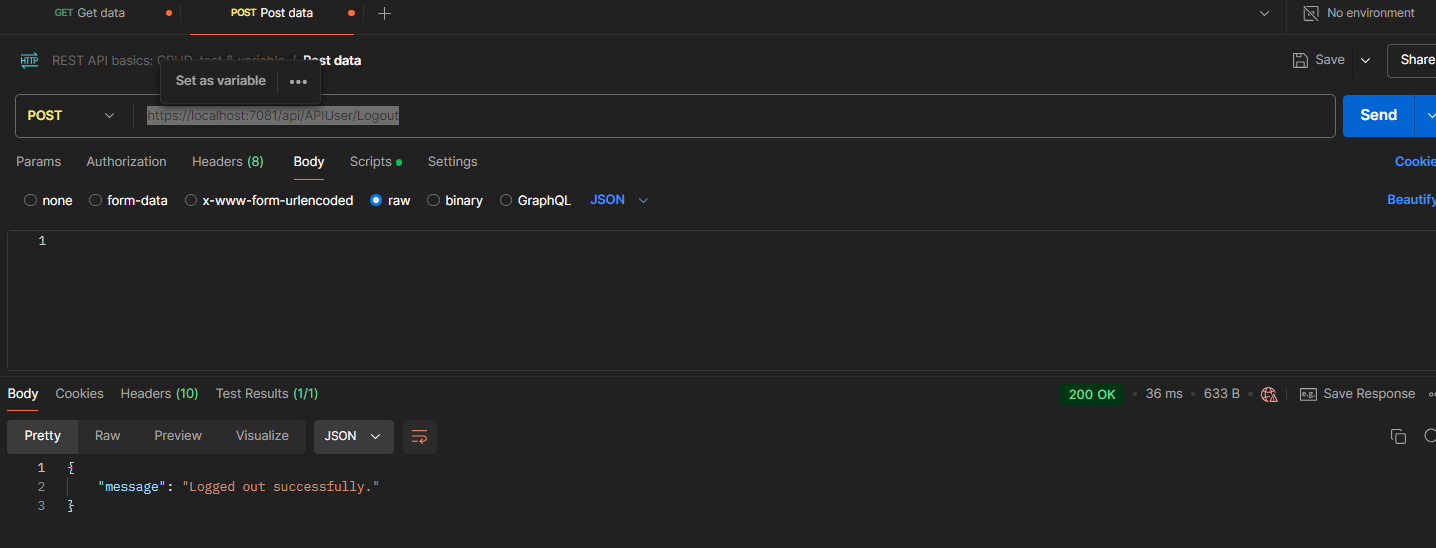
{

await \_signInManager.SignOutAsync();

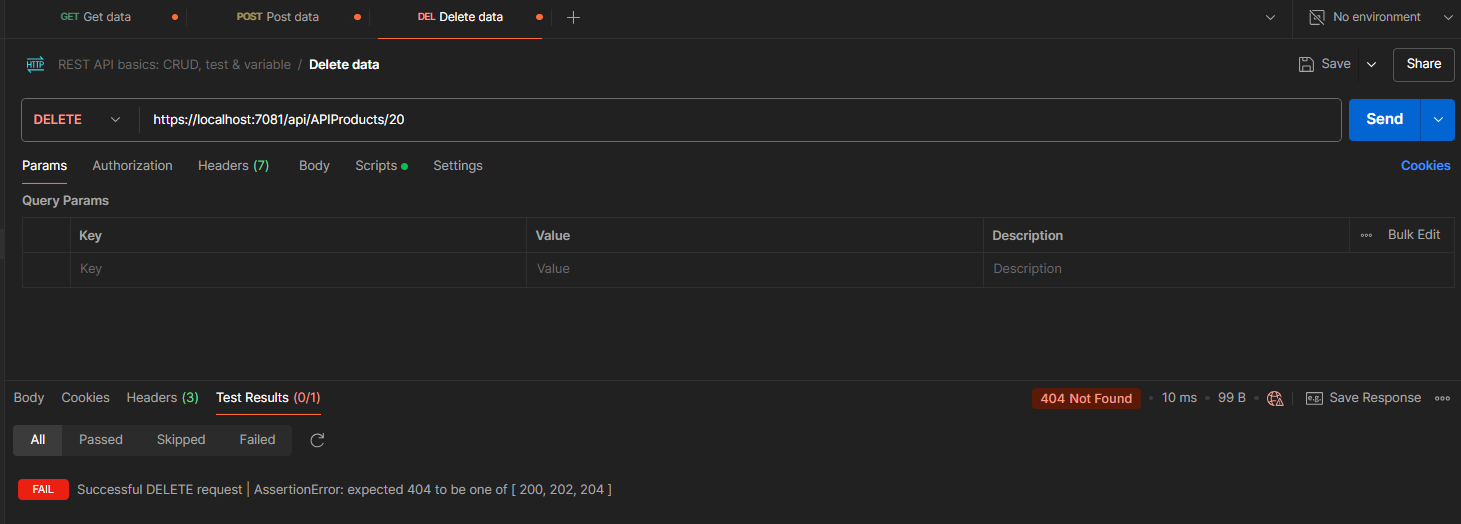
return Ok(new { message = "Logged out successfully." });

}

<https://localhost:7081/api/APIUser/Logout>



Перевірка – спроба удалити продукт без авторизації



Після авторизації цей же продукт видалити вдалося