**Ashutosh Dash  
Superset ID – 6365404  
WebAPI - 4**

using Microsoft.AspNetCore.Mvc;

using CustomWebApi.Models;

using CustomWebApi.Filters;

using System.Collections.Generic;

namespace CustomWebApi.Controllers

{

[ApiController]

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

[ServiceFilter(typeof(CustomAuthFilter))]

public class EmployeeController : ControllerBase

{

private readonly List<Employee> \_employees;

public EmployeeController()

{

\_employees = GetStandardEmployeeList();

}

[HttpGet]

[ProducesResponseType(StatusCodes.Status200OK)]

[ProducesResponseType(StatusCodes.Status500InternalServerError)]

public ActionResult<List<Employee>> GetStandard()

{

//throw new Exception("Test Exception"); // Uncomment to test custom exception filter

return Ok(\_employees);

}

[HttpPost]

[ProducesResponseType(StatusCodes.Status201Created)]

[ProducesResponseType(StatusCodes.Status400BadRequest)]

public ActionResult<Employee> CreateEmployee([FromBody] Employee newEmp)

{

if (newEmp == null || newEmp.Id <= 0)

{

return BadRequest("Invalid employee data");

}

var employees = GetStandardEmployeeList();

employees.Add(newEmp);

return CreatedAtAction(nameof(GetStandard), new { id = newEmp.Id }, newEmp);

}

[HttpPut("{id}")]

[ProducesResponseType(StatusCodes.Status200OK)]

[ProducesResponseType(StatusCodes.Status400BadRequest)]

public ActionResult<Employee> UpdateEmployee(int id, [FromBody] Employee updatedEmployee)

{

if (id <= 0)

{

return BadRequest("Invalid employee id");

}

var existingEmployee = \_employees.FirstOrDefault(e => e.Id == id);

if (existingEmployee == null)

{

return BadRequest("Invalid employee id");

}

existingEmployee.Name = updatedEmployee.Name;

existingEmployee.Salary = updatedEmployee.Salary;

existingEmployee.Permanent = updatedEmployee.Permanent;

existingEmployee.Department = updatedEmployee.Department;

existingEmployee.Skills = updatedEmployee.Skills;

existingEmployee.DateOfBirth = updatedEmployee.DateOfBirth;

return Ok(existingEmployee);

}

[HttpDelete("{id}")]

[ProducesResponseType(StatusCodes.Status200OK)]

[ProducesResponseType(StatusCodes.Status400BadRequest)]

public IActionResult DeleteEmployee(int id)

{

if (id <= 0)

{

return BadRequest("Invalid employee id");

}

var employees = GetStandardEmployeeList();

var emp = employees.FirstOrDefault(e => e.Id == id);

if (emp == null)

{

return BadRequest("Employee not found");

}

employees.Remove(emp);

return Ok($"Employee with ID {id} deleted");

}

private List<Employee> GetStandardEmployeeList()

{

return new List<Employee>

{

new Employee

{

Id = 1,

Name = "Ashutosh Dash",

Salary = 50000,

Permanent = true,

DateOfBirth = new DateTime(1999, 8, 15),

Department = new Department { Id = 101, Name = "IT" },

Skills = new List<Skill>

{

new Skill { Id = 1, Name = "C#" },

new Skill { Id = 2, Name = ".NET Core" }

}

},

new Employee

{

Id = 2,

Name = "Simran Sharma",

Salary = 60000,

Permanent = false,

DateOfBirth = new DateTime(1998, 11, 20),

Department = new Department { Id = 102, Name = "HR" },

Skills = new List<Skill>

{

new Skill { Id = 3, Name = "Excel" },

new Skill { Id = 4, Name = "Recruitment" }

}

}

};

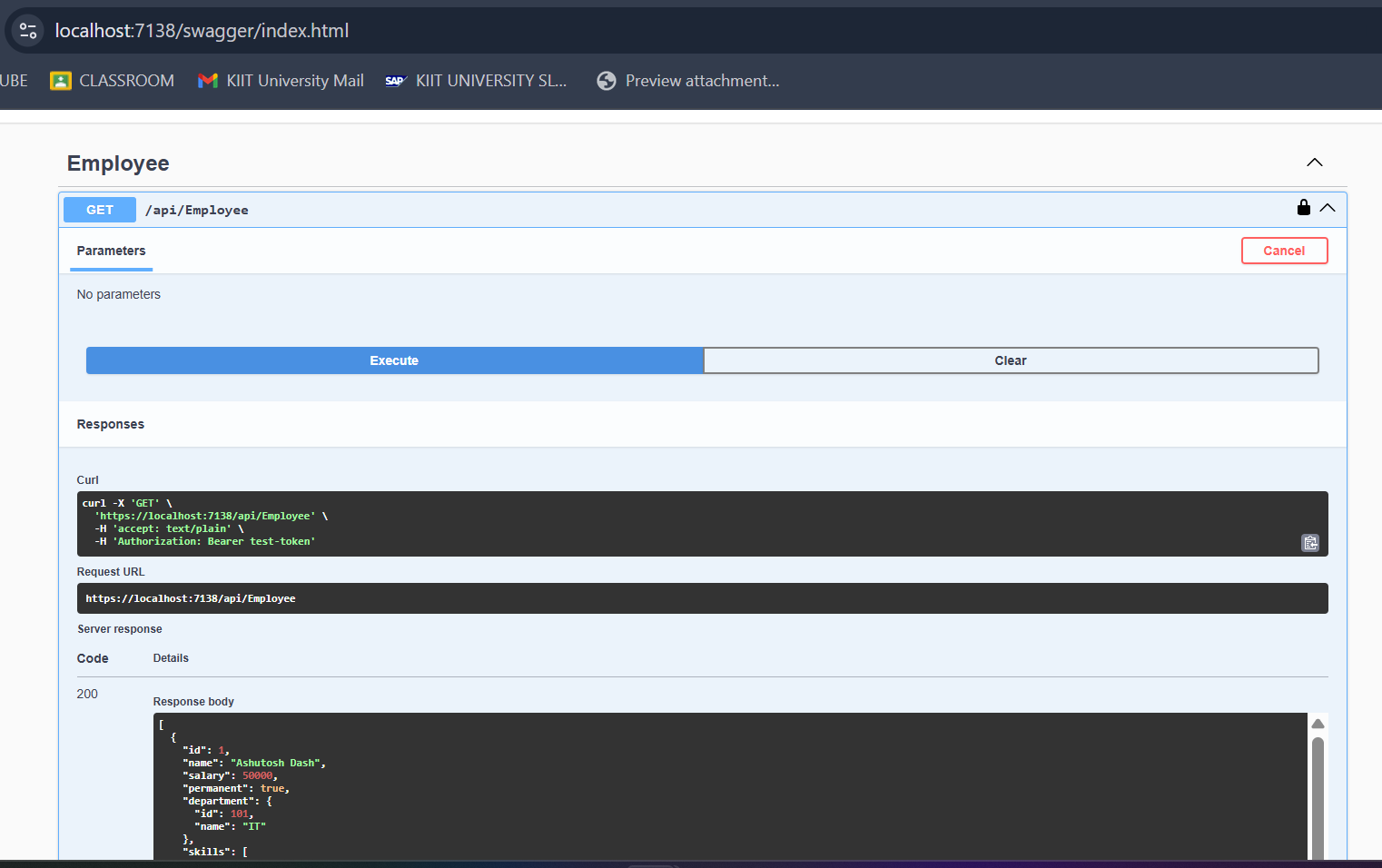
}

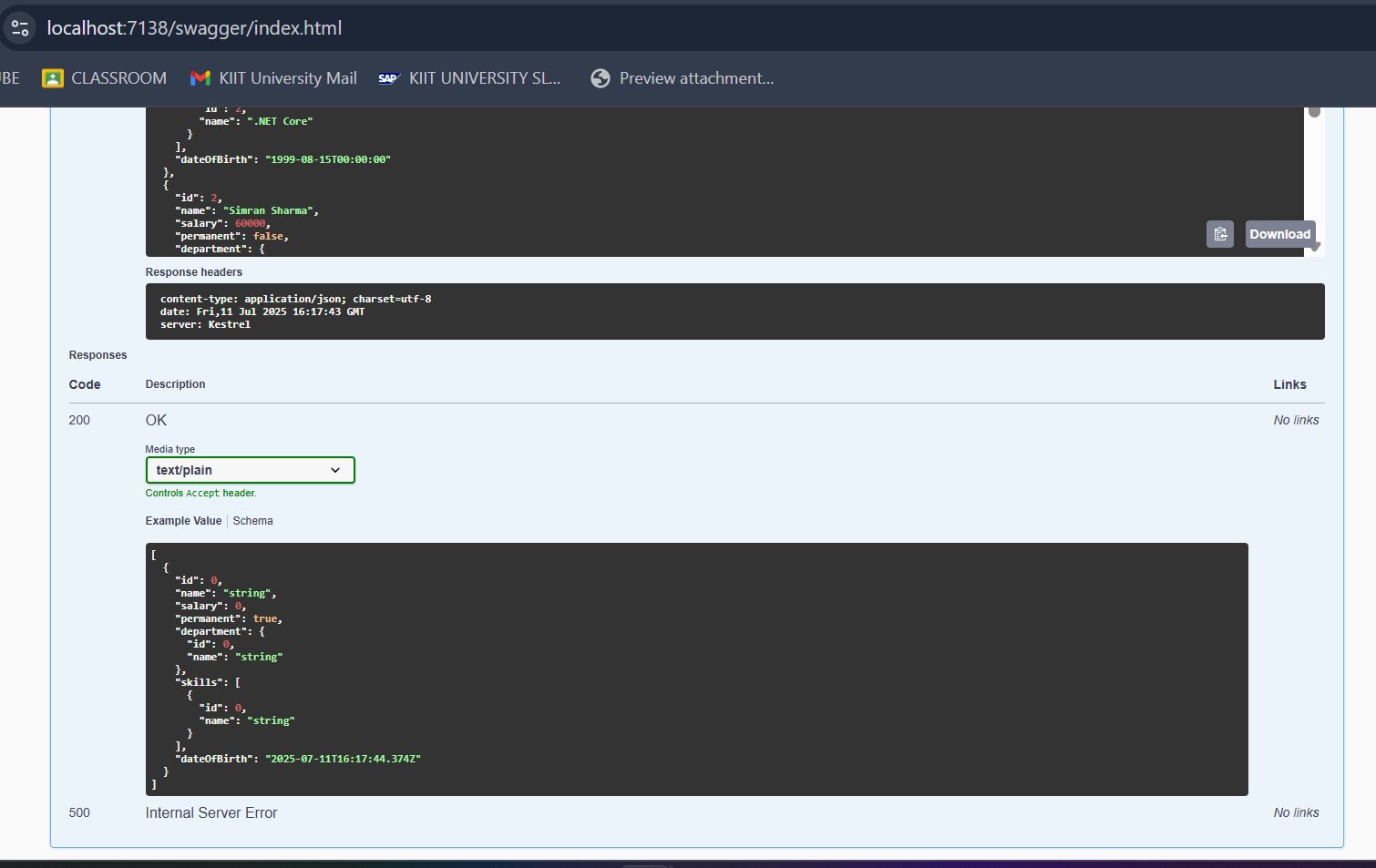
}

}

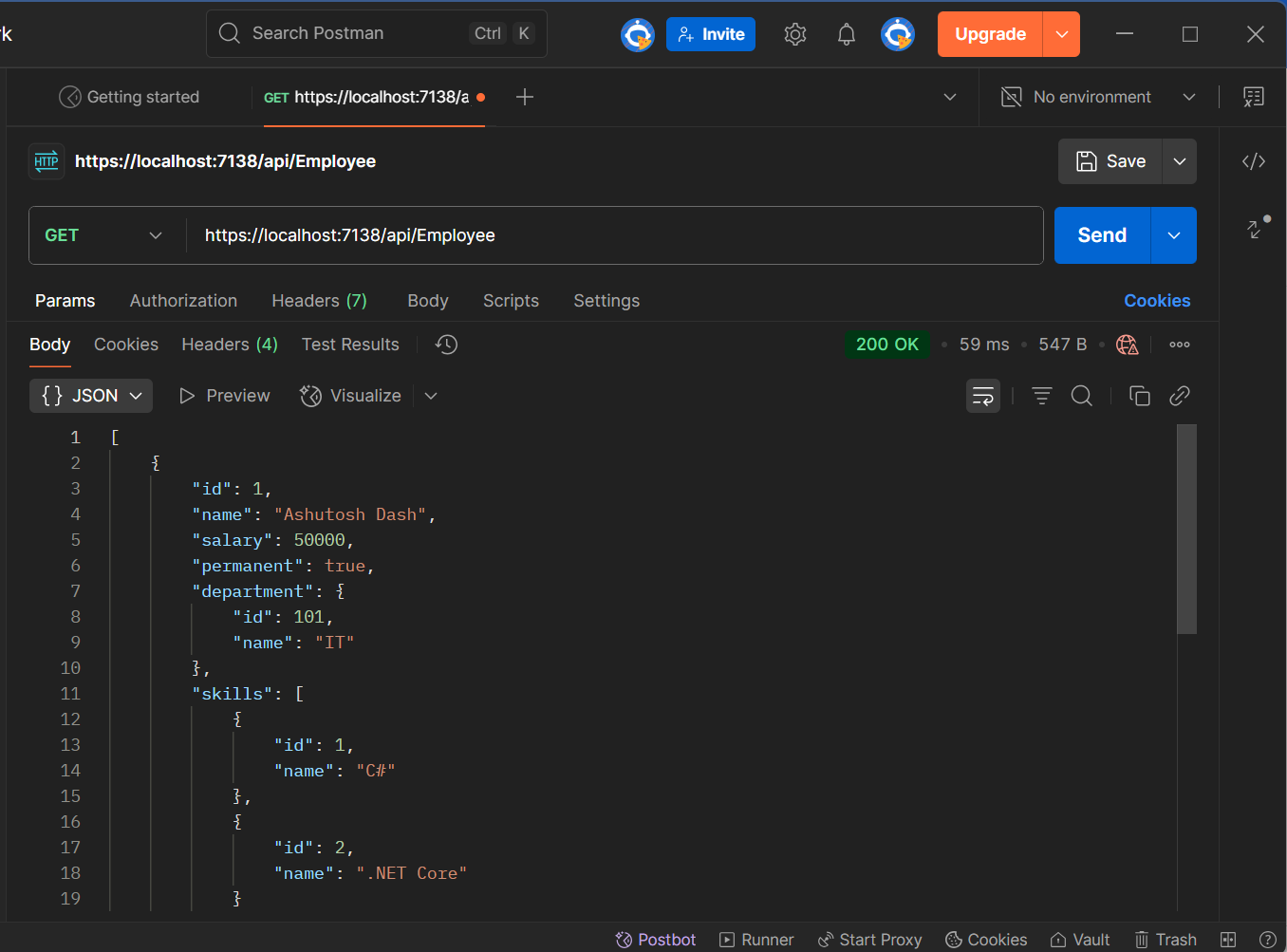
**GET**

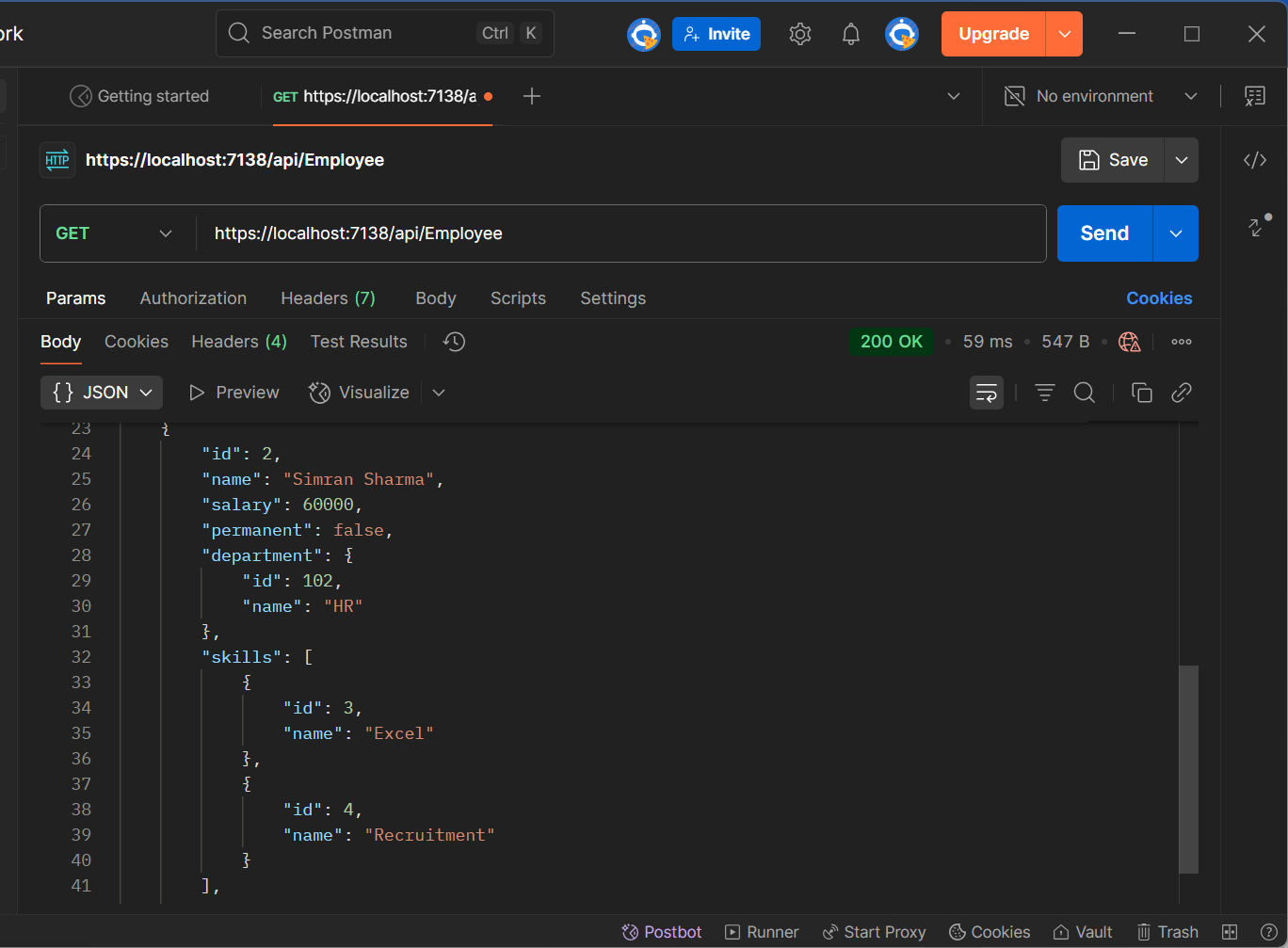
**Swagger**



****

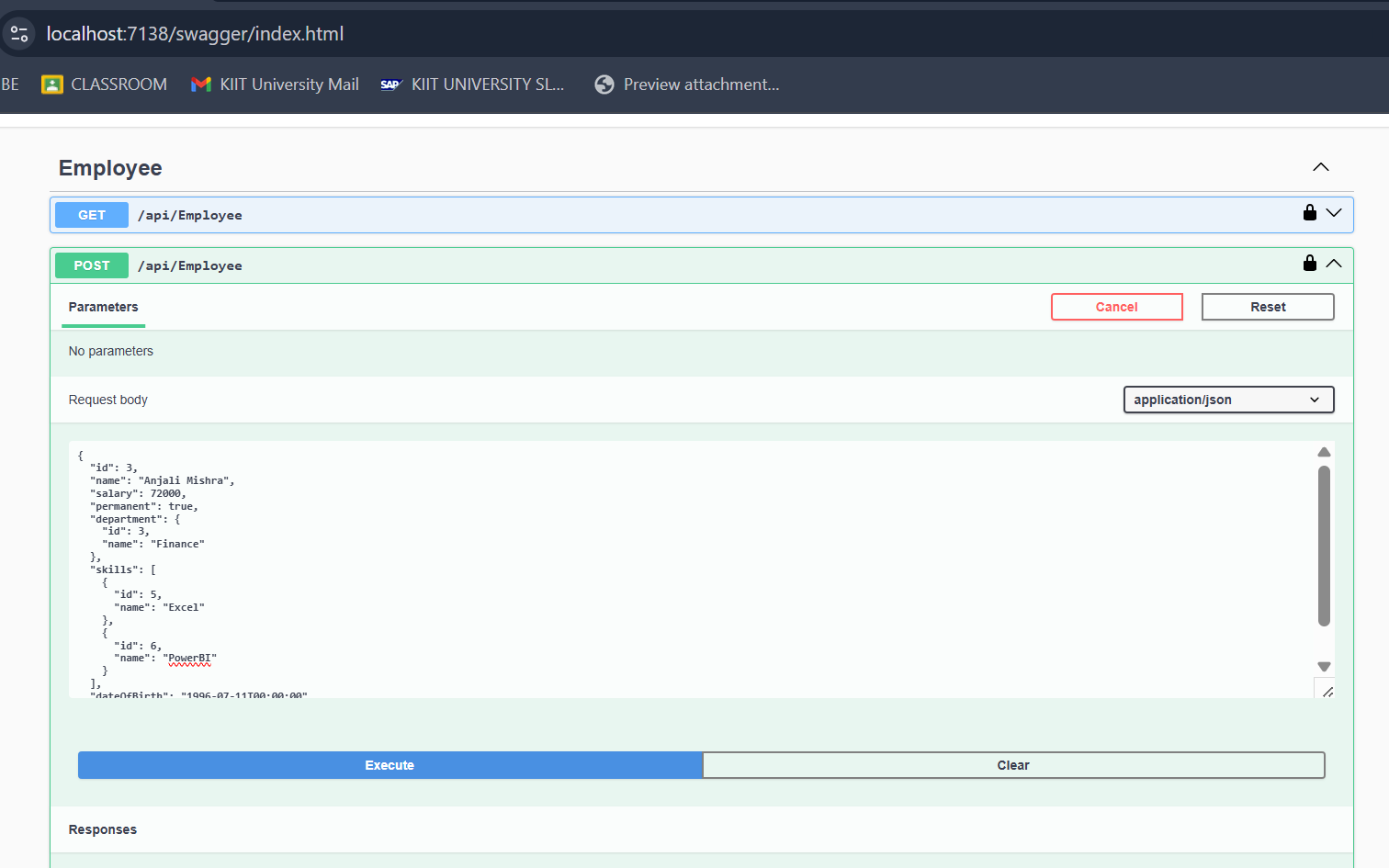
**Postman**

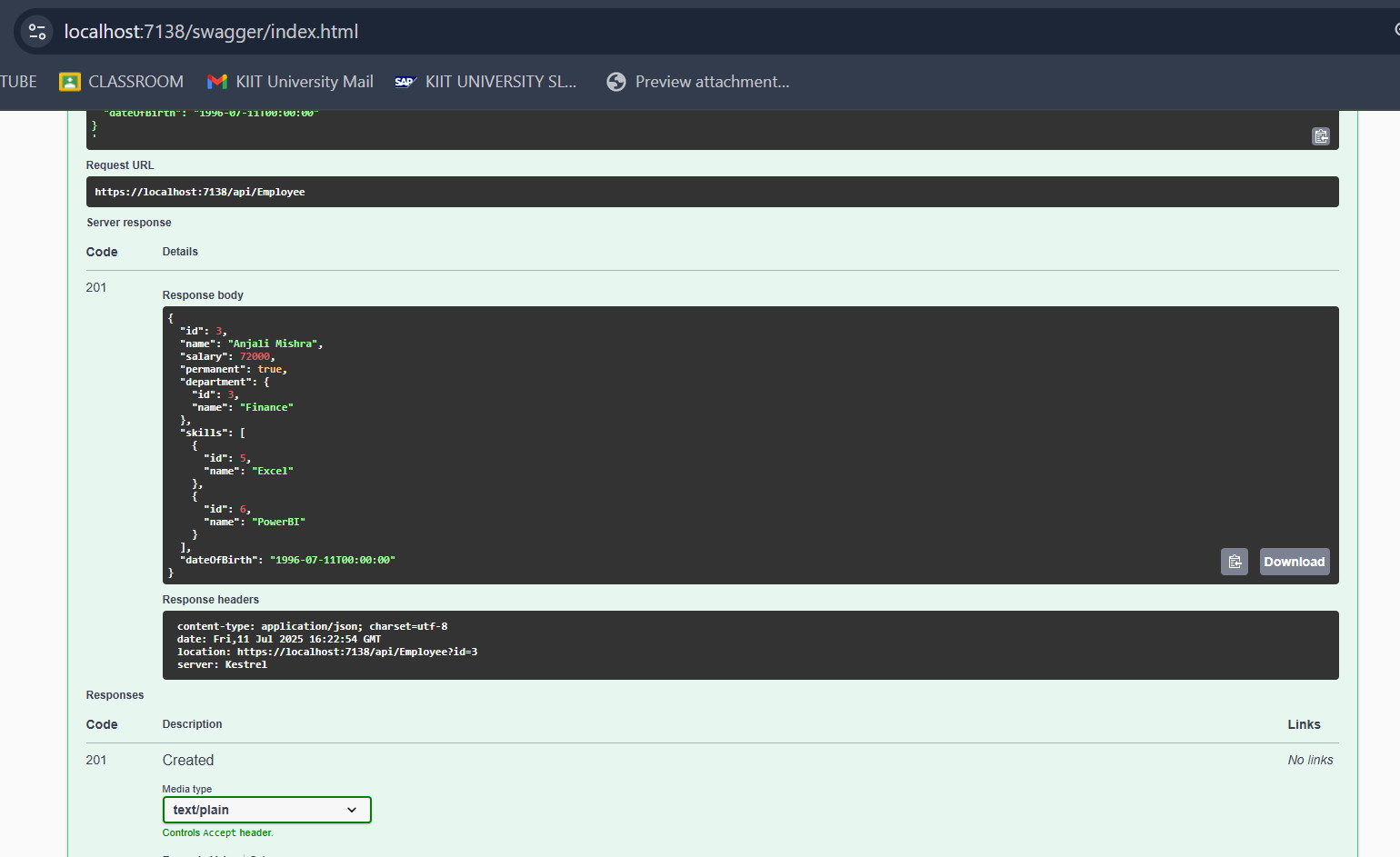
****

****

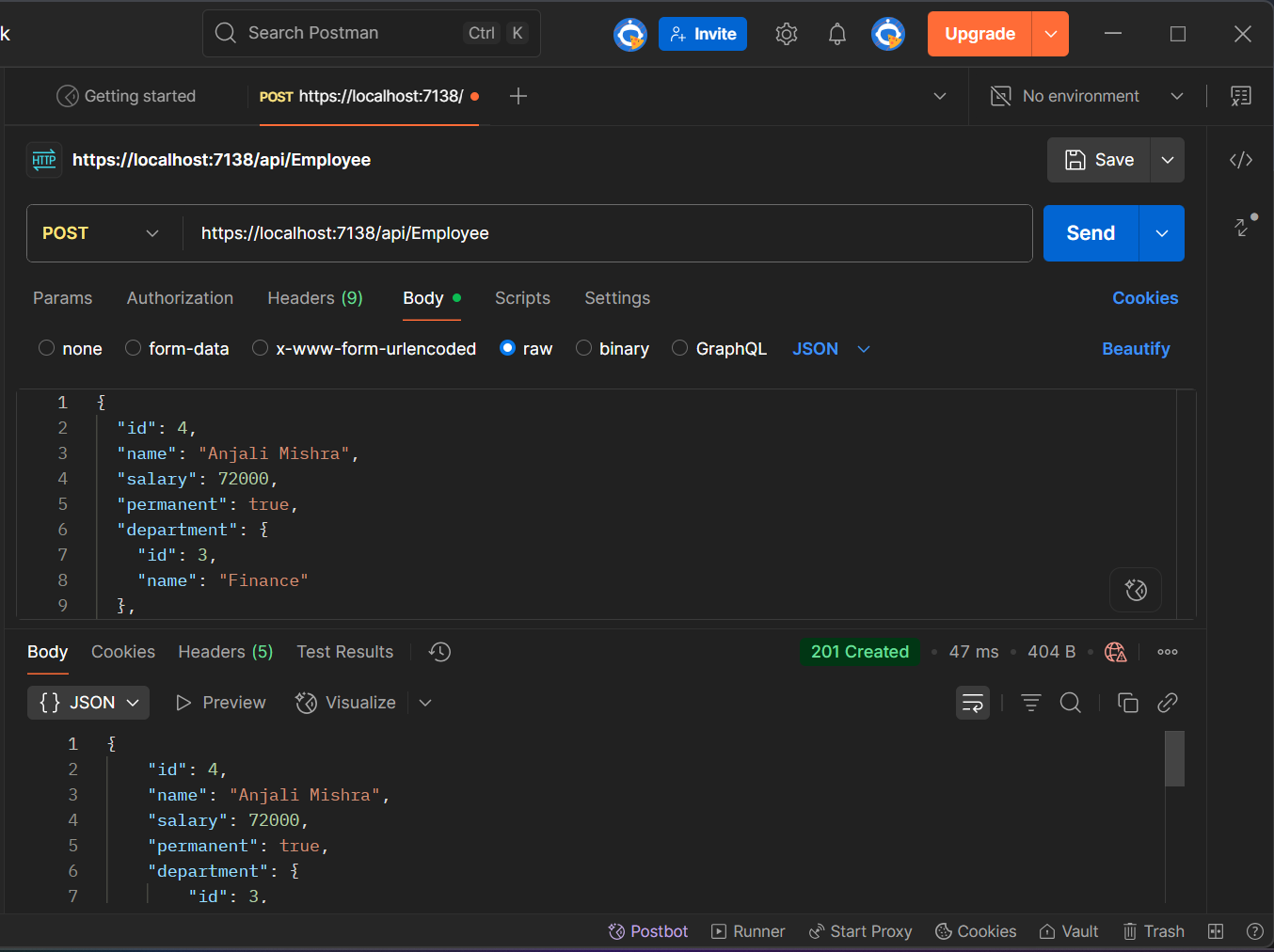
**POST**

**Swagger**

****

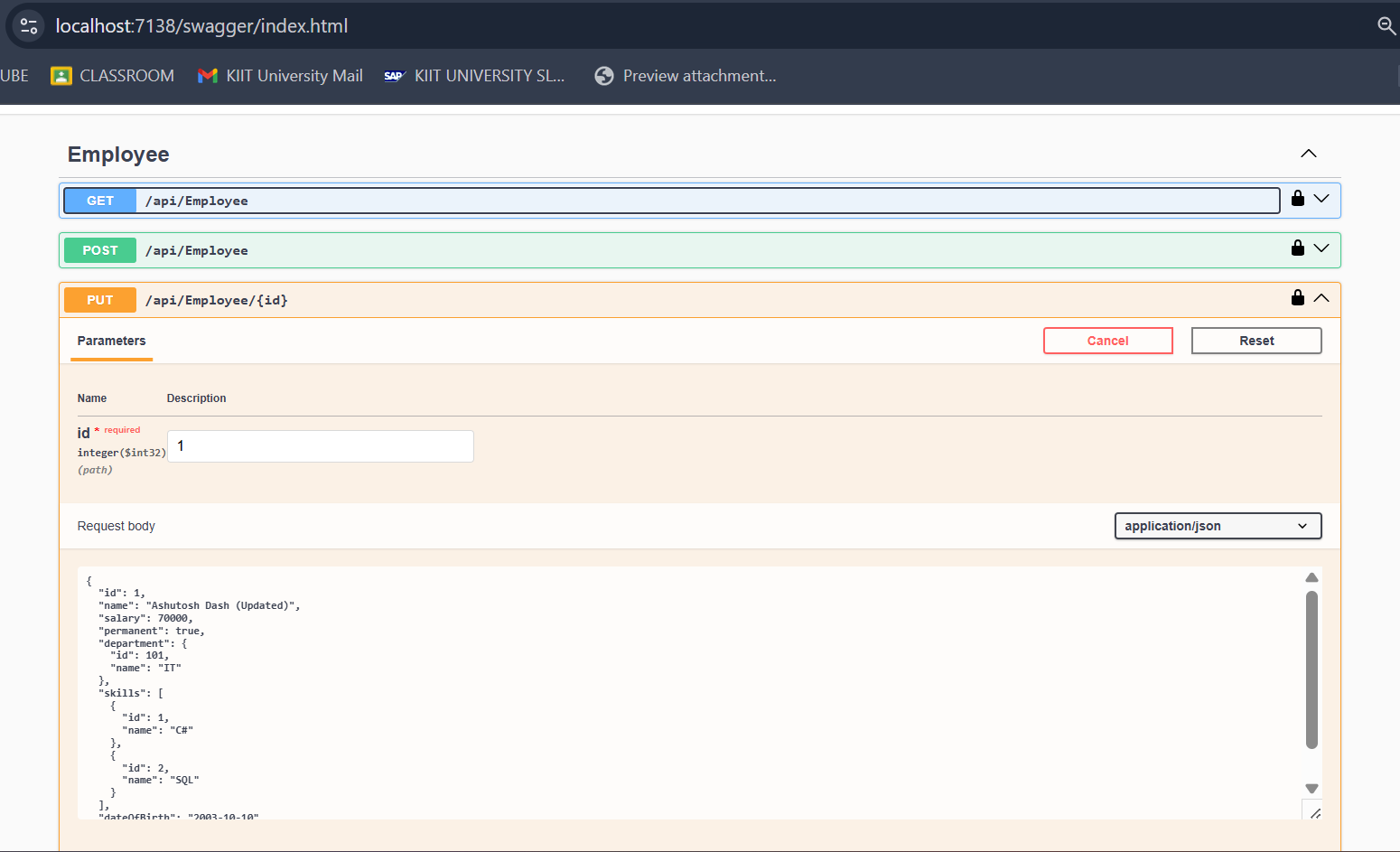
****

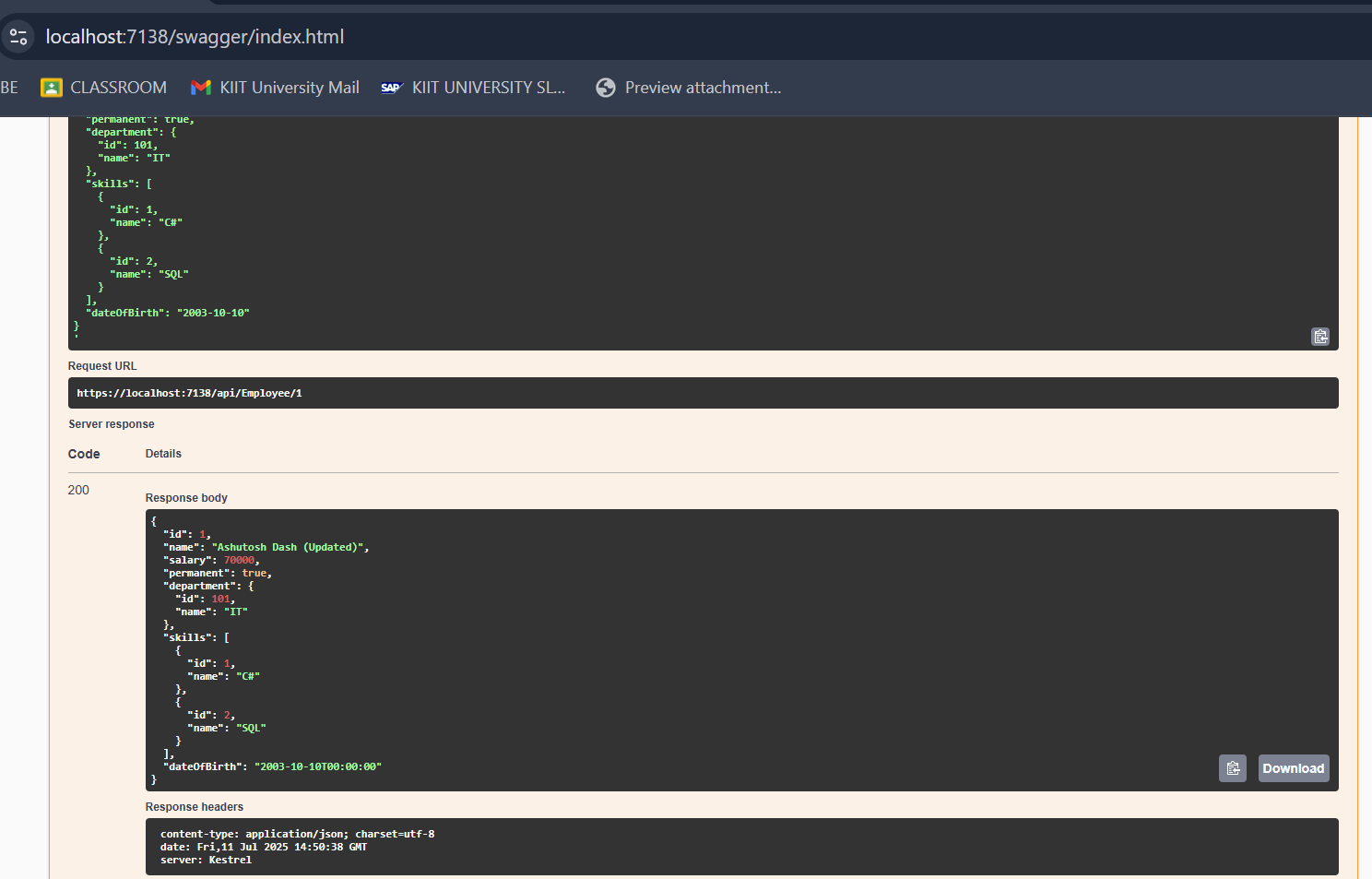
**Postman**

****

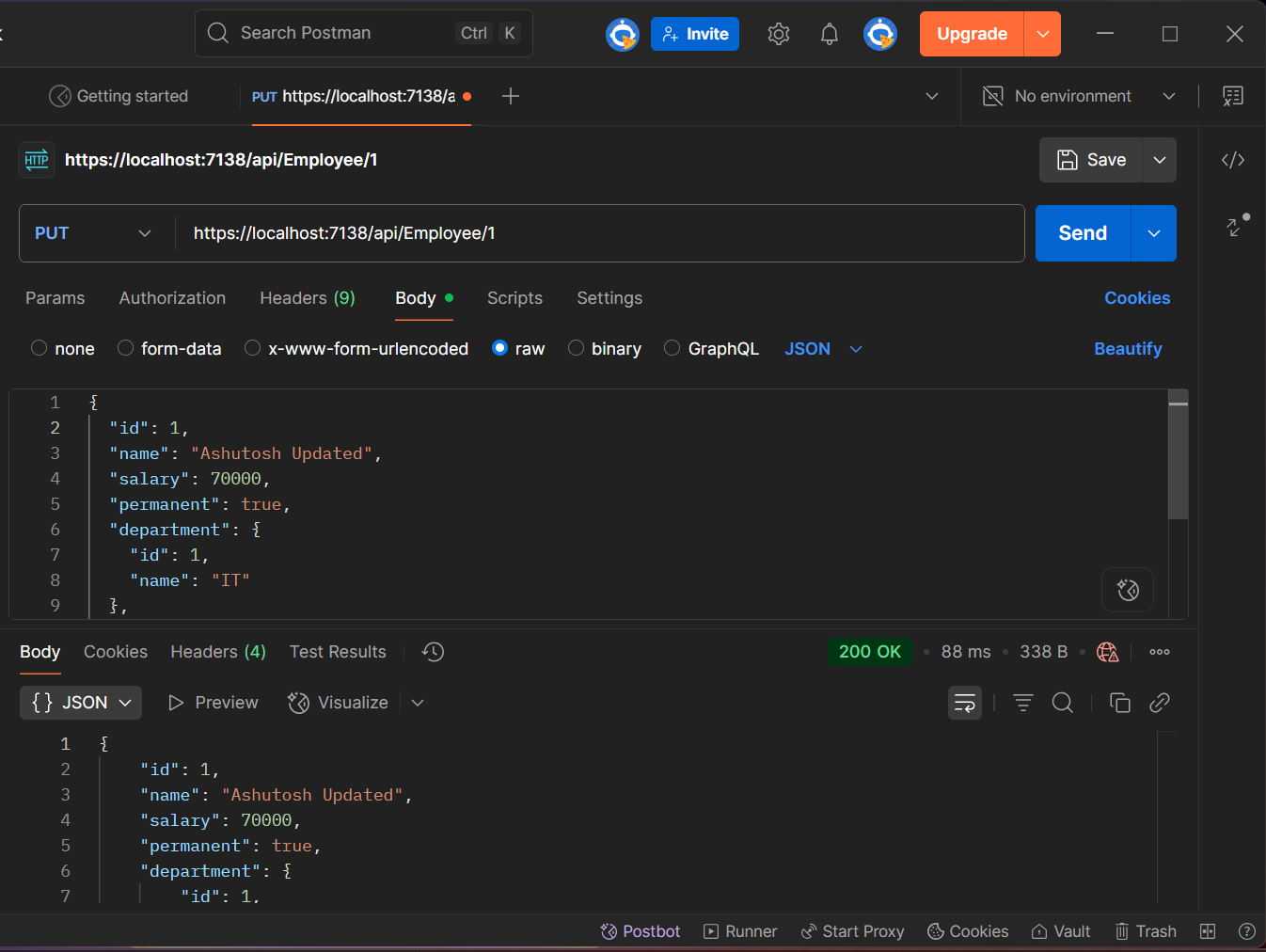
**PUT**

**Swagger**

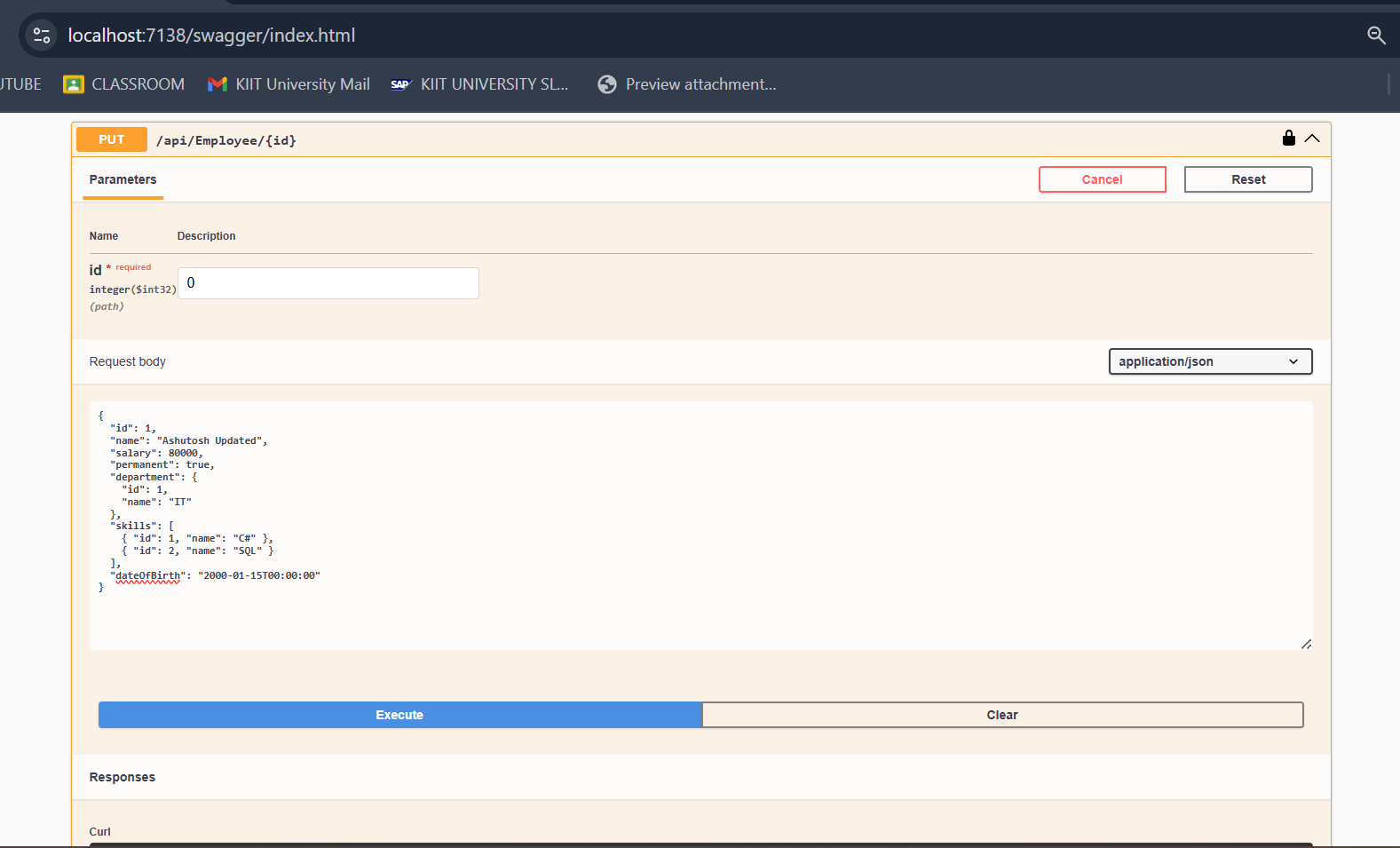


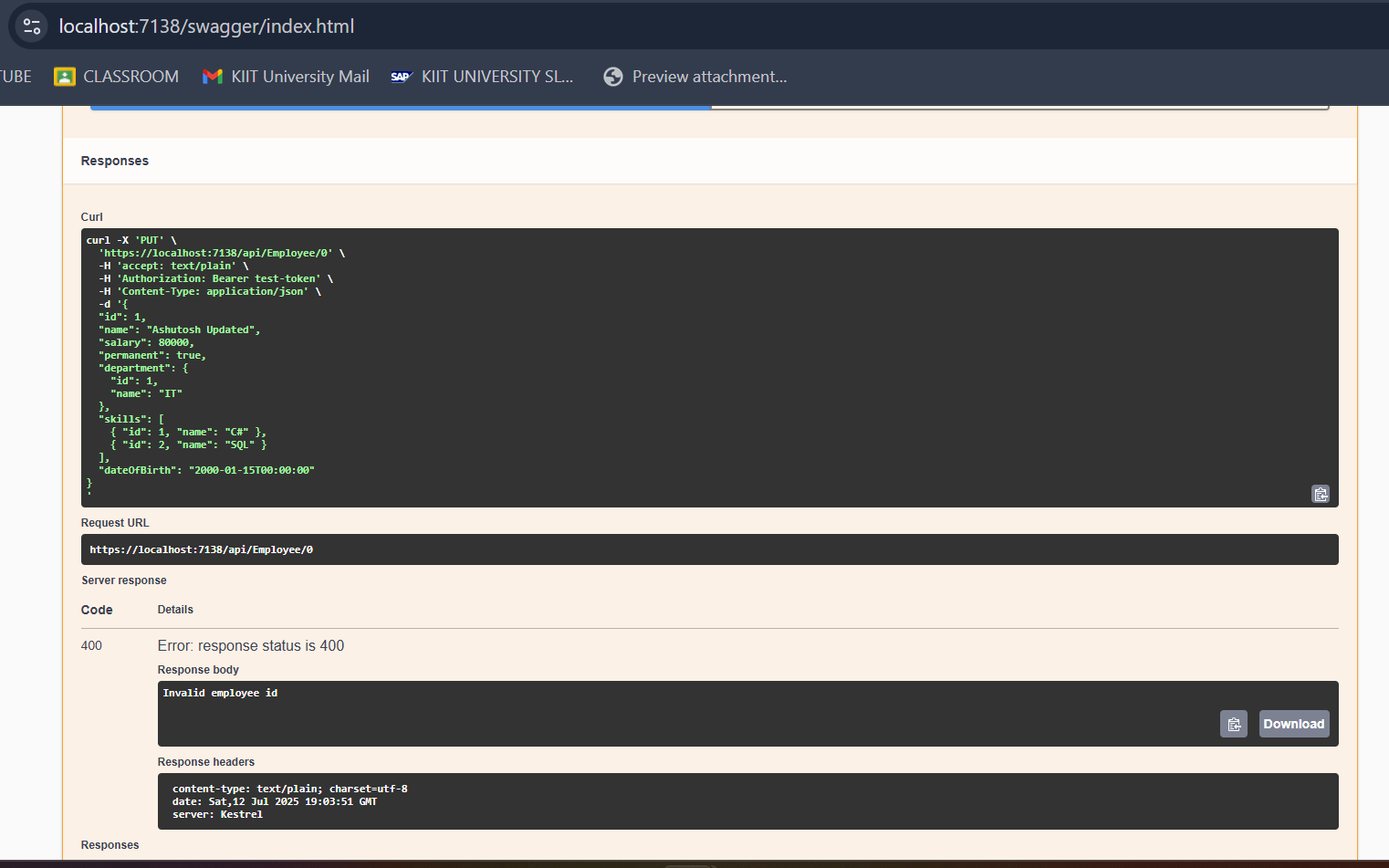


**Postman**

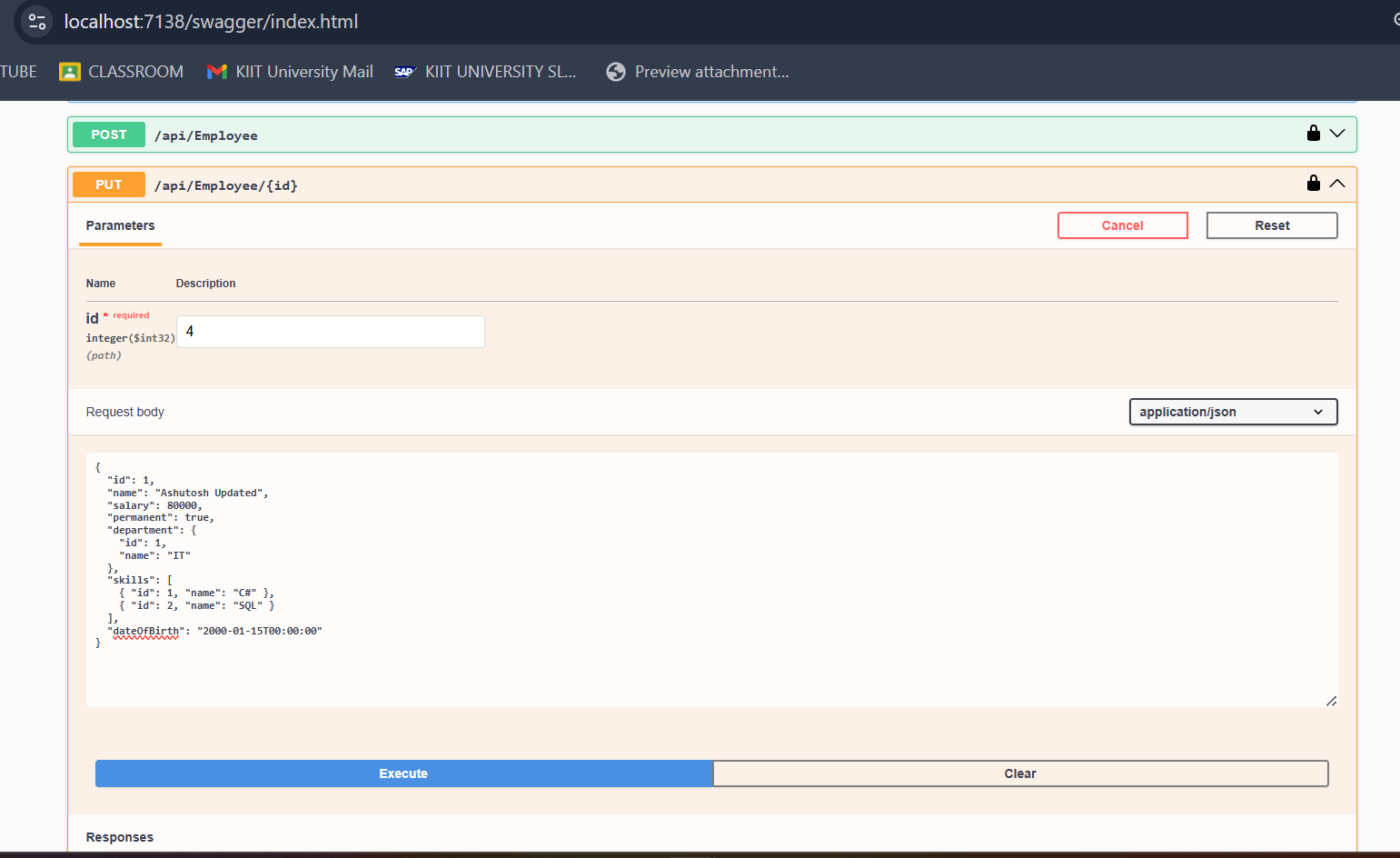
****

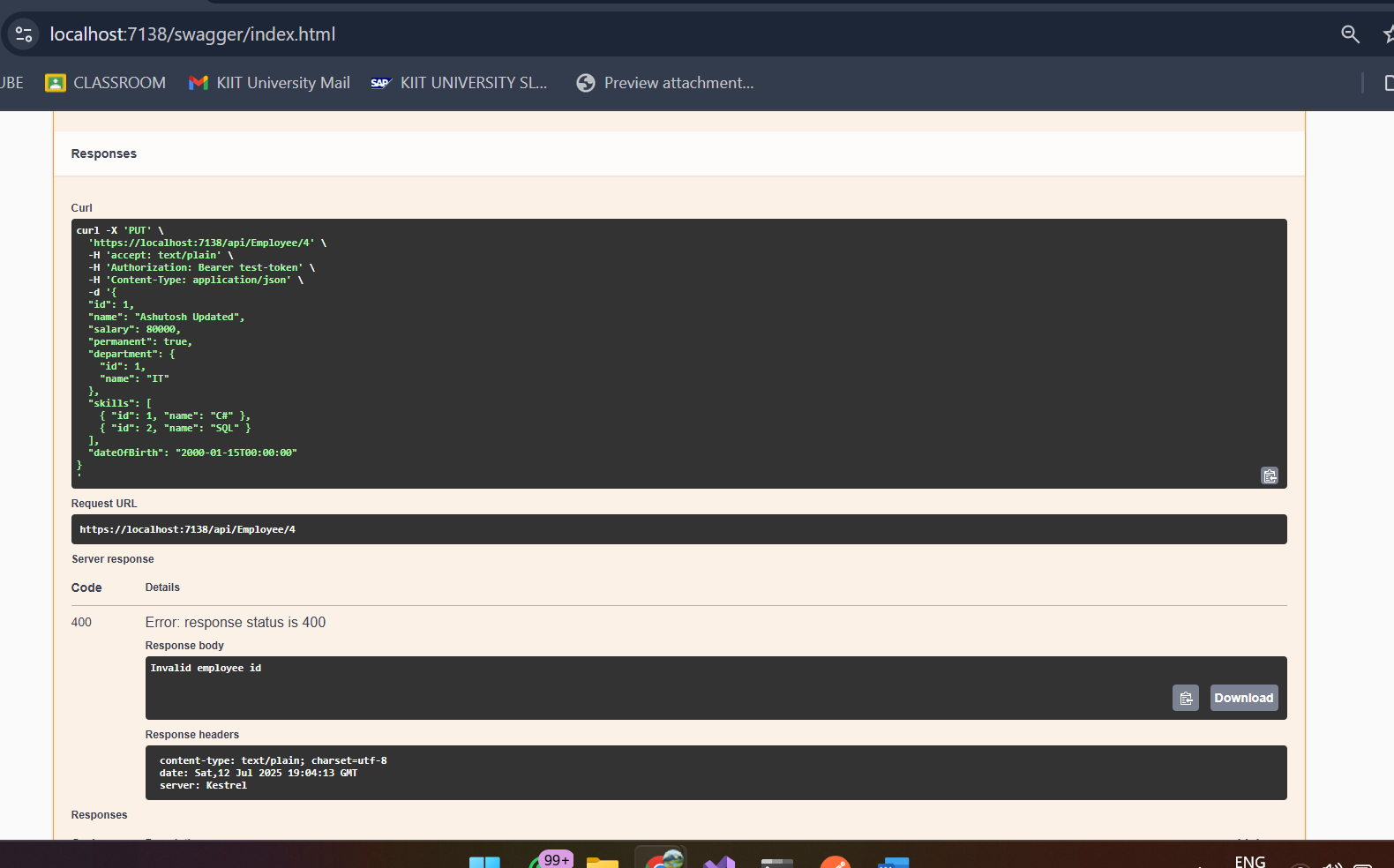
**Check if the id value is lesser than or equal to 0. If true, throw BadRequest action result with the message ‘Invalid employee id’**



****

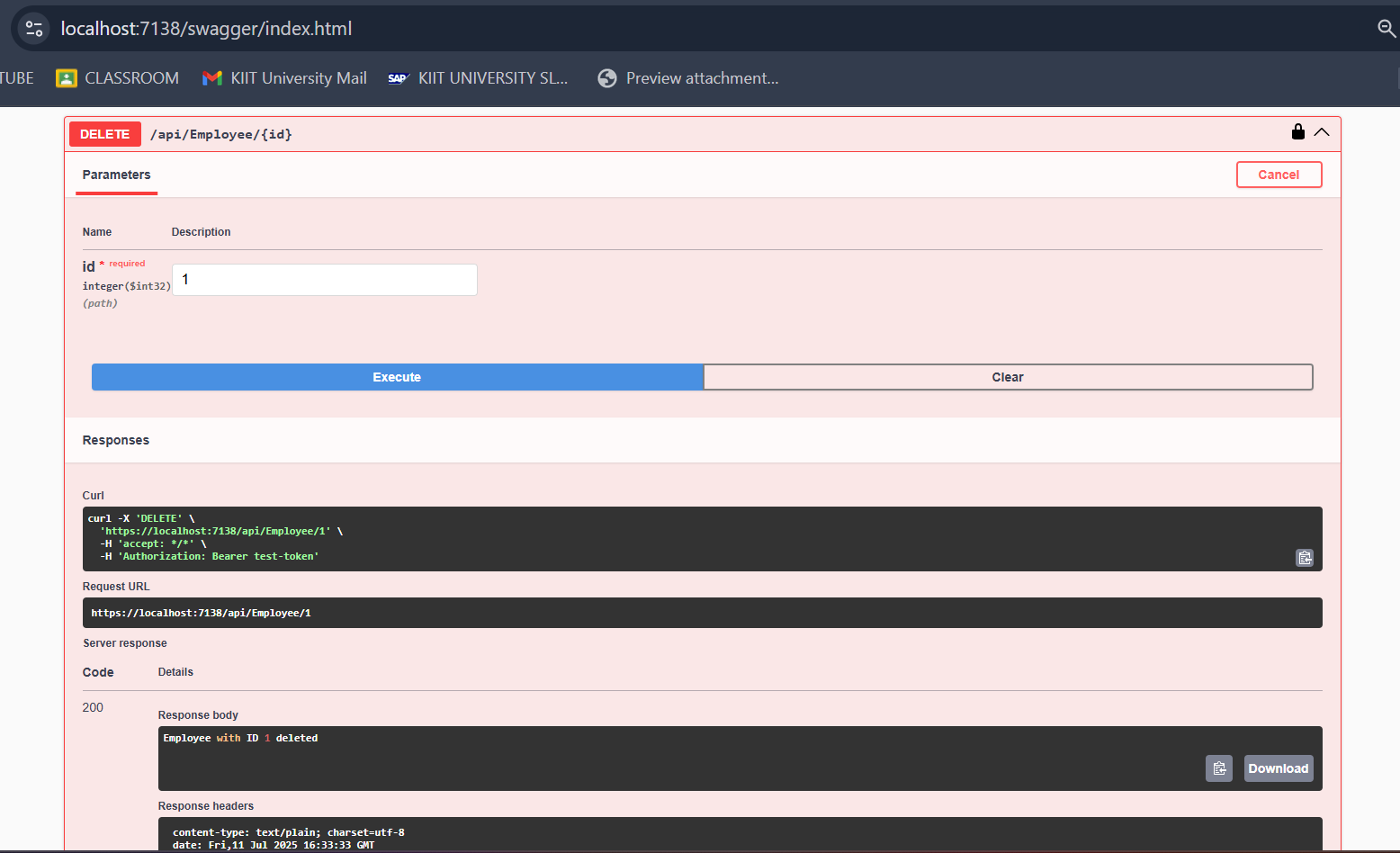
**If the value is greater than 0 but not available in the list of employee ids that is there in the hardcoded list of employees, throw BadRequest action result with the same message as stated above.**

****

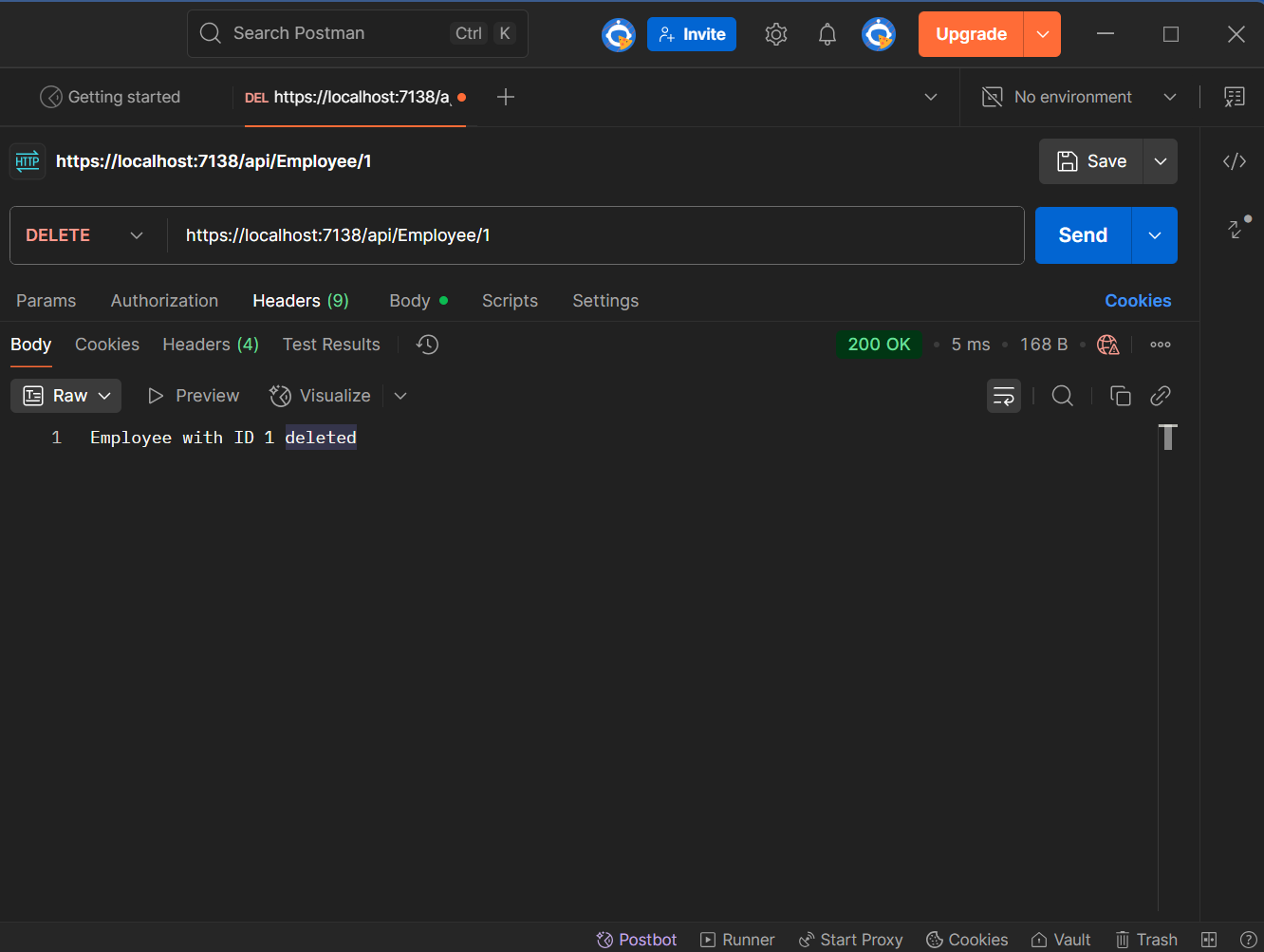
****

**DELETE**

**Swagger**

****

**Postman**

****