

```
// EmployeeService.java

package com.example.demo;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import java.util.List;

@Service
public class EmployeeService {

    @Autowired
    private EmployeeRepository employeeRepository;

    public List<Employee> getAllEmployees() {
        return employeeRepository.findAll();
    }

    public void addEmployee(Employee employee) {
        employeeRepository.save(employee);
    }

    // New method to get an employee by ID
    public Employee getEmployeeById(Long id) {
        return employeeRepository.findById(id).orElse(null);
    }
}
```

```
// EmployeeController.java

package com.example.demo;
```

```

import io.swagger.annotations.ApiOperation;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.http.ResponseEntity;

import org.springframework.web.bind.annotation.*;


import java.net.URI;

import java.util.List;


@RestController

@RequestMapping("/employees")

public class EmployeeController {


    @Autowired

    private EmployeeService employeeService;


    @GetMapping

    @ApiOperation("Get the list of all employees")

    public List<Employee> getEmployees() {

        return employeeService.getAllEmployees();

    }


    @PostMapping

    @ApiOperation("Add a new employee")

    public ResponseEntity<Object> addEmployee(@RequestBody Employee employee) {

        employeeService.addEmployee(employee);


        URI location = ServletUriComponentsBuilder

            .fromCurrentRequest()

            .path("/{id}")

            .buildAndExpand(employee.getId())

```

```

        .toUri();

        return ResponseEntity.created(location).build();
    }

    // New endpoint to get an employee by ID
    @GetMapping("/{id}")
    @ApiOperation("Get an employee by ID")
    public ResponseEntity<Employee> getEmployeeById(@PathVariable Long id) {
        Employee employee = employeeService.getEmployeeById(id);
        return ResponseEntity.ok(employee);
    }
}

```

```

// EmployeeServiceTest.java
package com.example.demo;

import org.junit.jupiter.api.Test;
import org.mockito.InjectMocks;
import org.mockito.Mock;
import org.mockito.Mockito;
import org.springframework.boot.test.context.SpringBootTest;

import java.util.Arrays;
import java.util.List;
import java.util.Optional;

import static org.junit.jupiter.api.Assertions.assertEquals;

@SpringBootTest

```

```

class EmployeeServiceTest {

    @Mock
    private EmployeeRepository employeeRepository;

    @InjectMocks
    private EmployeeService employeeService;

    @Test
    void testGetAllEmployees() {
        // Arrange
        Employee employee1 = new Employee("1", "John", "Doe", "john.doe@example.com", "Software Engineer");
        Employee employee2 = new Employee("2", "Jane", "Smith", "jane.smith@example.com", "Product Manager");

        Mockito.when(employeeRepository.findAll()).thenReturn(Arrays.asList(employee1, employee2));

        // Act
        List<Employee> result = employeeService.getAllEmployees();

        // Assert
        assertEquals(2, result.size());
    }

    @Test
    void testGetEmployeeById() {
        // Arrange
        Employee employee = new Employee("1", "John", "Doe", "john.doe@example.com", "Software Engineer");
        Long employeeId = 1L;

```

```
Mockito.when(employeeRepository.findById(employeeId)).thenReturn(Optional.of(employee));
```

```
// Act
```

```
Employee result = employeeService.getEmployeeById(employeeId);
```

```
// Assert
```

```
assertEquals(employeeId, result.getId());
```

```
assertEquals("John", result.getFirstName());
```

```
// Add assertions for other fields
```

```
}
```

```
}
```