# Documentation for src\main\java\com\example\EmployeeManagementSystem\service\ServiceEmployeeIn.java

Generated on: 2025-09-05 22:42:34

Repository: C:\Users\User\VisualStudio\Employee-Management-Sys\EmployeeManagementSystem

Language: java

---

# createEmployee

## Description

This function is used to create a new employee in the Employee Management System using the provided employee data transfer object (dtoEmployee).

## Parameters

* dtoEmployee: The data transfer object containing the information of the employee to be created.

## Usage Example

DtoEmployee employeeData = new DtoEmployee("John Doe", "john.doe@example.com", "Manager");

createEmployee(employeeData);

## Important Notes

* Make sure to provide all necessary information in the dtoEmployee parameter to successfully create a new employee.
* This function does not handle any validation or error checking, so ensure the data provided is accurate before calling the function.\n\n---\n\n# findById Function Documentation

## Description

The findById function is used to retrieve an employee record from the Employee Management System database based on the provided employee ID.

## Parameters

* `id`: The unique identifier of the employee whose record needs to be retrieved.

## Usage Example

Employee employee = findById(123);

System.out.println(employee);

## Important Notes

* This function will return null if no employee record is found for the provided ID.
* Ensure that the ID provided is a valid and existing employee ID in the database.\n\n---\n\n# findAllEmployee Function

## Description

This function retrieves all employees from the Employee Management System.

## Parameters

* None

## Usage Example

List<Employee> allEmployees = findAllEmployee();

for(Employee employee : allEmployees) {

System.out.println(employee.getName());

}

## Notes

* This function does not require any parameters as it retrieves all employees in the system.\n\n---\n\n# updateEmployee Function

## Description

The updateEmployee function is used to update the information of an employee in the Employee Management System.

## Parameters

* `id`: The unique identifier of the employee to be updated.
* `updatedemployee`: The updated information of the employee.

## Usage Example

// Update employee with id 1234

Employee updatedEmployee = new Employee("John Doe", "john.doe@example.com", "Manager");

serviceEmployee.updateEmployee(1234, updatedEmployee);

## Important Notes

* Make sure to provide the correct `id` of the employee to be updated.
* The `updatedemployee` parameter should contain the new information of the employee.\n\n---\n\n# deleteEmployee Function

## Description

The deleteEmployee function is used to remove an employee from the employee management system based on their unique identifier (id).

## Parameters

* `id`: The unique identifier of the employee to be deleted.

## Usage Example

// Delete employee with id 123

deleteEmployee(123);

## Important Notes

* This function permanently removes the employee from the system and cannot be undone.
* Ensure that the correct id is provided to avoid deleting the wrong employee.\n\n---\n\n