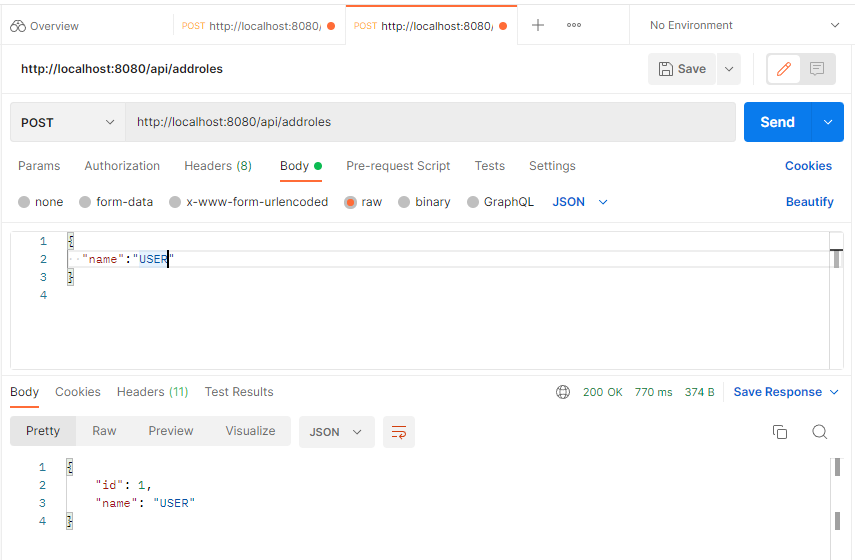
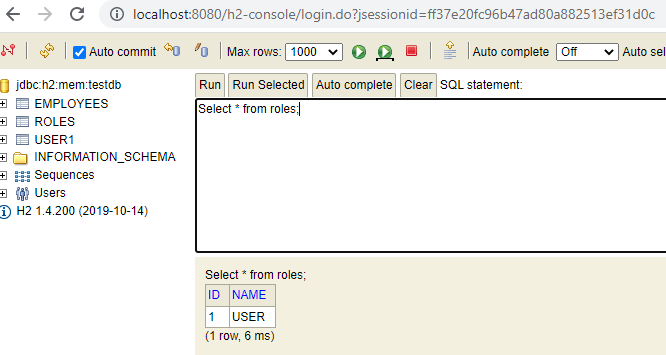
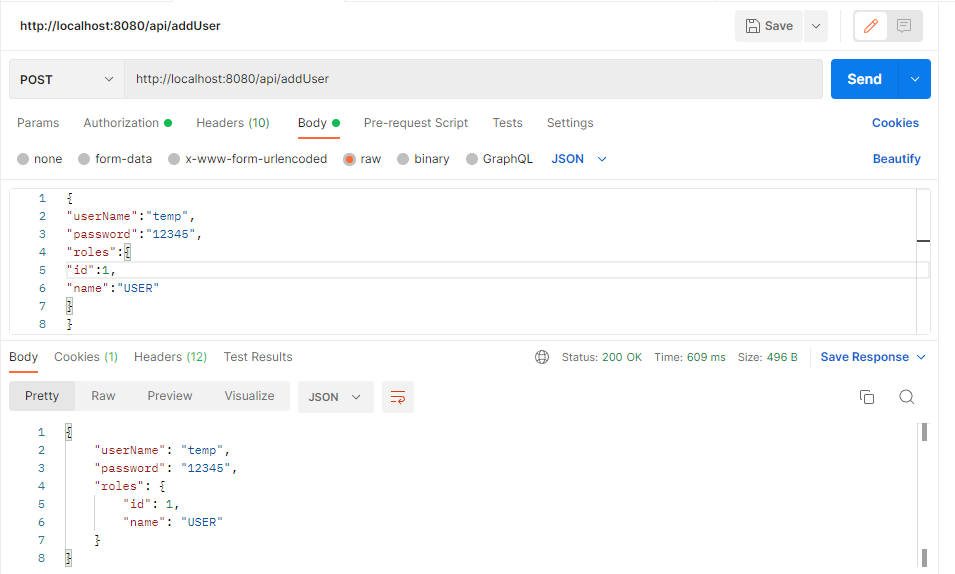
**SCREENSHOTS of the Listed Questions.**

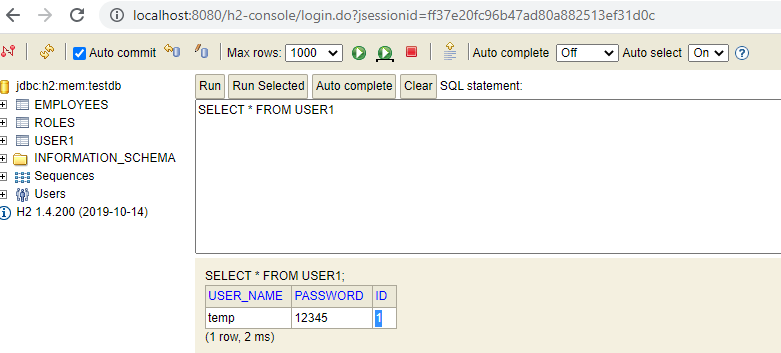
1.Your application should be able to add roles in the database dynamically in the db.





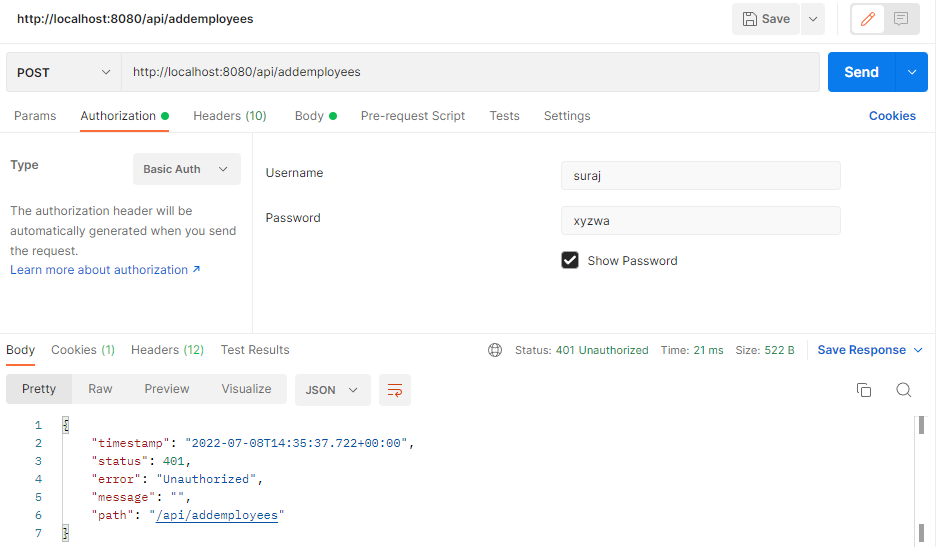
2. Your application should be able to add Users in the db which can be used for authentication purposes.



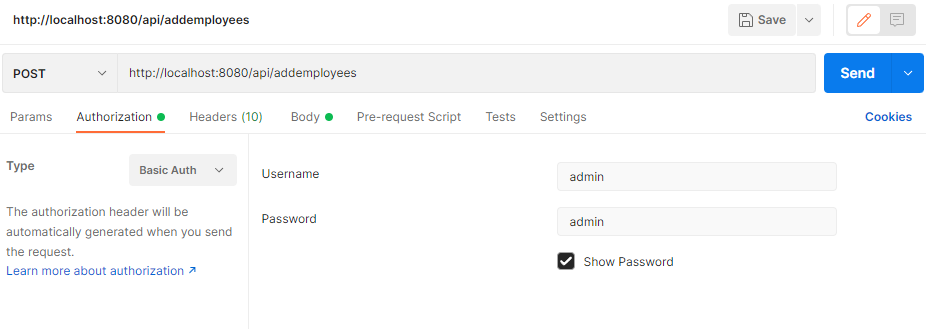


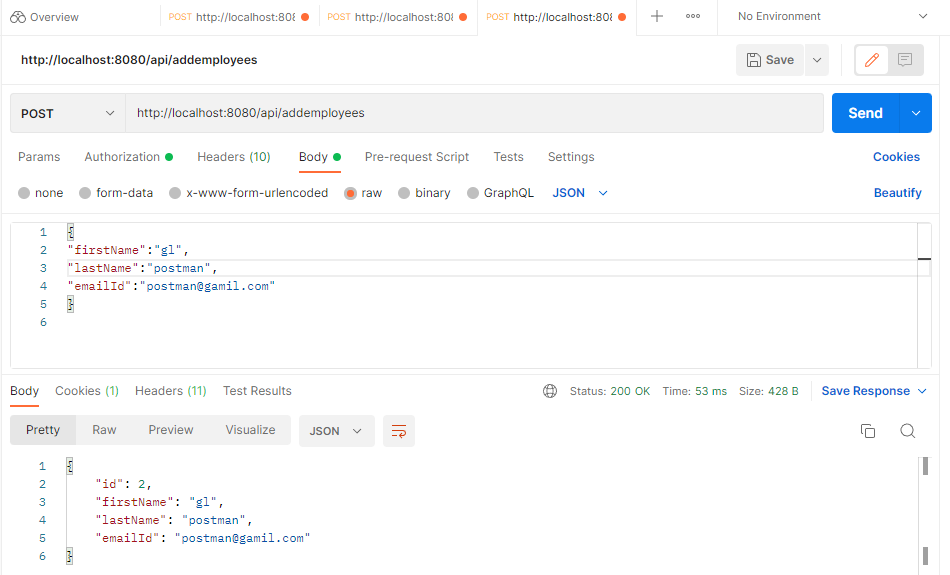
3. Now Your application should be able to add employees data in the db if and only if the authenticated user is **ADMIN**-

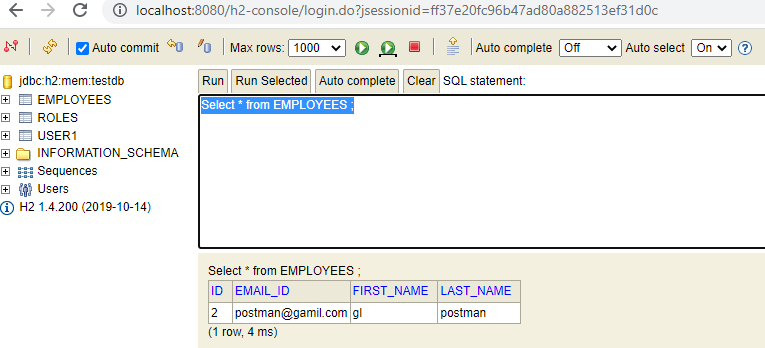
Unauthorized request if username and password not correct.



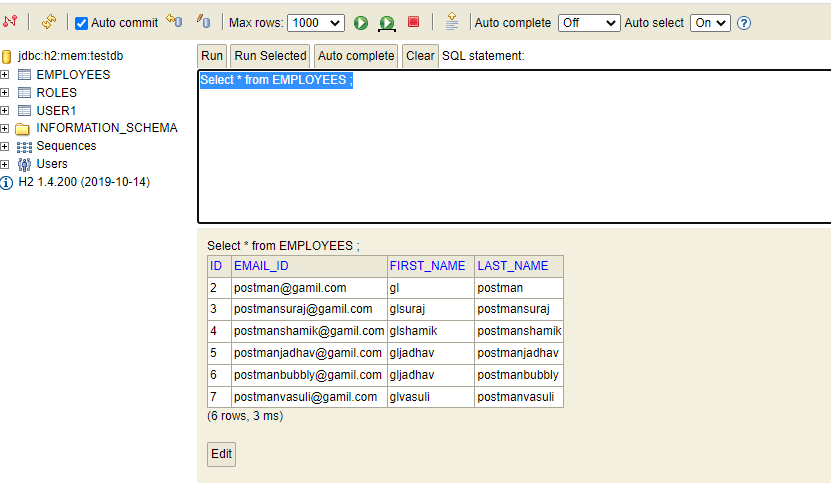
Authorized request only when username and password is ADMIN.

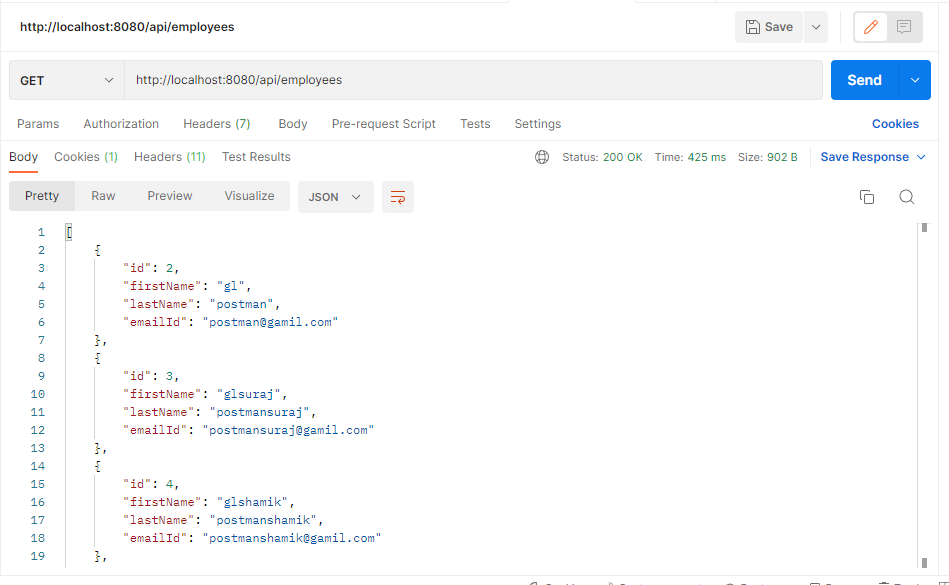




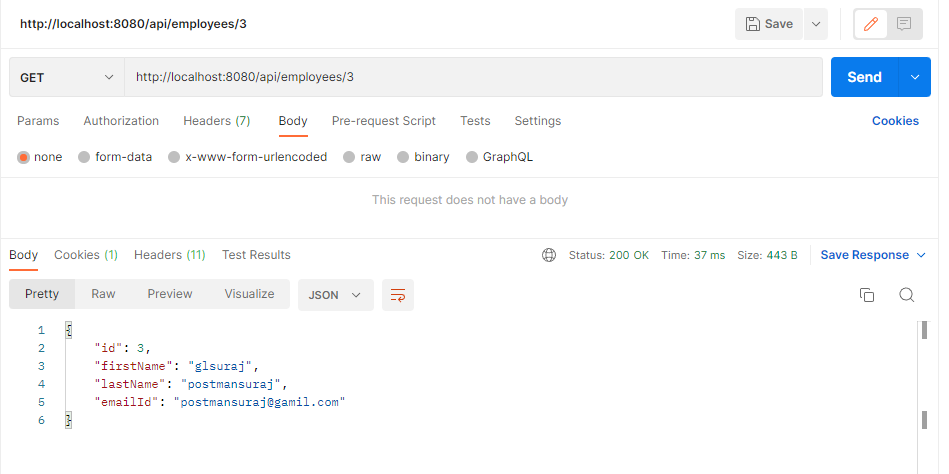


4.Your application should provide an endpoint to list all the employees stored in the database.

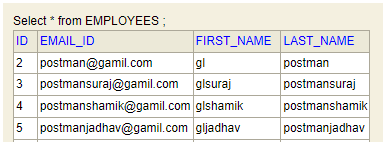




5. Your application should provide endpoint to fetch or get an employee record specifically based on the id of that employee.



6. Your application should provide an endpoint to update an existing employee record with the given updated json object.



Updating employee with employeeId-2. Changing name to glsuraj2

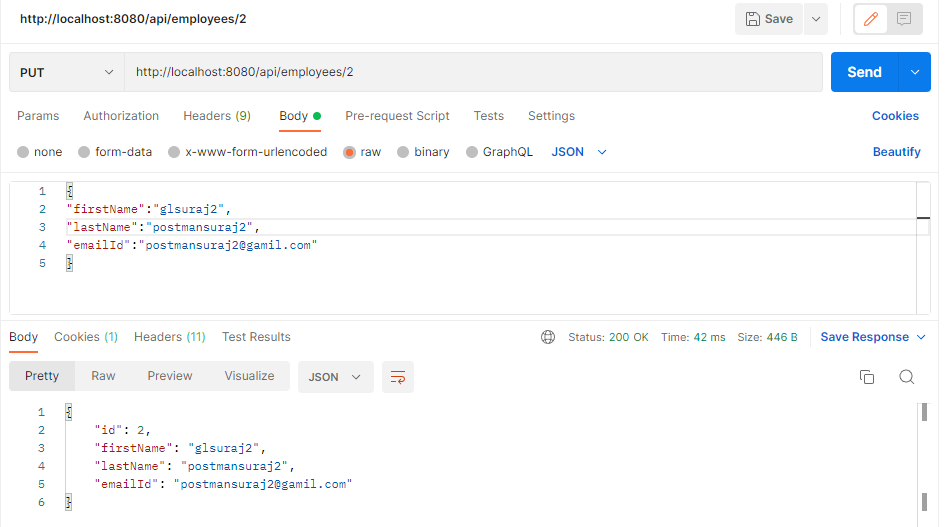
{

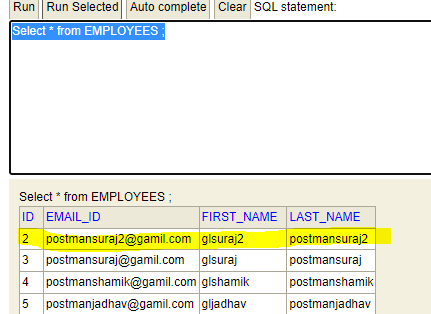
"firstName":"glsuraj2",

"lastName":"postmansuraj2",

"emailId":"postmansuraj2@gamil.com"

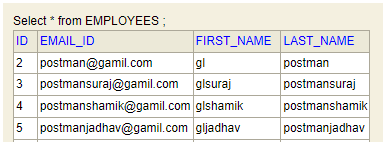
}

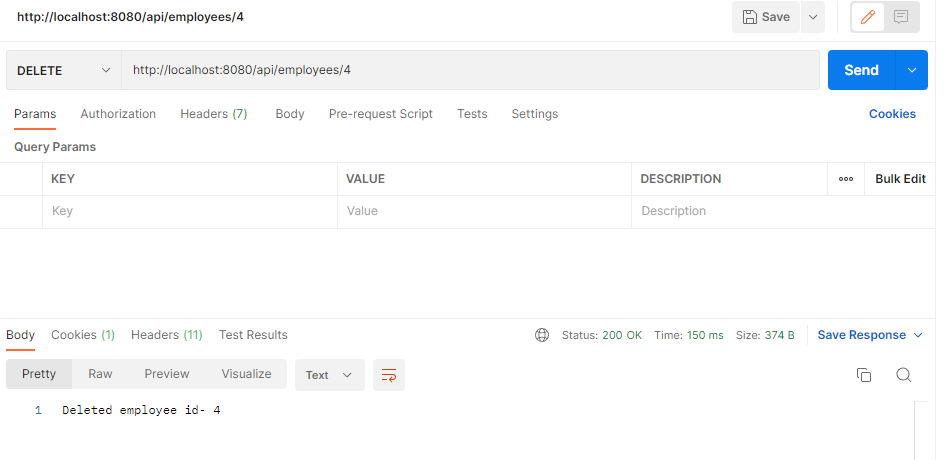


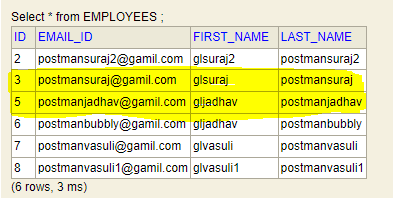


7. Your application should also provide an endpoint to delete an existing employee record based on the id of the employee.

Deleting employee with Id-4

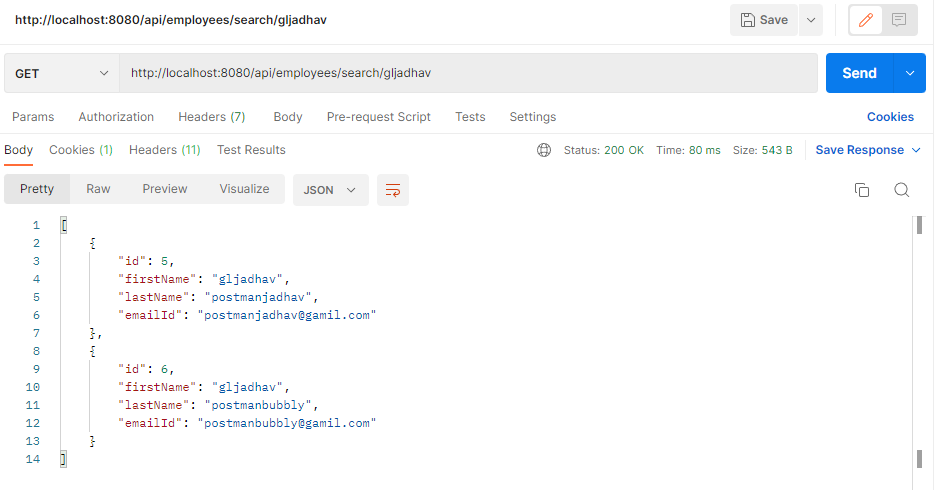






8. Your application should provide an endpoint to fetch an employee by his/her first name and if found more than one record then list them all-

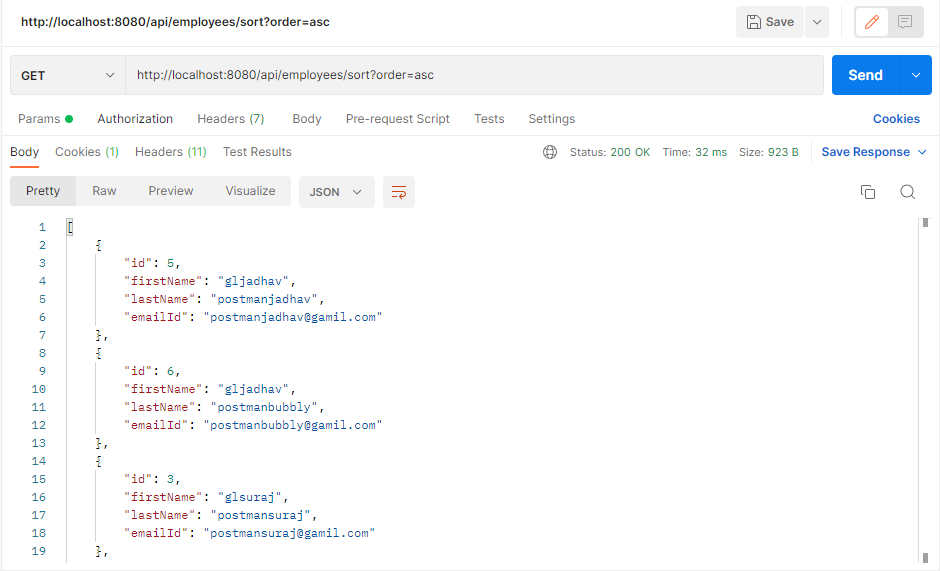
Fetching employee with firstname -**glJadhav.**

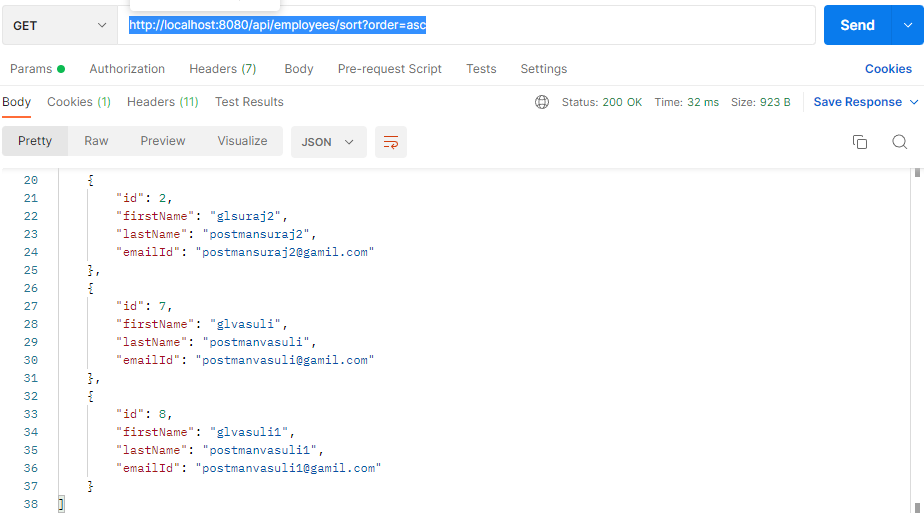


9. Your application should be able to list all employee records sorted on their first name in either ascending order or descending order.

**Employees in ascending order**

http://localhost:8080/api/employees/sort?order=asc





**Employees in ascending order**

http://localhost:8080/api/employees/sort?order=desc

