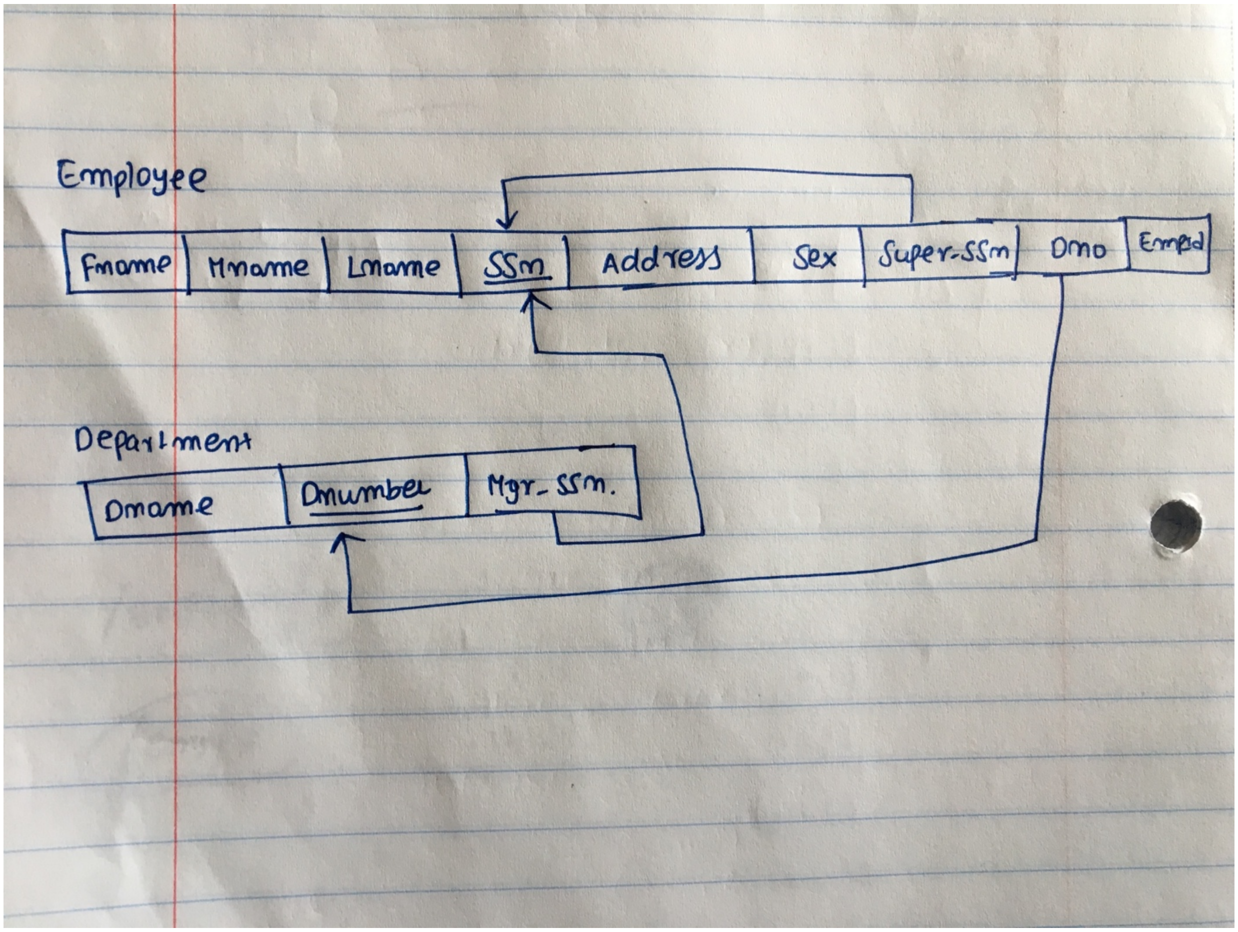
PROJECT

EMPLOYEE MANAGEMENT

Introduction: "Employee Management"

Objective: The objective of this project is to develop RESTFUL API's to perform GET, PUT, POST, DELETE, PATCH operations on employee and department entities. The schema of the tables is as follows. The underlined attributes represent a primary key and the arrows indicate a foreign key.



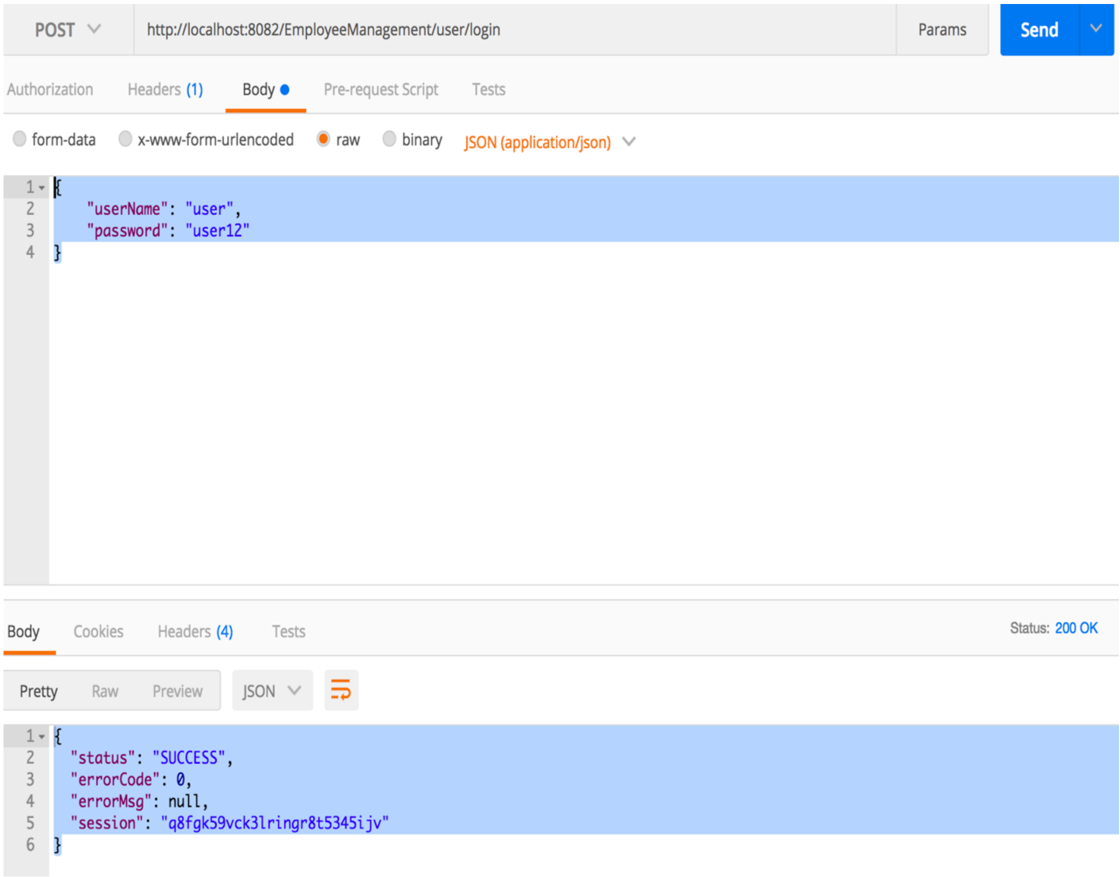
Technologies:

1. JSON
2. Spring MVC
3. Maven
4. Servlet
5. Log4J
6. Apache Tomcat
7. Java
8. Hibernate

Features Implemented:

1. **Session Management :** The API's are restricted so that only the authenticated users can access the API. This has been implemented by creating a unique session id for the user once the user is successfully authenticated and sending that session id as part of the response. The subsequent api requests from the user has to have this session id appended to the URI for the API's to be accessed.

   Example: When the user logs in initially by hitting the api then a session-id is generated
   http://localhost:8082/EmployeeManagement/user/login



POST http://localhost:8082/EmployeeManagement/user/login

Body: raw, JSON (application/json)

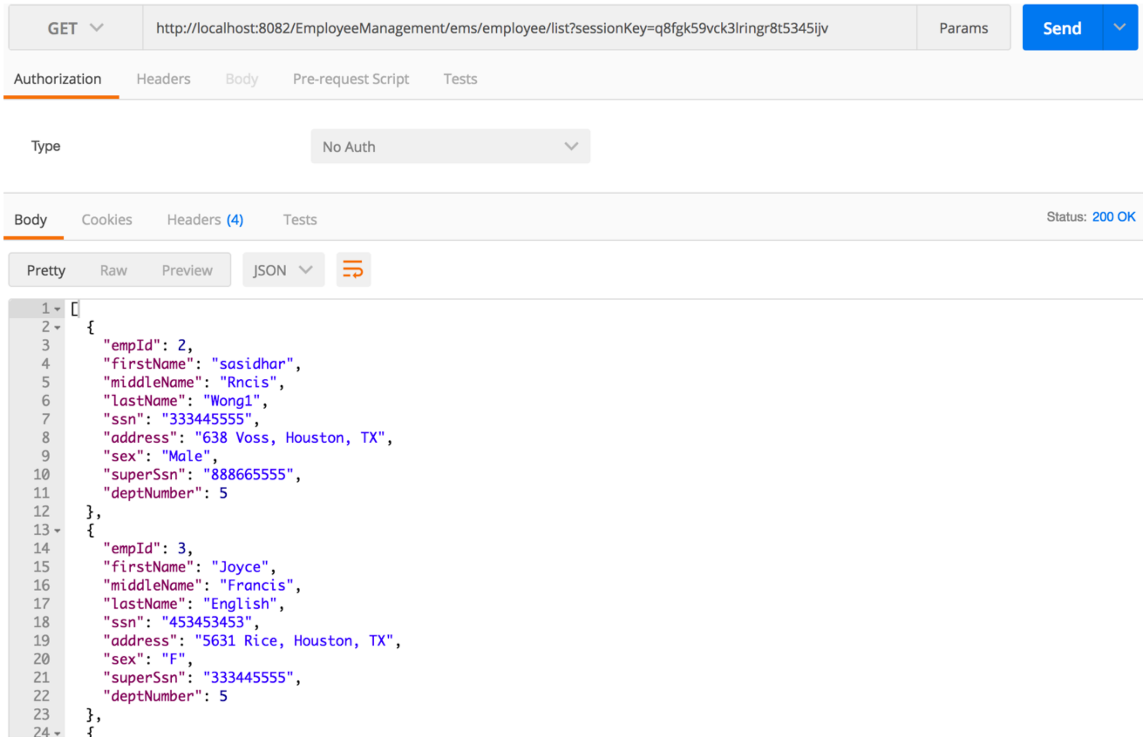
```
{
    "userName": "user",
    "password": "user12"
}
```

Status: 200 OK

JSON

```
{
  "status": "SUCCESS",
  "errorCode": 0,
  "errorMsg": null,
  "session": "q8fgk59vck3lringr8t5345ijv"
}
```

The subsequent requests has have the session-id appended at the end.



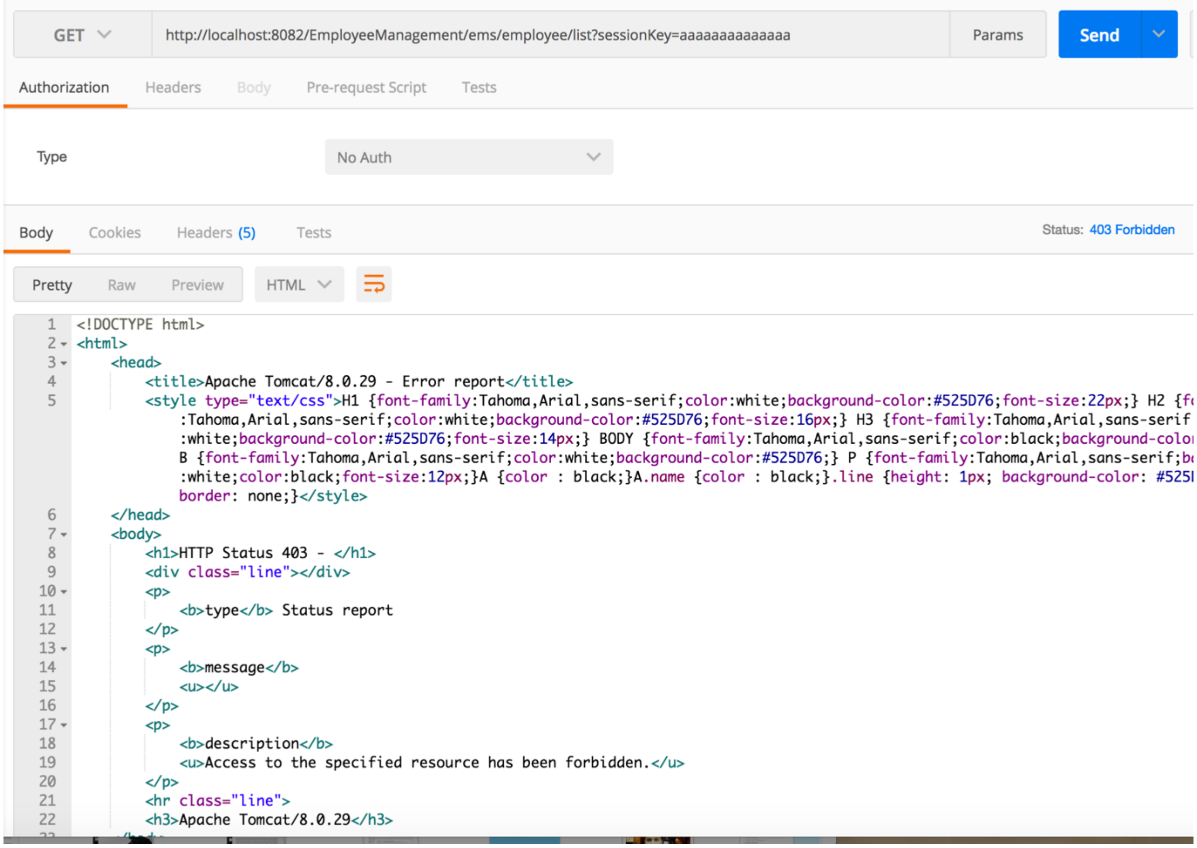
http://localhost:8082/EmployeeManagement/ems/employee/list?sessionKey=q8fgk59vck3lringr8t5345ijv

Type: No Auth

Status: 200 OK

```
[
  {
    "empId": 2,
    "firstName": "sasidhar",
    "middleName": "Rncis",
    "lastName": "Wong1",
    "ssn": "333445555",
    "address": "638 Voss, Houston, TX",
    "sex": "Male",
    "superSsn": "888665555",
    "deptNumber": 5
  },
  {
    "empId": 3,
    "firstName": "Joyce",
    "middleName": "Francis",
    "lastName": "English",
    "ssn": "453453453",
    "address": "5631 Rice, Houston, TX",
    "sex": "F",
    "superSsn": "333445555",
    "deptNumber": 5
  },
  {
```

If we send the api request without session id or some random session id, the access will be denied.



http://localhost:8082/EmployeeManagement/ems/employee/list?sessionKey=aaaaaaaaaaaaa

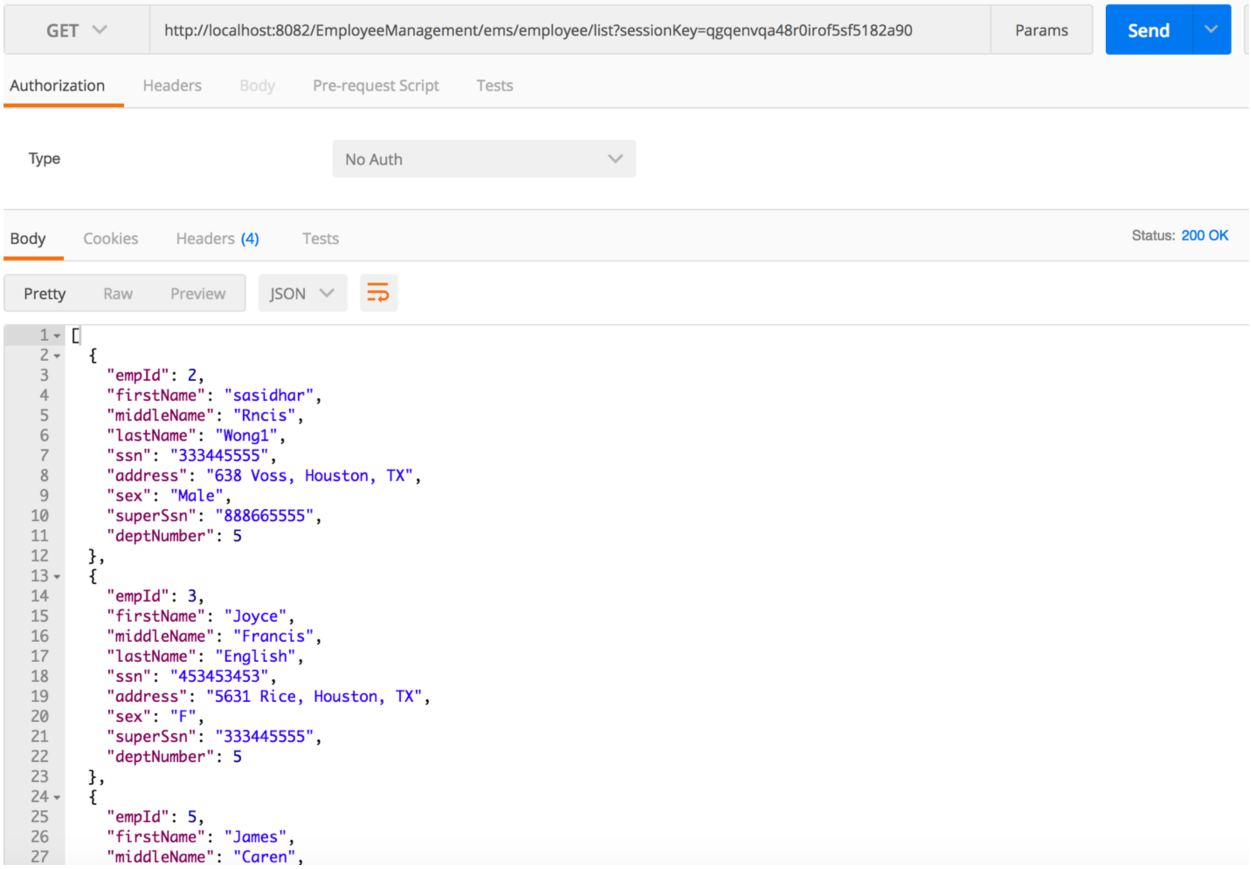
Type: No Auth

Status: 403 Forbidden

```
<!DOCTYPE html>
<html>
    <head>
        <title>Apache Tomcat/8.0.29 - Error report</title>
        <style type="text/css">H1 {font-family:Tahoma,Arial,sans-serif;color:white;background-color:#525D76;font-size:22px;} H2 {f
            :Tahoma,Arial,sans-serif;color:white;background-color:#525D76;font-size:16px;} H3 {font-family:Tahoma,Arial,sans-serif
            :white;background-color:#525D76;font-size:14px;} BODY {font-family:Tahoma,Arial,sans-serif;color:black;background-colo
            B {font-family:Tahoma,Arial,sans-serif;color:white;background-color:#525D76;} P {font-family:Tahoma,Arial,sans-serif;b
            :white;color:black;font-size:12px;}A {color : black;}A.name {color : black;}.line {height: 1px; background-color: #525
            border: none;}</style>
    </head>
    <body>
        <h1>HTTP Status 403 - </h1>
        <div class="line"></div>
        <p>
            <b>type</b> Status report
        </p>
        <p>
            <b>message</b>
            <u></u>
        </p>
        <p>
            <b>description</b>
            <u>Access to the specified resource has been forbidden.</u>
        </p>
        <hr class="line">
        <h3>Apache Tomcat/8.0.29</h3>
```

2. **Caching :** Caching has been used, to minimize the round about time from the API to the backend. Caching has been implemented using Memcache. For this purpose Memcached-Java-Client has been copied into the source from http://mac-dev-env.patrickbougie.com/memcached/ . And then the memcache server has to be started following the instructions from the following given link http://mac-dev-env.patrickbougie.com/memcached/ .

   When we access the api for the first time, since the cache will be free the information will be retrieved from the database and after the first hit the information will be retrieved from the cache.

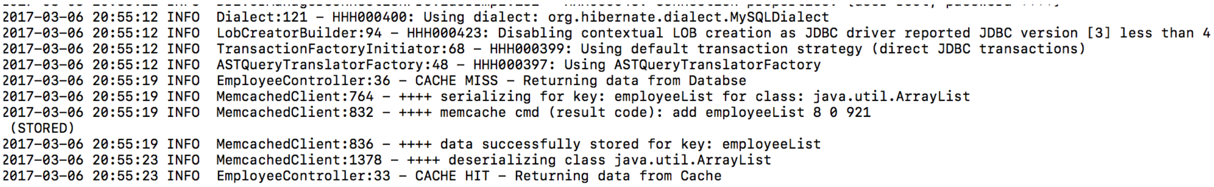


GET http://localhost:8082/EmployeeManagement/ems/employee/list?sessionKey=qgqenvqa48r0irof5sf5182a90

Type: No Auth

Status: 200 OK

```
[
  {
    "empId": 2,
    "firstName": "sasidhar",
    "middleName": "Rncis",
    "lastName": "Wong1",
    "ssn": "333445555",
    "address": "638 Voss, Houston, TX",
    "sex": "Male",
    "superSsn": "888665555",
    "deptNumber": 5
  },
  {
    "empId": 3,
    "firstName": "Joyce",
    "middleName": "Francis",
    "lastName": "English",
    "ssn": "453453453",
    "address": "5631 Rice, Houston, TX",
    "sex": "F",
    "superSsn": "333445555",
    "deptNumber": 5
  },
  {
    "empId": 5,
    "firstName": "James",
    "middleName": "Caren",
```



```
2017-03-06 20:55:12 INFO  Dialect:121 - HHH000400: Using dialect: org.hibernate.dialect.MySQLDialect
2017-03-06 20:55:12 INFO  LobCreatorBuilder:94 - HHH000423: Disabling contextual LOB creation as JDBC driver reported JDBC version [3] less than 4
2017-03-06 20:55:12 INFO  TransactionFactoryInitiator:68 - HHH000399: Using default transaction strategy (direct JDBC transactions)
2017-03-06 20:55:12 INFO  ASTQueryTranslatorFactory:48 - HHH000397: Using ASTQueryTranslatorFactory
2017-03-06 20:55:19 INFO  EmployeeController:36 - CACHE MISS - Returning data from Databse
2017-03-06 20:55:19 INFO  MemcachedClient:764 - ++++ serializing for key: employeeList for class: java.util.ArrayList
2017-03-06 20:55:19 INFO  MemcachedClient:832 - ++++ memcache cmd (result code): add employeeList 8 0 921
 (STORED)
2017-03-06 20:55:19 INFO  MemcachedClient:836 - ++++ data successfully stored for key: employeeList
2017-03-06 20:55:23 INFO  MemcachedClient:1378 - ++++ deserializing class java.util.ArrayList
2017-03-06 20:55:23 INFO  EmployeeController:33 - CACHE HIT - Returning data from Cache
```

3. **Logging :** Logging has been implemented using Log4j api
4. **HTTP Methods :** GET,POST,PUT,PATCH,DELETE HTTP methods have been implemented to perform basic CRUD operations.