Spring boot crud operations example with hibernate

You will require all these

- **spring-boot-starter-web** : It is used for building web layer, including REST APIs, applications using Spring MVC. Uses Tomcat as the default embedded container.
- **spring-boot-starter-data-jpa** : It includes spring data, hibernate, HikariCP, JPA API, **JPA Implementation (default is hibernate)**, JDBC and other required libraries.
- **h2** : Though we can add any database easily using datasource properties in `application.properties` file, we are using h2 database in reduce unnecessacery complexity.
- **spring-boot-starter-test** : It is used to test Spring Boot applications with libraries including JUnit, Hamcrest and Mockito.

When you will generate from https://start.spring.io/

Step 1:

Open application.properties file which you will see under resources folder

```
spring.datasource.url=jdbc:h2:file:C:/temp/test
spring.datasource.driverClassName=org.h2.Driver
spring.datasource.username=sa
spring.datasource.password=
spring.jpa.database-platform=org.hibernate.dialect.H2Dialect

# Enabling H2 Console
spring.h2.console.enabled=true
```

```
# Custom H2 Console URL
spring.h2.console.path=/h2

spring.jpa.hibernate.ddl-auto=none

#Turn Statistics on and log SQL stmts
spring.jpa.show-sql=true
spring.jpa.properties.hibernate.format_sql=true
spring.jpa.properties.hibernate.generate_statistics=false
#logging.level.org.hibernate.type=trace
#logging.level.org.hibernate.stat=debug

logging.pattern.console=%d{yyyy-MM-dd HH:mm:ss} - %msg%n
```

create model package folder inside it

Create EmployeeEntity.java file code :-

```
package com.howtodoinjava.demo.model;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.Table;

@Entity
@Table(name="TBL_EMPLOYEES")
public class EmployeeEntity {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    @Column(name="first_name")
    private String firstName;

    @Column(name="last_name")
    private String lastName;

    @Column(name="email", nullable=false, length=200)
```

```
    private String email;

    public Long getId() {
        return id;
    }

    public void setId(Long id) {
        this.id = id;
    }

    public String getFirstName() {
        return firstName;
    }

    public void setFirstName(String firstName) {
        this.firstName = firstName;
    }

    public String getLastName() {
        return lastName;
    }

    public void setLastName(String lastName) {
        this.lastName = lastName;
    }

    public String getEmail() {
        return email;
    }

    public void setEmail(String email) {
        this.email = email;
    }

    @Override
    public String toString() {
        return "EmployeeEntity [id=" + id + ", firstName=" + firstName +
                ", lastName=" + lastName + ", email=" + email   + "]";
    }
}
```

Step 3:-

Create repository package folder and under it create

EmployeeRepository.java file code

```
package com.howtodoinjava.demo.repository;

import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;

import com.howtodoinjava.demo.model.EmployeeEntity;

@Repository
public interface EmployeeRepository
        extends JpaRepository<EmployeeEntity, Long> {

}
```

Step 4:- create service package folder and under it create

EmployeeService.java file code

```
package com.howtodoinjava.demo.service;

import java.util.ArrayList;
import java.util.List;
import java.util.Optional;

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

import com.howtodoinjava.demo.exception.RecordNotFoundException;
import com.howtodoinjava.demo.model.EmployeeEntity;
import com.howtodoinjava.demo.repository.EmployeeRepository;

@Service
public class EmployeeService {

    @Autowired
    EmployeeRepository repository;

    public List<EmployeeEntity> getAllEmployees()
    {
        List<EmployeeEntity> employeeList = repository.findAll();
```

```
        if(employeeList.size() > 0) {
            return employeeList;
        } else {
            return new ArrayList<EmployeeEntity>();
        }
    }

    public EmployeeEntity getEmployeeById(Long id) throws RecordNotFoundException
    {
        Optional<EmployeeEntity> employee = repository.findById(id);

        if(employee.isPresent()) {
            return employee.get();
        } else {
            throw new RecordNotFoundException("No employee record exist for given
id");
        }
    }

    public EmployeeEntity createOrUpdateEmployee(EmployeeEntity entity) throws
RecordNotFoundException
    {
        Optional<EmployeeEntity> employee = repository.findById(entity.getId());

        if(employee.isPresent())
        {
            EmployeeEntity newEntity = employee.get();
            newEntity.setEmail(entity.getEmail());
            newEntity.setFirstName(entity.getFirstName());
            newEntity.setLastName(entity.getLastName());

            newEntity = repository.save(newEntity);

            return newEntity;
        } else {
            entity = repository.save(entity);

            return entity;
        }
    }

    public void deleteEmployeeById(Long id) throws RecordNotFoundException
    {
        Optional<EmployeeEntity> employee = repository.findById(id);
```

```
        if(employee.isPresent())
        {
            repository.deleteById(id);
        } else {
            throw new RecordNotFoundException("No employee record exist for given
id");
        }
    }
}
```

And create web package folder and write code for

EmployeeController.java:-

```
package com.howtodoinjava.demo.web;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.HttpHeaders;
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.DeleteMapping;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

import com.howtodoinjava.demo.exception.RecordNotFoundException;
import com.howtodoinjava.demo.model.EmployeeEntity;
import com.howtodoinjava.demo.service.EmployeeService;

@RestController
@RequestMapping("/employees")
public class EmployeeController
{
    @Autowired
    EmployeeService service;

    @GetMapping
    public ResponseEntity<List<EmployeeEntity>> getAllEmployees() {
        List<EmployeeEntity> list = service.getAllEmployees();
```

```
        return new ResponseEntity<List<EmployeeEntity>>(list, new HttpHeaders(),
HttpStatus.OK);
    }

    @GetMapping("/{id}")
    public ResponseEntity<EmployeeEntity> getEmployeeById(@PathVariable("id")
Long id)
                                                    throws
RecordNotFoundException {
        EmployeeEntity entity = service.getEmployeeById(id);

        return new ResponseEntity<EmployeeEntity>(entity, new HttpHeaders(),
HttpStatus.OK);
    }

    @PostMapping
    public ResponseEntity<EmployeeEntity> createOrUpdateEmployee(EmployeeEntity
employee)
                                                    throws
RecordNotFoundException {
        EmployeeEntity updated = service.createOrUpdateEmployee(employee);
        return new ResponseEntity<EmployeeEntity>(updated, new HttpHeaders(),
HttpStatus.OK);
    }

    @DeleteMapping("/{id}")
    public HttpStatus deleteEmployeeById(@PathVariable("id") Long id)
                                                    throws
RecordNotFoundException {
        service.deleteEmployeeById(id);
        return HttpStatus.FORBIDDEN;
    }

}
```

Your main Application file code:-

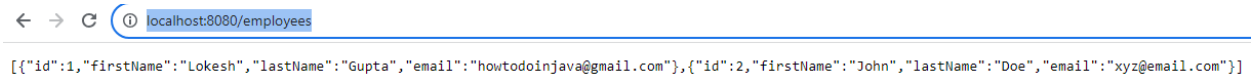
DemoApplication.java file code:-

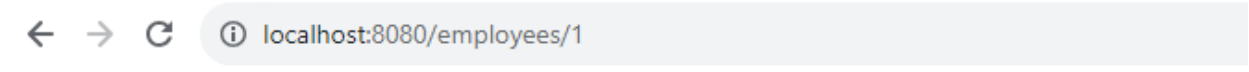
```
package com.howtodoinjava.demo;
```

```java
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class DemoApplication {

    public static void main(String[] args) {
        SpringApplication.run(DemoApplication.class, args);
    }

}
```

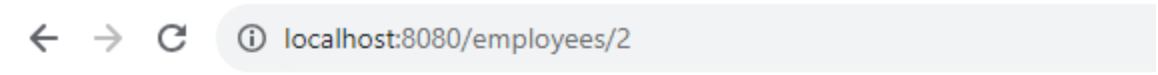
Output:-

http://localhost:8080/employees





{"id":1,"firstName":"Lokesh","lastName":"Gupta","email":"howtodoinjava@gmail.com"}



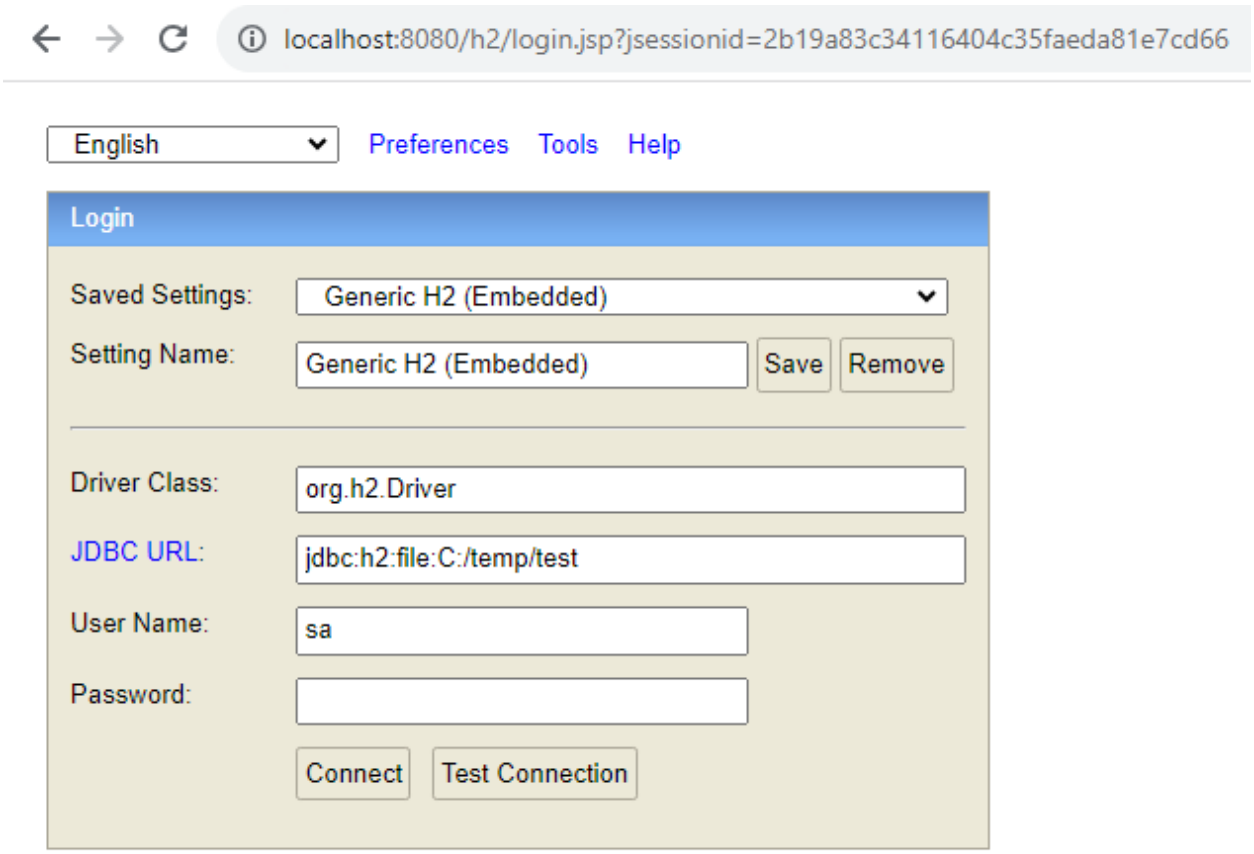
{"id":2,"firstName":"John","lastName":"Doe","email":"xyz@email.com"}

You can check database from

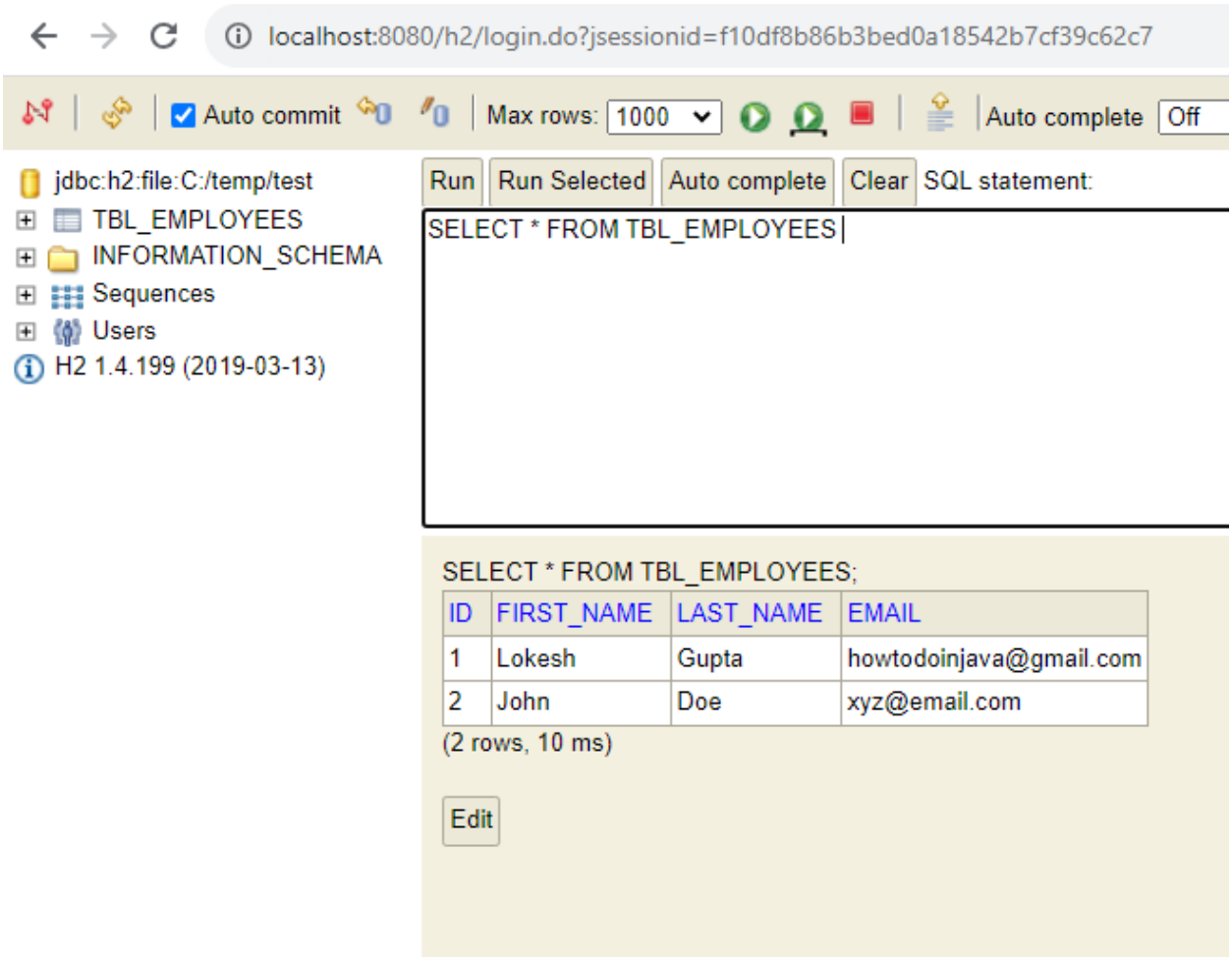
Open your browser and type

Localhost:8080/h2

You will see



Click on connect and you will see your database .


localhost:8080/h2/login.do?jsessionid=f10df8b86b3bed0a18542b7cf39c62c7
Auto commit
Max rows: 1000
Auto complete Off
jdbc:h2:file:C:/temp/test
TBL_EMPLOYEES
INFORMATION_SCHEMA
Sequences
Users
H2 1.4.199 (2019-03-13)
Run
Run Selected
Auto complete
Clear
SQL statement:
SELECT * FROM TBL_EMPLOYEES
SELECT * FROM TBL_EMPLOYEES;
ID FIRST_NAME LAST_NAME EMAIL
1 Lokesh Gupta howtodoinjava@gmail.com
2 John Doe xyz@email.com
(2 rows, 10 ms)
Edit