

Spring Data JPA

(MVC, @NamedQuery, @Query, 메소드 이름을 통한 쿼리작성 maria DB 실습)

오라클자바커뮤니티

(ojc.asia, ojcedu.com)

Spring Data JPA

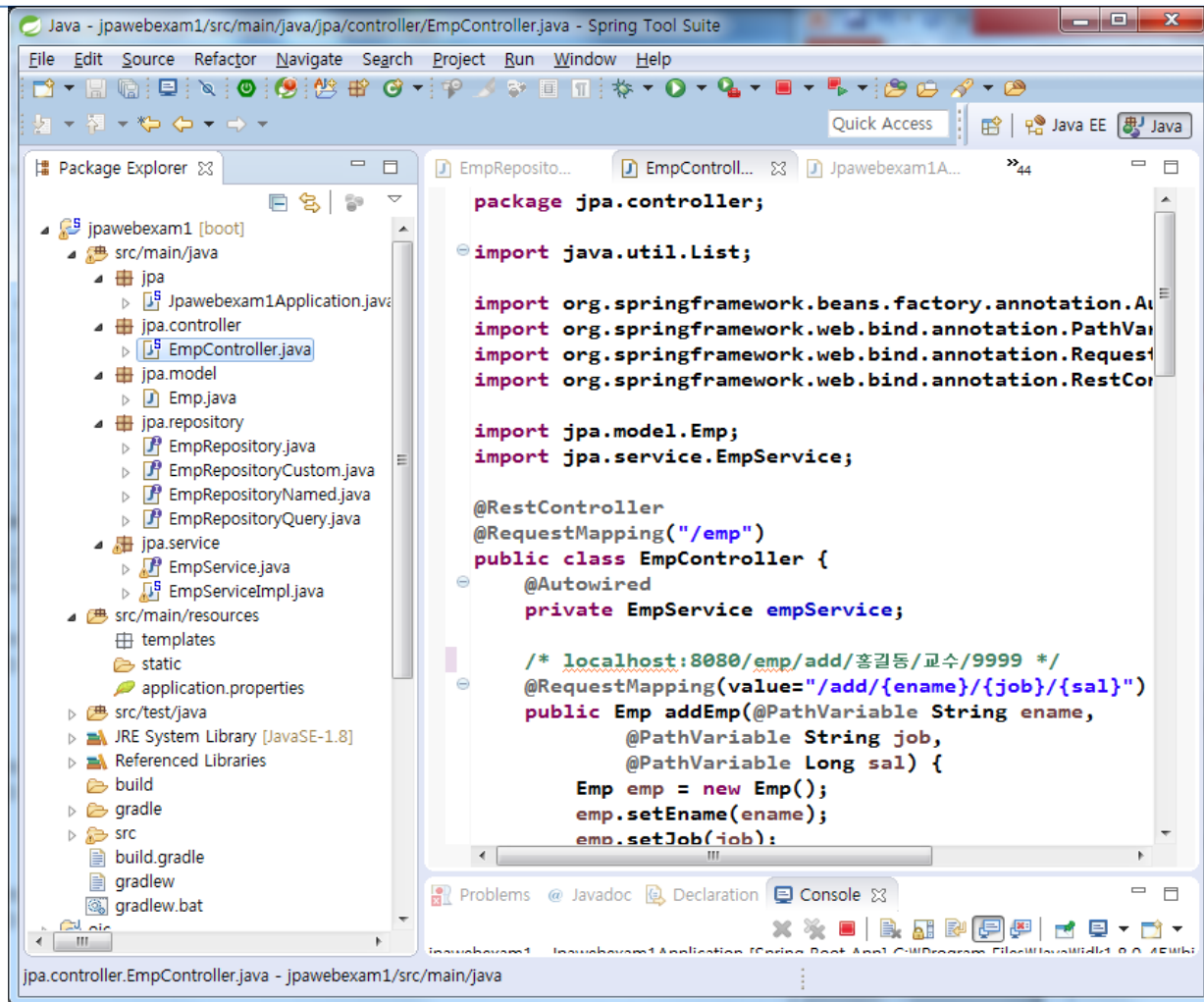
Spring Data JPA 실습 - 1

- 지금까지 학습한 Spring Data JPA, Spring WEB MVC, @NamedQuery, @Query, 메소드 이름을 통해 쿼리를 정의하는 예제를 마리아DB를 이용하여 작성해 보자.

STS -> Spring Starter Project

project name :
jpawebexam1
Type : Gradle(STS)
package : jpa

Data -> JPA
DataBase -> MySQL
Wen -> Web



Spring Data JPA

Spring Data JPA 실습 - 2

[Jpawebexam1Application.java]

```
package jpa;

import
org.springframework.boot.SpringApplication;im
port
org.springframework.boot.autoconfigure.Spring
BootApplication;@SpringBootApplication

public class Jpawebexam1Application {
public static void main(String[] args) {

        SpringApplication.run(Jpawebexam1
Application.class, args);
    }
}
```

[application.properties]

```
spring.datasource.platform=mysql
spring.datasource.sql-script-encoding=UTF-8
spring.datasource.url=jdbc:mysql://localhost/jpa
webexam1?createDatabaseIfNotExist=true
spring.datasource.username=root
spring.datasource.password=1111
spring.datasource.driver-class-
name=com.mysql.jdbc.Driver
spring.jpa.show-sql=true
spring.jpa.hibernate.ddl-auto=update
logging.level.jpa=DEBUG
```

Spring Data JPA 실습 - 3

[Emp.java]

```
package jpa.model;
```

```
@Entity @NamedQuery(name="Emp.findBySal",  
    query="select ename, job, sal from  
    Emp e where e.sal > ?1")
```

```
public class Emp implements Serializable {  
    private static final long  
    serialVersionUID = 1L;  
    @Id @GeneratedValue  
    private Long empno;  
    private String ename;  
    private String job;  
    private Long sal;  
  
    public Long getEmpno() {  
        return empno;  
    }  
    public void setEmpno(Long empno)  
{  
        this.empno = empno;  
    }  
    public String getEname() {
```

```
        return ename;  
    }  
    public void setName(String ename) {  
        this.ename = ename;  
    }  
    public String getJob() {  
        return job;  
    }  
    public void setJob(String job) {  
        this.job = job;  
    }  
    public Long getSal() {  
        return sal;  
    }  
    public void setSal(Long sal) {  
        this.sal = sal;  
    }  
}
```

Spring Data JPA 실습 - 4

[EmpErpository.java]

```
package jpa.repository;

public interface
EmpRepository extends
JpaRepository<Emp, Long>{
}
```

[EmpRepositoryCustom.java]

```
package jpa.repository;

public interface EmpRepositoryCustom extends
Repository<Emp,
Long> {
    @Query(value = "select e from Emp e where e.sal = :sal")
    List<Emp> findBySalCustom(@Param("sal") Long sal);

    /* 아래는 메소드 이름으로 쿼리를 생성한다 */
    List<Emp>    findFirst3BySalBetweenOrderByEnameDesc
                  (Long sal1, Long Sal2);
}
```

Spring Data JPA 실습 - 5

[EmpRepositoryNamed.java]

```
package jpa.repository;
```

```
public interface EmpRepositoryNamed  
    extends Repository<Emp, Long> {  
    //Emp엔티티에서 정의된 NameQuery지  
    정  
    List<Emp> findBySal(Long sal);  
}
```

[EmpRepositoryQuery.java]

```
package jpa.repository;
```

```
public interface EmpRepositoryQuery  
    extends Repository<Emp, Long> {  
    @Query(value="select * from  
    #{#entityName} e where e.ename = ?1",  
            nativeQuery=true)  
    List<Emp> findByEname(String ename);
```

```
    @Query(value="select ename, job, sal  
    from Emp e where e.sal > ?1 and e.sal < ?2  
    ")
```

```
    List<Emp> findBySalRange(Long sal1,  
    Long sal2);
```

```
    @Query(value="select ename, job, sal  
    from Emp e where e.ename  
    like %:ename% ")
```

```
    List<Emp>  
    findByEnameMatch(@Param("ename")  
    String ename);
```

//@Param : 쿼리파라미터로 넣을 려고...

```
@Query(value="select ename, job, sal from  
    Emp e where e.ename = :ename and  
    e.job=:job and e.sal = :sal")
```

```
    List<Emp>  
    findByNamedParam(@Param("ename")  
    String ename, @Param("job") String job,  
    @Param("sal") Long sal);
```

```
}
```

Spring Data JPA 실습 - 6

[EmpService.java]

```
package jpa.service;
```

```
public interface EmpService {  
    List<Emp> findAll();  
    void saveEmp(Emp emp);  
    Emp findOne(Long empno);  
    void delete(Long empno);  
    List<Emp> findByName(String ename);  
    List<Emp> findBySalCustom(Long sal);  
    List<Emp> findBySal(Long sal);  
    List<Emp> findBySalRange(Long sal1,  
Long sal2);  
    List<Emp> findByNameMatch(String  
ename);  
    List<Emp> findByNameParam(String  
ename, String job, Long sal);  
    List<Emp>  
findFirst3BySalBetweenOrderByEnameDesc(  
Long sal1, Long sal2);  
}
```

[EmpServiceImpl.java] – 다음 페이지 계속

```
package jpa.service;
```

```
@Service("empService")
```

```
public class EmpServiceImpl implements  
EmpService {
```

```
@Autowired private EmpRepository  
empRepository;
```

```
@Autowired private EmpRepositoryCustom  
empRepositoryCustom;;
```

```
@Autowired private EmpRepositoryQuery  
empRepositoryQuery;
```

```
@Autowired private EmpRepositoryNamed  
empRepositoryNamed;
```

```
public List<Emp> findAll() {  
    return empRepository.findAll();  
}
```

```
public void saveEmp(Emp emp) {  
    empRepository.save(emp);  
}
```

Spring Data JPA

Spring Data JPA 실습 - 7

```
public Emp findOne(Long empno) {
    return empRepository.findOne(empno);
}

public void delete(Long empno) {
    empRepository.delete(empno);
}

public List<Emp> findByEname(String
    ename) { return
    empRepositoryQuery.findByEname(ename
    );
}

public List<Emp> findBySal(Long sal) {
    return
    empRepositoryNamed.findBySal(sal);
}

public List<Emp> findBySalRange(Long
    sal1, Long sal2) { return
    empRepositoryQuery.findBySalRange(sal1,
    sal2);
}
```

```
public List<Emp> findByEnameMatch(String
    ename) {
    return
    empRepositoryQuery.findByEnameMatch(en
    ame); }

public List<Emp>
    findByNamedParam(String ename, String
    job, Long sal) {
    return
    empRepositoryQuery.findByNamedParam(e
    name, job, sal); }

public List<Emp> findBySalCustom(Long
    sal) {
    return
    empRepositoryCustom.findBySalCustom(sal
    );
}

public List<Emp>
    findFirst3BySalBetweenOrderByEnameDesc
    (Long sal1, Long sal2) {
    return
    empRepositoryCustom.findFirst3BySalBetw
    eenOrderByEnameDesc(sal1, sal2);
}
```

Spring Data JPA 실습 - 8

[EmpController.java]

```
package jpa.controller;
@RestController
@RequestMapping("/emp")
public class EmpController {
    @Autowired
    private EmpService empService;

    /* localhost:8080/emp/add/홍길동/교수/9999 */
    @RequestMapping(value="/add/{ename}/{job}/{sal}")
    public Emp addEmp(@PathVariable String ename,
                     @PathVariable String job,
                     @PathVariable Long sal) {
        Emp emp = new Emp();
        emp.setEname(ename);
        emp.setJob(job);
        emp.setSal(sal);
        empService.saveEmp(emp);

        return emp;
    }
}
```

Spring Data JPA 실습 - 9

```
/* localhost:8080/emp/delete/9999 */  
@RequestMapping(value="/delete/{empno}")  
public void deleteEmp(@PathVariable Long empno) {  
    empService.delete(empno);  
}
```

```
/* localhost:8080/emp/ */  
@RequestMapping(value="/")  
public List<Emp> list() {  
    return empService.findAll();  
}
```

```
/* localhost:8080/emp/1 */  
@RequestMapping(value="/{empno}")  
public Emp getEmp(@PathVariable Long empno) {  
    return empService.findOne(empno);  
}
```

Spring Data JPA 실습 - 10

```
/* localhost:8080/emp/search/ename/홍길동 */
```

```
@RequestMapping(value="/search/ename/{ename}")  
public List<Emp> getEmpByName(@PathVariable String ename) {  
    return empService.findByEname(ename);  
}
```

```
/* localhost:8080/emp/search/custom/sal/9999 */
```

```
@RequestMapping(value="/search/custom/sal/{sal}")  
public List<Emp> getEmpBySalCustom(@PathVariable Long sal) {  
    return empService.findBySalCustom(sal);  
}
```

```
@RequestMapping(value="/search/custom/top3/{sal1}/{sal2}")  
public List<Emp> getEmpByNameCustom(@PathVariable Long sal1,
```

```
@PathVariable Long sal2) {  
    return empService.findFirst3BySalBetweenOrderByEnameDesc(sal1,  
sal2);  
}
```

Spring Data JPA 실습 - 11

```
@RequestMapping(value="/search/match/ename/{ename}")
public List<Emp> getEmpByEnameMatch(@PathVariable String ename) {
    return empService.findByEnameMatch(ename);
}
/* localhost:8080/emp/search/param/홍길동/교수/9999 */
@RequestMapping(value="/search/param/{ename}/{job}/{sal}")
public List<Emp> getEmpByNamedParam(@PathVariable String ename,
    @PathVariable String job,
    @PathVariable Long sal) {
    return empService.findByNameParam(ename, job, sal);
}
@RequestMapping(value="/search/sal/{sal}")
public List<Emp> getEmpBySal(@PathVariable Long sal) {
    return empService.findBySal(sal);
}
/* localhost:8080/emp/search/sal/5555/9999 */
@RequestMapping(value="/search/sal/{sal1}/{sal2}")
public List<Emp> getEmpBySal(@PathVariable Long sal1,
    @PathVariable Long sal2) {
    return empService.findBySalRange(sal1, sal2);
}
}
```