

스프링부트,JPA,그래들 HelloWorld(Spring Boot, Gradle)

간단히 Spring Boot, Gradle, JPAmf 이용하여 HelloWorld 형태의 심플 예제를 만들어 보자, 원래 HelloWorld 는 프로그래밍을 처음 공부할 때 간단하지만 해당 언어의 특색을 느껴보는 프로그램이니 너무 복잡하지 않게 간단히 작성하고 이해해 보자.

Spring Boot 는 간단히 스프링 프로젝트를 구축해주는 것이며 JPA 는 JPA 는 EJB 3.0 스펙 작업에서 기존 EJB 의 ORM 인 엔티티빈(Entity Bean)을 JPA 로 변경하고 자바에서 영속성 관리를위한 표준 API 이다. JPA 는 Hibernate, OpenJPA, EclipseLink, TopLink Essentials 과 같은 구현체가 있고 표준 인터페이스가 바로 JPA 인 것이다.

간단히 예제를 작성해보자.

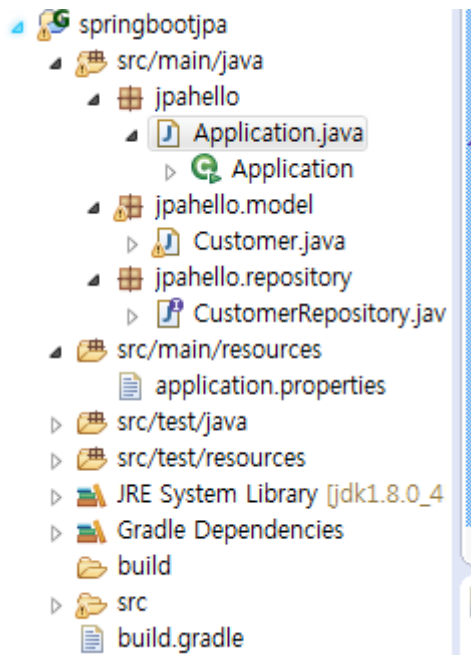
0. 그래들 설치 및 이클립스 플러그인 설치는 아래 경로에서 확인하자.

http://ojc.asia/bbs/board.php?bo_table=LecGradle&wr_id=1

http://ojc.asia/bbs/board.php?bo_table=LecGradle&wr_id=8

1. ECLipse 에서 New -> Project -> Gradle -> Gradle 프로젝트를 생성하고 프로젝트명을 springbootjpa 라고 주자.

전체 프로젝트 구조는 다음과 같다.



2. build.gradle

//아래 부트스크립트는 Spring Boot Gradle Plugin을 위한 것이다.

//그런다음 apply plugin: 'spring-boot' 하면 된다.

```
buildscript {  
    repositories {  
        mavenCentral()  
    }  
    dependencies {  
        classpath("org.springframework.boot:spring-boot-gradle-  
plugin:1.2.6.RELEASE")  
    }  
}  
  
apply plugin: 'java'  
apply plugin: 'spring-boot'  
  
jar {  
    baseName = 'ipa-exam'  
    version = '0.1.0'  
}  
  
repositories {  
    mavenCentral()  
    mavenLocal()  
}
```

```

sourceCompatibility = 1.8
targetCompatibility = 1.8

dependencies {
    compile("org.springframework.boot:spring-boot-starter-data-jpa")
    compile("com.oracle:ojdbc6:11.2.0.4")
    testCompile("junit:junit")
}

task wrapper(type: Wrapper) {
    gradleVersion = '2.4'
}

```

3. application.properties

```

spring.datasource.url=jdbc:oracle:thin:@192.168.0.27:1521:onj
spring.datasource.driverClassName=oracle.jdbc.driver.OracleDriver
spring.datasource.username=scott
spring.datasource.password=tiger
//always create table on start project
spring.jpa.hibernate.ddl-auto=create
spring.jpa.database=oracle
spring.jpa.show-sql=true

```

4. Customer.java

```

package jpahello.model;

import java.io.Serializable;

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

@Entity
public class Customer implements Serializable {

    @Id //id 항목이 테이블 PK

```

```
@GeneratedValue(strategy = GenerationType.SEQUENCE, generator =
"id_Sequence") //id 항목이 시퀀스를 이용하여 자동생성됨
@SequenceGenerator(name = "id_Sequence", sequenceName = "ID_SEQ")
private Integer id; //오라클 시퀀스를 자동생성

@Column(name = "name", nullable = false)
private String name;

@Column(name = "age", nullable = false)
private int age;

public Customer() {}

public Customer(String name, int age) {
    this.name = name;
    this.age = age;
}

    public Integer getId() {
        return id;
    }

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

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public int getAge() {
        return age;
    }

    public void setAge(int age) {
        this.age = age;
    }
}
```

5. CustomerRepository.java

```
package jpahello.repository;

import org.springframework.data.repository.CrudRepository;
import jpahello.model.Customer;

/*
 * Custpmer Entity 를 이용하기 위한 Repository 인터페이스인데
 * save, delete, findAll, findOne 과 같은 기본적인 메소드 외에도 추가적인
 * 메소드를 만들 수도 있다.
 * 현재는 그대로 기본 기능만 사용하므로 빈 상태로 두자.
 */

public interface CustomerRepository extends CrudRepository<Customer, Long>
{

}
```

6. Application.java

```
package jpahello;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.CommandLineRunner;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

import jpahello.model.Customer;
import jpahello.repository.CustomerRepository;

@SpringBootApplication
public class Application implements CommandLineRunner{
    public static void main(String[] args) {
        SpringApplication.run(Application.class, args);
    }

    @Autowired
    CustomerRepository customerRepository;
```



```
complete
2015-09-29 10:27:58.138 INFO 5140 --- [           main]
o.s.j.e.a.AnnotationMBeanExporter      : Registering beans for JMX
exposure on startup
Hibernate: select ID_SEQ.nextval from dual
Hibernate: insert into customer (age, name, id) values (?, ?, ?)
Hibernate: insert into customer (age, name, id) values (?, ?, ?)
Hibernate: insert into customer (age, name, id) values (?, ?, ?)
Hibernate: select customer0_.id as id1_0_, customer0_.age as
age2_0_, customer0_.name as name3_0_ from customer customer0_
findAll() Method.
50:일길동:10
51:이길동:20
52:사길동:40
.....
.....
.....
```

실무프로그래머 전문교육

(오라클,SQL,자바,스프링프레임워크,닷넷,안드로이드,웹퍼블리싱)

오라클자바커뮤니티교육센터(100%환급, 개인부담 0~20%) <http://ojcedu.com>