

# 学习Spring boot

在学习 Spring 的时候，可能采用的是 eclipse 或者 IntelliJ IDEA。其中 IDEA 分为社区版和旗舰版。更多的情况使用的是社区版进行编码学习。

在 IntelliJ 的官网可以发现，社区版没有集成 Spring 全家桶，如果想要手动创建一个 Spring 项目，过程非常麻烦。好在可以在 Spring 的网站上面创建<http://start.spring.io>中创建。

1. 选择依赖管理工具，一般采用 Maven

2. 填写项目名称和包名

3. 选择 Spring 版本

4. 添加依赖项。

4.1 Spring Boot DevTools

4.2 Lombok

4.3 Spring web

4.4 JDBC API

4.5 Spring Data JDBC

4.6 Spring Data JPA

4.7 MySQL Driver

5. 导出项目

以上就是我们用 Spring Boot 进行创建Web项目通常引用的依赖包。

在采用 IDEA 打开第五步导出的项目，修改资源下面的 application.property 的名称为 application.yml

```
service:
  port: 8080
spring:
  datasource:
    driver-class-name: com.mysql.cj.jdbc.Driver
    url: jdbc:mysql://localhost:3306/example?
useUnicode=true&characterEncoding=utf-8
    username: root
    password: root
  jap:
    database-platform: org.hibernate.dialect.MySQL5InnoDBDialect
    show-sql: true
    hibernate:
```

## 实体映射类

```
@MappedSuperclass
public class BaseEntity{
    public int getVersion() {
        return version;
    }

    public void setVersion(int version) {
        this.version = version;
    }

    public int getCreatedBy() {
        return createdBy;
    }
}
```

```

    }

    public void setCreatedBy(int createdBy) {
        this.createdby = createdBy;
    }

    public Date getCreateddate() {
        return createddate;
    }

    public void setCreateddate(Date createddate) {
        this.createddate = createddate;
    }

    public int getLastupdatedby() {
        return lastupdatedby;
    }

    public void setLastupdatedby(int lastupdatedby) {
        this.lastupdatedby = lastupdatedby;
    }

    public Date getLastupdateddate() {
        return lastupdateddate;
    }

    public void setLastupdateddate(Date lastupdateddate) {
        this.lastupdateddate = lastupdateddate;
    }

    public String getTransaction() {
        return transaction;
    }

    public void setTransaction(String transaction) {
        this.transaction = transaction;
    }

    private int version;
    private int createdBy;
    private Date createddate;
    private int lastupdatedby;
    private Date lastupdateddate;
    private String transaction;
}

@Entity
public class UserEntity extends BaseEntity {
    @Id
    private int id;
    private String name;
    private int tid;

    public int getId() {
        return id;
    }

    public void setId(int id) {

```

```

        this.id = id;
    }

    public String getName() {
        return name;
    }

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

    public int getTid() {
        return tid;
    }

    public void setTid(int tid) {
        this.tid = tid;
    }
}

```

## Dao

```

// 使用JPA链接, 自己定义的接口需要继承JpaRepository<映射实体类, 主键类型>
public interface UserRepository extends JpaRepository<UserEntity,Integer> {

    @Query(value = "SELECT * FROM user", nativeQuery = true)
    List<UserEntity> fetchAll();
}

```

## Controller

```

@RestController
public class UserController {
    @Resource
    private UserRepository userRepository;

    @Resource
    private LabRepository labRepository;

    @RequestMapping("/fetch")
    public List<UserEntity> fetchAll(){
        return userRepository.fetchAll();
    }

    @RequestMapping("/labs")
    public List<LabEntity> getAllLabs(){
        return labRepository.fetchAll();
    }
}

```

定义的Controller, 需要标记为 `@RestController`, 在方法上方标记 `@RequestMapping` 来定义 REST 格式。

## 运行

在Maven 管理工具中，点开Plugin,找到 Spring-Boot:run.  
可以在Run Configuration中添加Maven配置。

