

## Spring Boot 数据校验

Spring Boot 数据校验依赖于 Hibernate-validator 来完成，使用 Hibernate-validator 进行数据校验，需要定义一个接收校验结果的数据模型，使用注解的形式描述字段的校验规则。

## Spring Boot 整合 JDBC

- pom.xml

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
</dependency>

<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-jdbc</artifactId>
</dependency>

<dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <version>8.0.11</version>
</dependency>
```

- application.yml

```
spring:
  datasource:
    url: jdbc:mysql://localhost:3306/mybatis?
    useUnicode=true&characterEncoding=UTF-8
    username: root
    password: root
    driver-class-name: com.mysql.cj.jdbc.Driver
```

- User

```
package com.southwind.entity;

import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;

@Data
@AllArgsConstructor
@NoArgsConstructor
```

```
public class User {
    private long id;
    private String username;
    private String password;
    private int age;
}
```

- UserRepository

```
package com.southwind.repository;

import com.southwind.entity.User;

import java.util.List;

public interface UserRepository {
    public List<User> findAll();
    public User findById(long id);
    public void save(User user);
    public void update(User user);
    public void deleteById(long id);
}
```

- UserRepositoryImpl

```
package com.southwind.repository.impl;

import com.southwind.entity.User;
import com.southwind.repository.UserRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.jdbc.core.BeanPropertyRowMapper;
import org.springframework.jdbc.core.JdbcTemplate;
import org.springframework.stereotype.Repository;

import java.util.List;

@Repository
public class UserRepositoryImpl implements UserRepository {

    @Autowired
    private JdbcTemplate jdbcTemplate;

    @Override
    public List<User> findAll() {
        return jdbcTemplate.query("select * from t_user", new
        BeanPropertyRowMapper<>(User.class));
    }
}
```

```

@Override
public User findById(long id) {
//      return jdbcTemplate.query("select * from t_user where id = ?",new
BeanPropertyRowMapper<>(User.class),id).get(0);
      return jdbcTemplate.queryForObject("select * from t_user where id =
?",new Object[] {id},new BeanPropertyRowMapper<>(User.class));
}

@Override
public void save(User user) {
    jdbcTemplate.update("insert into t_user(username,password,age)
values(?,?,?)",user.getUsername(),user.getPassword(),user.getAge());
}

@Override
public void update(User user) {
    jdbcTemplate.update("update t_user set username = ?,password = ?,age
= ? where id =
?",user.getUsername(),user.getPassword(),user.getAge(),user.getId());
}

@Override
public void deleteById(long id) {
    jdbcTemplate.update("delete from t_user where id = ?",id);
}
}

```

- UserHandler

```

package com.southwind.controller;

import com.southwind.entity.User;
import com.southwind.repository.UserRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;

import java.util.List;

@RestController
@RequestMapping("/user")
public class UserHandler {

    @Autowired
    private UserRepository userRepository;

    @GetMapping("/findAll")
    public List<User> findAll(){
        return userRepository.findAll();
    }
}

```

```

@GetMapping("/findById/{id}")
public User findById(@PathVariable("id") long id){
    return userRepository.findById(id);
}

@PostMapping("/save")
public void save(@RequestBody User user){
    userRepository.save(user);
}

@PutMapping("/update")
public void update(@RequestBody User user){
    userRepository.update(user);
}

@DeleteMapping("/deleteById/{id}")
public void deleteById(@PathVariable("id") long id){
    userRepository.deleteById(id);
}
}

```

## Spring Boot 整合 MyBatis

- pom.xml

```

<dependencies>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-web</artifactId>
    </dependency>

    <!-- 整合 MyBatis -->
    <dependency>
        <groupId>org.mybatis.spring.boot</groupId>
        <artifactId>mybatis-spring-boot-starter</artifactId>
        <version>1.3.1</version>
    </dependency>

    <dependency>
        <groupId>mysql</groupId>
        <artifactId>mysql-connector-java</artifactId>
        <version>8.0.11</version>
    </dependency>

    <dependency>
        <groupId>org.projectlombok</groupId>

```

```

        <artifactId>lombok</artifactId>
        <version>1.18.6</version>
        <scope>provided</scope>
    </dependency>
</dependencies>

```

- application.yml

```

spring:
  datasource:
    url: jdbc:mysql://localhost:3306/mybatis?
    useUnicode=true&characterEncoding=UTF-8
    username: root
    password: root
    driver-class-name: com.mysql.cj.jdbc.Driver
  server:
    port: 9090

mybatis:
  mapper-locations: classpath:mapping/*.xml
  type-aliases-package: com.southwind.entity

```

- ClazzRepository

```

package com.southwind.repository;

import com.southwind.entity.Clazz;

import java.util.List;

public interface ClazzRepository {
    public List<Clazz> findAll();
    public Clazz findById(long id);
    public void save(Clazz clazz);
    public void update(Clazz clazz);
    public void deleteById(long id);
}

```

- ClazzRepository.xml

```

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
"http://mybatis.org/dtd/mybatis-3-mapper.dtd">
<mapper namespace="com.southwind.repository.ClazzRepository">

    <select id="findAll" resultType="Clazz">
        select * from classes
    </select>

```

```

<select id="findById" parameterType="long" resultType="Clazz">
    select * from classes where id = #{id}
</select>

<insert id="save" parameterType="Clazz">
    insert into classes(name) values(#{name})
</insert>

<update id="update" parameterType="Clazz">
    update classes set name = #{name} where id = #{id}
</update>

<delete id="deleteById" parameterType="long">
    delete from classes where id = #{id}
</delete>

</mapper>

```

- ClazzHandler

```

package com.southwind.controller;

import com.southwind.entity.Clazz;
import com.southwind.repository.ClazzRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;

import java.util.List;

@RestController
@RequestMapping("/clazz")
public class ClazzHandler {

    @Autowired
    private ClazzRepository clazzRepository;

    @GetMapping("/findAll")
    public List<Clazz> findAll(){
        return clazzRepository.findAll();
    }

    @GetMapping("/findById/{id}")
    public Clazz findById(@PathVariable("id") long id){
        return clazzRepository.findById(id);
    }

    @PostMapping("/save")
    public void save(@RequestBody Clazz clazz){

```

```

        clazzRepository.save(clazz);
    }

    @PutMapping("/update")
    public void update(@RequestBody Clazz clazz){
        clazzRepository.update(clazz);
    }

    @DeleteMapping("/deleteById/{id}")
    public void deleteById(@PathVariable long id){
        clazzRepository.deleteById(id);
    }
}

```

- 启动类

```

package com.southwind;

import org.mybatis.spring.annotation.MapperScan;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
@MapperScan("com.southwind.repository")
public class HelloApplication {
    public static void main(String[] args) {
        SpringApplication.run(HelloApplication.class, args);
    }
}

```

注解说明：

- `@SpringBootApplication`：声明该类是 Spring Boot 的入口。
- `@MapperScan`：配合 MyBatis 框架使用，将 Mapper 代理扫描到 IoC 中，注解中的值对应接口所在的包。