

面试官：如何在 Spring Boot 中进行参数校验？

面试官：如何在 Spring Boot 中进行参数校验？

Java面试那些事儿



Java 面试那些事儿

一个能带你进入BATM大厂的技术号，这里不仅有面试题、内推渠道，还有Java技术...

172 篇原创内容 2164 位朋友关注



长按扫一扫 关注我们 带你进入大厂

作者：狂乱的贵公子

来源：cnblogs.com/cjsblog/p/8946768.html

开发过程中，后台的参数校验是必不可少的，所以经常会看到类似下面这样的代码

```
public BaseResult save(CouponEntity coupon) {  
    if (StringUtils.isBlank(coupon.getCouponName())) {  
        return BaseResult.failure("优惠券名称不能为空");  
    }  
    if (null == coupon.getCouponType()) {  
        return BaseResult.failure("优惠券类型不能为空");  
    }  
    if (coupon.getCouponType() == CouponTypeEnum.ZHE_KOU.getType() && null == coupon.getDiscountRate()) {  
        return BaseResult.failure("折扣率不能为空");  
    }  
    if (coupon.getCouponType() != CouponTypeEnum.ZHE_KOU.getType() && null == coupon.getCouponAmount()) {  
        return BaseResult.failure("优惠券面值不能为空");  
    }  
    if (null == coupon.getCouponNum()) {  
        return BaseResult.failure("发放总量不能为空");  
    }  
    if (null == coupon.getReleaseStartTime() || null == coupon.getReleaseEndTime()) {  
        return BaseResult.failure("发放起止时间不能为空");  
    }  
    if (coupon.getReleaseStartTime().compareTo(coupon.getReleaseEndTime()) > 0) {  
        return BaseResult.failure("发放开始时间不能大于发放结束时间");  
    }  
    if (null == coupon.getAvailableType()) {  
        return BaseResult.failure("有效期不能为空");  
    }  
    if (coupon.getAvailableType() == CouponValidityTypeEnum.FIXED_DATE.getType()) {  
        if (null == coupon.getAvailableStartTime() || null == coupon.getAvailableEndTime()) {  
            return BaseResult.failure("有效日期不能为空");  
        }  
        if (coupon.getAvailableStartTime().compareTo(coupon.getAvailableEndTime()) > 0) {  
            return BaseResult.failure("有效日期开始时间不能大于结束时间");  
        }  
        if (coupon.getAvailableStartTime().compareTo(coupon.getReleaseStartTime()) < 0) {  
            return BaseResult.failure("有效日期开始时间不能小于发放开始时间");  
        }  
        if (coupon.getAvailableEndTime().compareTo(coupon.getReleaseEndTime()) < 0) {  
            return BaseResult.failure("发放结束时间不能大于有效期结束时间");  
        }  
    }  
    if (coupon.getAvailableType() == CouponValidityTypeEnum.GET_DATE.getType() && null == coupon.getAvailableDay()) {  
        return BaseResult.failure("领取之日不能为空");  
    }  
    if (null == coupon.getShopRange()) {  
        return BaseResult.failure("适用门店不能为空");  
    }  
    if (coupon.getShopRange() == 2 && StringUtils.isBlank(coupon.getShopIds())) {  
        return BaseResult.failure("请指定门店");  
    }  
}
```

```

}
if (null == coupon.getProductRange()) {
    return BaseResult.failure("适用商品不能为空");
}
if (coupon.getProductRange() == 2 && StringUtils.isBlank(coupon.getProductTypeId())) {
    return BaseResult.failure("请指定分类");
}

```

Java面试那些事儿

这样写并没有什么错，还挺工整的，只是看起来不是很优雅而已。

接下来，用Validation来改写这段

Spring Boot文档中的Validation

在 Spring Boot 的官网中，关于Validation只是简单的提了一句，如下

Spring Boot Reference Guide

35. Validation

The method validation feature supported by Bean Validation 1.1 is automatically enabled as long as a JSR-303 implementation (such as Hibernate validator) is on the classpath. This lets bean methods be annotated with `javax.validation` constraints on their parameters and/or on their return value. Target classes with such annotated methods need to be annotated with the `@Validated` annotation at the type level for their methods to be searched for inline constraint annotations.

For instance, the following service triggers the validation of the first argument, making sure its size is between 8 and 10:

```

@Service
@Validated
public class MyBean {

    public Archive findByCodeAndAuthor(@Size(min = 8, max = 10) String code,
        Author author) {
        ...
    }
}

```

Java面试那些事儿

其实，Spring Validator 和Hibernate Validator是两套Validator，可以混着用，这里我们用Hibernate Validator

Hibernate Validator

https://docs.jboss.org/hibernate/stable/validator/reference/en-US/html_single/#preface

Spring Validator

<https://docs.spring.io/spring/docs/5.0.5.RELEASE/spring-framework-reference/core.html#validation>

示例

1、引入spring-boot-starter-validation

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

2、定义一个对象

```
@Data
public class LoginForm {

    @NotBlank(message = "用户名不能为空")
    @Email
    private String username;

    @NotBlank(message = "密码不能为空")
    @Length(min = 6, message = "密码长度至少6位")
    private String password;
}
```

Java面试那些事儿

3、适用@Valid校验，并将校验结果放到BindingResult对象中

```
@Controller
@RequestMapping("/login")
public class LoginController extends BaseController {

    private ObjectError error;

    @GetMapping("/index.html")
    public ModelAndView index() {
        return new ModelAndView("login");
    }

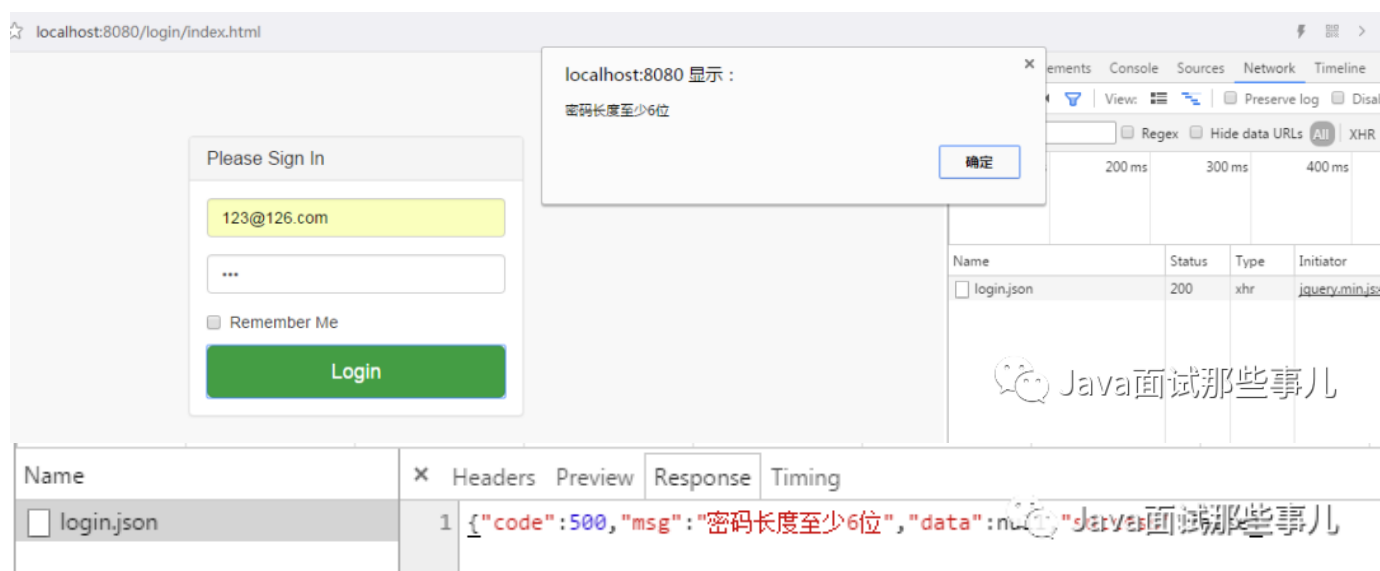
    @PostMapping("/login.json")
    @ResponseBody
    public RespResult login(@Valid LoginForm loginRequest, BindingResult bindingResult) {
        if (bindingResult.hasErrors()) {
            if (bindingResult.hasErrors()) {
                for (ObjectError error : bindingResult.getAllErrors()) {
                    return RespResult.failure(error.getDefaultMessage());
                }
            }
        }
        return RespResult.success();
    }
}
```

Java面试那些事儿

注意：

- 默认情况下，如果校验失败会抛`javax.validation.ConstraintViolationException`异常，可以用统一异常处理去对这些异常做处理
- An `Errors/BindingResult` argument is expected to be declared immediately after the model attribute

4、看效果



如果在校验的对象后面再加上`Model`对象的话，如果返回的是`ModelAndView`就可以将这个`Model`设置到其中，这样在页面就可以取到错误消息了

```
@PostMapping("/login.json")
@ResponseBody
public RespResult login(@Valid LoginForm loginRequest, BindingResult bindingResult, Model model) {
    System.out.println(model);
    if (bindingResult.hasErrors()) {
        if (bindingResult.hasErrors()) {
            for (ObjectError error : bindingResult.getAllErrors()) {
                return RespResult.failure(error.getDefaultMessage());
            }
        }
    }
    return RespResult.success();
}
```

仅仅只是单字段校验的话未免也太不灵活了吧，如果字段之间有关联关系，那该如何校验呢？
答案是自定义

5、自定义校验规则

<https://docs.jboss.org/hibernate/5/stable/validator/reference/en>

6.1. Creating a simple constraint

To create a custom constraint, the following three steps are required:

- Create a constraint annotation
- Implement a validator
- Define a default error message



这里，以优惠券创建为例来演示如何自定义校验规则

首先，优惠券表单如下（仅仅只是演示用）：

```
@Data
@CheckTimeInterval(startTime = "releaseStartTime", endTime = "releaseEndTime", message = "发放开始时间不能大于发放结束时间")
@CheckValidityMode(message = "领取后有效天数不能为空")
public class CouponForm {

    @NotNull(groups = {GroupCouponEdit.class}, message = "优惠券ID不能为空")
    private Long couponId; // 优惠券ID

    @NotNull(groups = {GroupCouponAdd.class, GroupCouponEdit.class}, message = "商家ID不能为空")
    private Integer merchantId; // 商家ID

    @NotBlank(groups = {GroupCouponAdd.class, GroupCouponEdit.class}, message = "优惠券名称不能为空")
    @Length(groups = {GroupCouponAdd.class, GroupCouponEdit.class}, max = 16, message = "优惠券名称最大长度为16")
    private String couponName; // 优惠券名称

    @NotNull(groups = {GroupCouponAdd.class, GroupCouponEdit.class}, message = "优惠券类型不能为空")
    private Integer couponType; // 优惠券类型

    @NotNull
    private Integer parValue; // 面值

    @NotNull
    private Integer quantity; // 发放数量

    private Date releaseStartTime; // 发放开始时间

    private Date releaseEndTime; // 发放结束时间

    @NotNull
    private Integer validityMode; // 有效期模式

    private Integer days; // 领取后多少天内有效

    private Integer limitType; // 限制领取类型

    private Integer limitNum; // 限制领取数量

    private Date validityStartTime; // 有效期开始时间

    private Date validityEndTime; // 有效期结束时间

    @Size(max = 200)
    private String remark; // 备注
}
```



这里除了自定义了两条校验规则之外，还用到了分组。

为什么要有分组这一说呢？因为，举个例子，添加的时候不需要校验id，而修改的时候id不能为空，有了分组以后，就可以添加的时候校验用组A，修改的时候校验用组B

下面重点看一下@CheckTimeInterval

第一步、定义一个注解叫CheckTimeInterval

```
@Target({TYPE, FIELD, METHOD, PARAMETER, ANNOTATION_TYPE, TYPE_USE })
@Retention(RUNTIME)
@Constraint(validatedBy = CheckTimeIntervalValidator.class)
@Documented
@Repeatable(CheckTimeInterval.List.class)
public @interface CheckTimeInterval {

    String startTime() default "from";

    String endTime() default "to";

    String message() default "{org.hibernate.validator.referenceguide.chapter06.CheckCase." +
        "message}";

    Class<?>[] groups() default { };

    Class<? extends Payload>[] payload() default { };

    @Target({TYPE, FIELD, METHOD, PARAMETER, ANNOTATION_TYPE })
    @Retention(RUNTIME)
    @Documented
    @interface List {
        CheckTimeInterval[] value();
    }
}
```

这两个字段就是起止时间的字段名称

Java面试那些事儿

第二步、定义Validator去校验它

```
import org.springframework.beans.BeanWrapper;
import org.springframework.beans.BeanWrapperImpl;

import javax.validation.ConstraintValidator;
import javax.validation.ConstraintValidatorContext;
import java.util.Date;

public class CheckTimeIntervalValidator implements ConstraintValidator<CheckTimeInterval, Object> {

    private String startTime;

    private String endTime;

    @Override
    public void initialize(CheckTimeInterval constraintAnnotation) {
        this.startTime = constraintAnnotation.startTime();
        this.endTime = constraintAnnotation.endTime();
    }

    @Override
    public boolean isValid(Object value, ConstraintValidatorContext context) {
        if (null == value) {
            return true;
        }

        BeanWrapper beanWrapper = new BeanWrapperImpl(value);
        Object start = beanWrapper.getPropertyValue(startTime);
        Object end = beanWrapper.getPropertyValue(endTime);

        if (null == start || null == end) {
            return true;
        }

        int result = ((Date) end).compareTo((Date) start);
    }
}
```

```

    if (result >= 0) {
        return true;
    }

    return false;
}
}

```

顺便提一句，这里BeanWrapper去取对象的属性值，我们稍微看一下BeanWrapper是做什么的

3.4. Bean manipulation and the BeanWrapper

The `org.springframework.beans` package adheres to the JavaBeans standard provided by Oracle. A JavaBean is simply a class with a default no-argument constructor, which follows a naming convention where (by way of an example) a property named `bingoMadness` would have a setter method `setBingoMadness(..)` and a getter method `getBingoMadness()`. For more information about JavaBeans and the specification, please refer to Oracle's website (java.beans).

One quite important class in the beans package is the `BeanWrapper` interface and its corresponding implementation (`BeanWrapperImpl`). As quoted from the javadocs, the `BeanWrapper` offers functionality to set and get property values (individually or in bulk), get property descriptors, and to query properties to determine if they are readable or writable. Also, the `BeanWrapper` offers support for nested properties, enabling the setting of properties on sub-properties to an unlimited depth. Then, the `BeanWrapper` supports the ability to add standard JavaBeans `PropertyChangeListeners` and `VetoableChangeListeners`, without the need for supporting code in the target class. Last but not least, the `BeanWrapper` provides support for the setting of indexed properties. The `BeanWrapper` usually isn't used by application code directly, but by the `DataBinder` and the `BeanFactory`.

The way the `BeanWrapper` works is partly indicated by its name: *it wraps a bean to perform actions on that bean, like setting and retrieving properties.*

The following code snippets show some examples of how to retrieve and manipulate some of the properties of instantiated `Companies` and `Employees`:

```

BeanWrapper company = new BeanWrapperImpl(new Company());
// setting the company name..
company.setPropertyValue("name", "Some Company Inc.");
// ... can also be done like this:
PropertyValue value = new PropertyValue("name", "Some Company Inc.");
company.setPropertyValue(value);

// ok, let's create the director and tie it to the company:
BeanWrapper jim = new BeanWrapperImpl(new Employee());
jim.setPropertyValue("name", "Jim Stravinsky");
company.setPropertyValue("managingDirector", jim.getWrappedInstance());

// retrieving the salary of the managingDirector through the company
Float salary = (Float) company.getPropertyValue("managingDirector.salary");

```

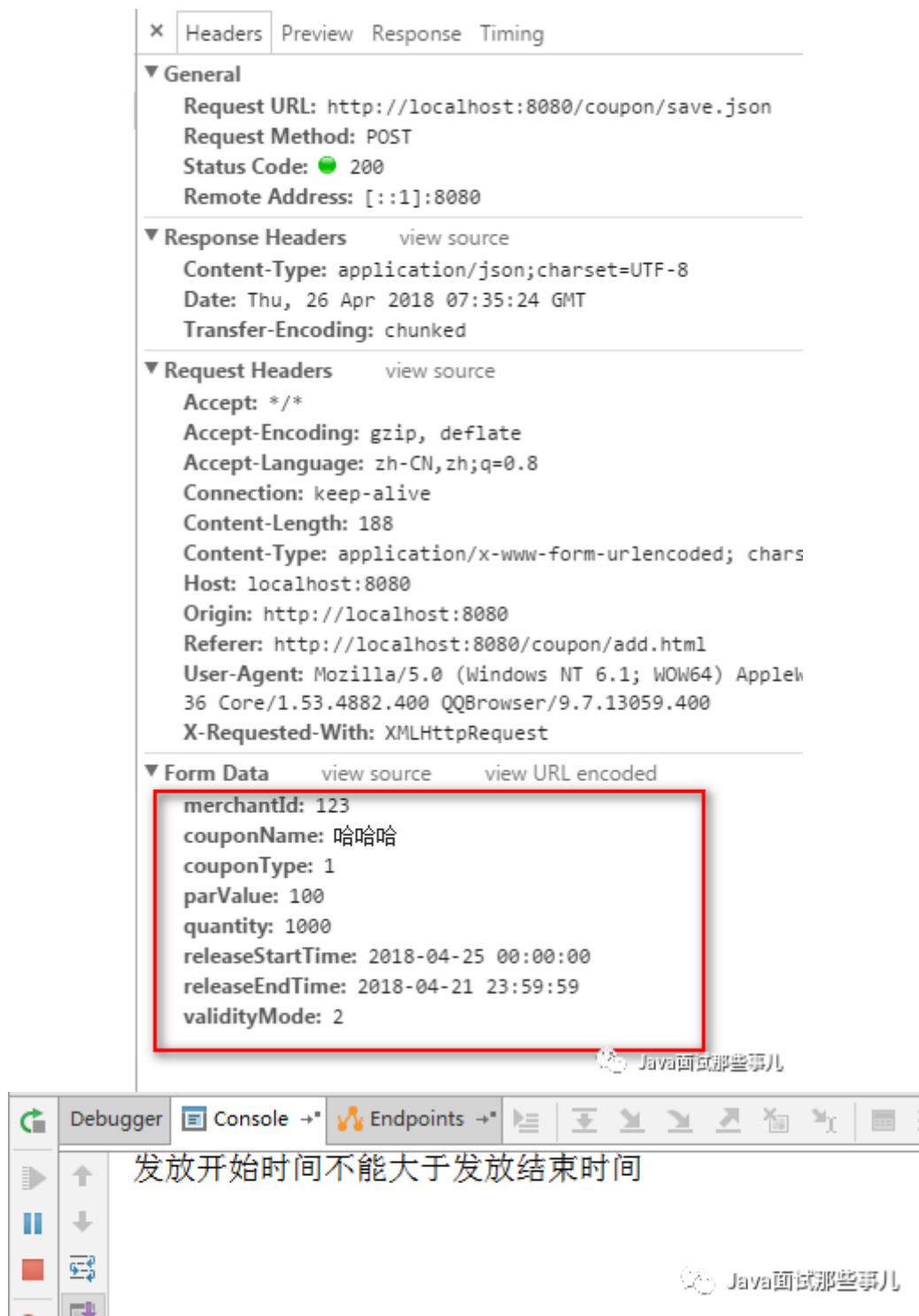
言归正传

第三步、验证

```

@PostMapping("/save.json")
@ResponseBody
public RespResult save(@Valid @Validated(GroupCouponAdd.class) CouponForm couponForm, BindingResult bindingResult) {
    if (bindingResult.hasErrors()) {
        for (ObjectError error : bindingResult.getAllErrors()) {
            System.out.println(error.getDefaultMessage());
        }
    }
    return RespResult.success();
}

```



看，自定义的校验生效了

补充

1、校验模式

https://docs.jboss.org/hibernate/stable/validator/reference/en-US/html_single/#section-fail-fast

下面补充一点，关于校验模式

默认会校验元数据所有属性，然后将错误信息一起返回，但很多时候不需要这样，一个校验失败了，其它就不必校验了

为此，需要这样设置

```
import javax.validation.Validation;
import javax.validation.Validator;
import javax.validation.ValidatorFactory;

@Configuration
public class ValidatorConfig {

    @Bean
    public Validator validator() {
        ValidatorFactory validatorFactory = Validation.byProvider(HibernateValidator.class)
            .configure()
            .failFast(true)
            .buildValidatorFactory();
        Validator validator = validatorFactory.getValidator();
        return validator;
    }
}
```

Java面试那些事儿

2、单个参数校验

```
@Controller
@RequestMapping("/coupon")
@Validated
public class CouponController extends BaseController {

    @GetMapping("/detail.html")
    public ModelAndView detail(@NotNull(message = "ID不能为空") Long id) {
        ModelAndView modelAndView = new ModelAndView("coupon/detail");
        // TODO 查询
        return modelAndView;
    }
}
```

Java面试那些事儿

localhost:8080/coupon/detail.html?id=

500

Java面试那些事儿



如果是调整页面的时候参数校验失败的话，这时可以不做处理，让其调到错误页面。

如果是接口参数校验失败的话，可以在这里进行统一处理，并返回。例如：

```
@ControllerAdvice
@Component
public class GlobalExceptionHandlerAdvice {
```

```

public class GlobalExceptionHandlerAdvice {

    @ExceptionHandler({ConstraintViolationException.class})
    public RespResult handler(HttpServletRequest request, ConstraintViolationException ex) {
        if (WebUtils.isAjax(request)) {
            StringBuffer sb = new StringBuffer();
            for (ConstraintViolation violation : ex.getConstraintViolations()) {
                sb.append(violation.getMessage());
            }
            return RespResult.failure(sb.toString());
        }
        throw ex;
    }
}

```

Java面试那些事儿

3、错误页面

```

@SpringBootApplication
public class CjsSpringbootExampleApplication {

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

    @Bean
    public ErrorPageRegistrar errorPageRegistrar() {
        return new ErrorPageRegistrar() {
            @Override
            public void registerErrorPages(ErrorPageRegistry registry) {
                registry.addErrorPages(new ErrorPage(HttpStatus.BAD_REQUEST, path: "/400.html"));
                registry.addErrorPages(new ErrorPage(HttpStatus.FORBIDDEN, path: "/403.html"));
                registry.addErrorPages(new ErrorPage(HttpStatus.NOT_FOUND, path: "/404.html"));
                registry.addErrorPages(new ErrorPage(HttpStatus.INTERNAL_SERVER_ERROR, path: "/500.html"));
            }
        };
    }

    // @Bean
    // public ErrorViewResolver MyErrorViewResolver() {
    //     return new ErrorViewResolver() {
    //         @Override
    //         public ModelAndView resolveErrorView(HttpServletRequest request, HttpStatus status, Map<String, Object>
    //             return null;
    //     }
    // }
}

```

Java面试那些事儿

```

resources
├── public
│   ├── 400.html
│   ├── 404.html
│   └── 500.html
├── static
└── templates
application

```

以刚才优惠券详情为例

http://localhost:8080/coupon/detail.html 400

http://localhost:8080/coupon/detail.html?id= 400

http://localhost:8080/coupon/detail.html?id=abc 400

http://localhost:8080/coupon/detail222.html?id=123 404

<http://localhost:8080/coupon/detail222.html?id=123> 404

无权限 403

int a = 1 / 0; 500

4、@Valid与@Validated

https://blog.csdn.net/qq_27680317/article/details/79970590

参考

<http://rensanning.iteye.com/blog/2357373>

<https://blog.csdn.net/kenight/article/details/77774465>

<https://www.cnblogs.com/mr-yang-localhost/p/7812038.html>

<https://www.phpsong.com/3567.html>

<https://www.cnblogs.com/mr-yang-localhost/p/7812038.html>

热文推荐

百度面试官：说说MySQL是如何对一条简单的更新语句加锁的？

我用这款神器攻破了公司的登录系统，设计的验证码简直不堪一击，附下载地址...

你竟然不知道分布式与集群的区别？



