1.嵌套的输入合法检查

@Valid进行检查，但会全部检查

@Validated进行分组检查

2.ControllerExceptionHandler中注入的HttpServletResponse

Controller的作用域是单例的

它的成员变量HttpServletResponse的作用域不是单例的，是原型/请求的

HttpServletResponse在接收到HTTP请求后更新

3.Singleton类型的Spring容器Bean对象是如何实现并发的

Spring容器中的Bean对象默认为Singleton类型，即只有一个对象。

有状态对象 Stateful Bean

无状态对象 Stateless Bean

servlet是多线程执行的，多线程是容器提供的能力。 servlet为了能并发执行， 是因为servlet被这些thread使用，tomcat里创建响应的socketServer线程类接收请求连接，然后在再创建或引用对应的servlet实例来处理请求连接。

当多用户同时请求一个服务时，容器会给每一个请求分配一个线程，这是多个线程会并发执行该请求多对应的业务逻辑（成员方法）

4.低效的异常

抛出异常，

BindingResult

if语句

高并发下（1万请求/秒以上），偶尔会出现CPU占用率呈递增状态的情况，只有在测试快结束的时候才能达到99%

高并发下（1万请求/秒以上），同上，偶然情况下处理速率呈递增状态，无法确定精确的处理速率

使用BindingResult是提高服务器处理效率的较佳方案。但是手动验证的灵活性较高。

抛出异常对程序执行速度的影响情况

C#中7个int赋值语句和7个try catch语句块耗时相同大约28ms（Debug）

一个throw语句大约耗时148838ms（Debug）

Java中7个int赋值语句和7个try catch语句块耗时相同大约10ns（Release）

一个throw语句大约耗时7200ns（ Release ）

生成错误堆栈可能是导致throw语句耗时较多的原因之一

5.哪个更快

方案1. 在MyBatis中用SQL完成表的关联，返回关联的对象模型

方案2. 在MyBatis中单独查询一个对象，在Dao层完成对象的关联

方案一在dao层返回的对象就是我们需要的对象模型

方案二需要分两次查询数据库，用Java语句关联

使用大负载的数据库，尽量不用Join语句关联，而在dao层使用Java代码来处理

更好的测试方案：分在两台服务器上测试，Jmeter测试

从性能上说，在sql完成连表返回对象模型的方案对于请求的响应速度要略快于在dao层完成对象关联的方案，但是快的不多。

从可管理性的角度说，在dao层完成对象关联的方案要优于在sql完成连表返回对象模型的方案。

10.29 MySQL的索引

Myisam和InnoDB都是B+树索引

Myisam可以没有主键索引，存的是数据地址

InnoDB一定要有主键索引，没有就默认建，存的是数据

Myisam数据、索引分开存放，InnoDB数据、索引一起存放

Myisam不支持事务，InnoDB支持事务

InnoDB还有聚簇索引，辅助索引是二次索引

• 选择 MyISAM 的场合：

• 操作简单，不需要用到事务

• 追求查询、更新、插入的效率

• 并发量不⼤

• 需要经常查询表的记录数 (count)

• 选择 InnoDB 的场合：

• 需要事务⽀持

• 需要⾏级锁的⽀持

• 需要定义外键

• 并发量⼤

可能的问题：

1、为什么MYSQL做了两个引擎Myisam和InnoDB，我们数据库需要用两个吗？

对于磁盘IO，InnoDB更快，对于随机读写（SSD），Myisam更快

Mysiam：只读不写，或者写操作很少（地区表、商品分类）

2、MVCC单独UPdate一个事务的时候，如何回滚？

3、如果MVCC读完第一个事务后，又有第二个事务对数据进行操作，又如何回滚？