



(12)发明专利申请

(10)申请公布号 CN 106445822 A

(43)申请公布日 2017. 02. 22

(21)申请号 201610846694.X

(22)申请日 2016.09.23

(71)申请人 上海图聚智能科技股份有限公司

地址 200082 上海市杨浦区民京路853号1
幢203室

(72)发明人 袁晓春

(74)专利代理机构 上海申汇专利代理有限公司

31001

代理人 翁若莹 柏子霖

(51)Int.Cl.

G06F 11/36(2006.01)

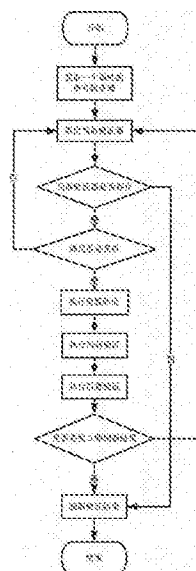
权利要求书1页 说明书2页 附图1页

(54)发明名称

高扩展性的前后端数据校验方法

(57)摘要

本发明提供了一种高扩展性的前后端数据校验方法,其特征在于,一次性验证多个属性,每个属性指定多个验证器,验证器在满足制定条件下运行,或在属性中添加注解,在注解中指定验证器完成一次验证,其中,多个验证器组成验证器链进行不同条件下的验证,为验证器链中的验证器指定前置验证器和后置验证器,所有验证器共享一个验证上下文,验证上下文中存储验证器链的共享变量。本发明提供的一种高扩展性的前后端数据校验方法实现了验证器在不同条件下的复用,而且增加了验证的灵活性,提高了代码的简洁度。



1. 一种高扩展性的前后端数据校验方法,其特征在于,一次性验证多个属性,每个属性指定多个验证器,验证器在满足制定条件下运行,或在属性中添加注解,在注解中指定验证器完成一次验证,其中,多个验证器组成验证器链进行不同条件下的验证,为验证器链中的验证器指定前置验证器和后置验证器,所有验证器共享一个验证上下文,验证上下文中存储验证器链的共享变量。

2. 如权利要求1所述的一种高扩展性的前后端数据校验方法,其特征在于,验证某个属性的方法包括以下步骤:

步骤1、获取当前属性的验证器链,从验证器链中取出第一个验证器作为当前验证器;

步骤2、判断当前验证器是否存在,若存在,进入步骤3,若不存在,进入步骤8;

步骤3、当前验证器的运行条件是否满足,若不满足,则将下一个验证器作为当前验证器,返回步骤2,若满足,则进入步骤4;

步骤4、执行当前验证器的前置验证器,进行前置验证;

步骤5、执行当前验证器,进行当前验证;

步骤6、执行当前验证器的后置验证器,进行后置验证;

步骤7、判断是否需要立即刷新验证结果,若需要,则进入步骤8,若不需要,则将下一个验证器作为当前验证器,返回步骤2;

步骤8、获取验证结果后结束。

高扩展性的前后端数据校验方法

技术领域

[0001] 本发明涉及一种数据验证方法。

背景技术

[0002] 在传统的信息系统开发中,服务端数据验证是一个必须的步骤,除了基本的数据格式验证之外,复杂的业务逻辑验证在代码中也到处存在。随着业务量的增加,大量的重复验证逻辑充斥在代码中,一个简单的存储业务,光验证逻辑就占据了大量的开发量,且造成了后期维护的不便。

[0003] 目前,服务端开发的数据验证可以采用的验证方式,如基于XML验证的Apache Commons Validator,基于JSR303的Hibernate Validator。这些框架可以满足基本的数据验证,但是一旦需要实现一些复杂的自定义业务逻辑,往往需要一大堆复杂的配置,开发成本也随之提高。在某些情况下,如果验证逻辑需要在某一个特定条件下执行或者验证逻辑之间也存在依赖关系,往往的解决方式就是在代码中加入一堆判断条件,然后调用对应的验证逻辑,这会使得组件的复用性大大降低。从代码可读性与维护性角度来讲,这显然不是开发人员所期望的。

发明内容

[0004] 本发明的目的是提供一种组件复用机制,能够在特定条件下执行某一个验证组件。

[0005] 为了达到上述目的,本发明的技术方案是提供了一种高扩展性的前后端数据校验方法,其特征在于,一次性验证多个属性,每个属性指定多个验证器,验证器在满足制定条件下运行,或在属性中添加注解,在注解中指定验证器完成一次验证,其中,多个验证器组成验证器链进行不同条件下的验证,为验证器链中的验证器指定前置验证器和后置验证器,所有验证器共享一个验证上下文,验证上下文中存储验证器链的共享变量。

[0006] 优选地,验证某个属性的方法包括以下步骤:

[0007] 步骤1、获取当前属性的验证器链,从验证器链中取出第一个验证器作为当前验证器;

[0008] 步骤2、判断当前验证器是否存在,若存在,进入步骤3,若不存在,进入步骤8;

[0009] 步骤3、当前验证器的运行条件是否满足,若不满足,则将下一个验证器作为当前验证器,返回步骤2,若满足,则进入步骤4;

[0010] 步骤4、执行当前验证器的前置验证器,进行前置验证;

[0011] 步骤5、执行当前验证器,进行当前验证;

[0012] 步骤6、执行当前验证器的后置验证器,进行后置验证;

[0013] 步骤7、判断是否需要立即刷新验证结果,若需要,则进入步骤8,若不需要,则将下一个验证器作为当前验证器,返回步骤2;

[0014] 步骤8、获取验证结果后结束。

[0015] 本发明提供了一种高扩展性的前后端数据校验方法实现了验证器在不同条件下的复用,而且增加了验证的灵活性,提高了代码的简洁度。

附图说明

[0016] 图1为本发明的一个属性验证的流程图。

具体实施方式

[0017] 下面结合具体实施例,进一步阐述本发明。应理解,这些实施例仅用于说明本发明而并不用于限制本发明的范围。此外应理解,在阅读了本发明讲授的内容之后,本领域技术人员可以对本发明作各种改动或修改,这些等价形式同样落于本申请所附权利要求书所限定的范围。

[0018] 验证组件在本发明中称为验证器,是整个验证组件的执行者。在本发明中,利用多个验证器组成验证器链,以进行不同条件下的验证,并且可以定义组合式验证器,为某一个验证器指定前置验证器和后置验证器。所有验证器共享一个验证上下文,验证上下文中可以存储验证器链的共享变量。可以在某一个验证过程中强制验证提交以跳过后续的验证。除此之外,本发明还提供了注解式的验证方式,只需要将定义的验证器加入到注解属性中,执行时便可自动调用对应的验证逻辑。下面分别从普通属性验证与注解式验证来叙述本发明的实现方式:

[0019] 1、普通属性验证

[0020] 本发明提供了类似JDK8的Stream式验证方式,可以一次性验证多个属性,每一个属性可以指定多个验证器,验证器可以在满足指定的条件下执行,并且采用了惰性求值的方式,一旦校验到错误,并不会把错误立即返回,除非手动刷新错误或者执行完毕。

[0021] 2、Bean属性验证

[0022] 除了可以使用基于属性的验证方式之外,还可以使用注解方式,通过在属性中添加@Check注解,同时在注解中指定验证器便可完成一次验证。

[0023] 普通属性验证与Bean属性验证各有特点,前者增加了验证的灵活性,后者提高了代码的简洁度。在实际开发中,可以同时使用这两种方式进行混合验证。

[0024] 结合图1,本发明验证一个属性的具体流程包括以下步骤:

[0025] 步骤1、获取当前属性的验证器链,从验证器链中取出第一个验证器作为当前验证器;

[0026] 步骤2、判断当前验证器是否存在,若存在,进入步骤3,若不存在,进入步骤8;

[0027] 步骤3、当前验证器的运行条件是否满足,若不满足,则将下一个验证器作为当前验证器,返回步骤2,若满足,则进入步骤4;

[0028] 步骤4、执行当前验证器的前置验证器,进行前置验证;

[0029] 步骤5、执行当前验证器,进行当前验证;

[0030] 步骤6、执行当前验证器的后置验证器,进行后置验证;

[0031] 步骤7、判断是否需要立即刷新验证结果,若需要,则进入步骤8,若不需要,则将下一个验证器作为当前验证器,返回步骤2;

[0032] 步骤8、获取验证结果后结束。

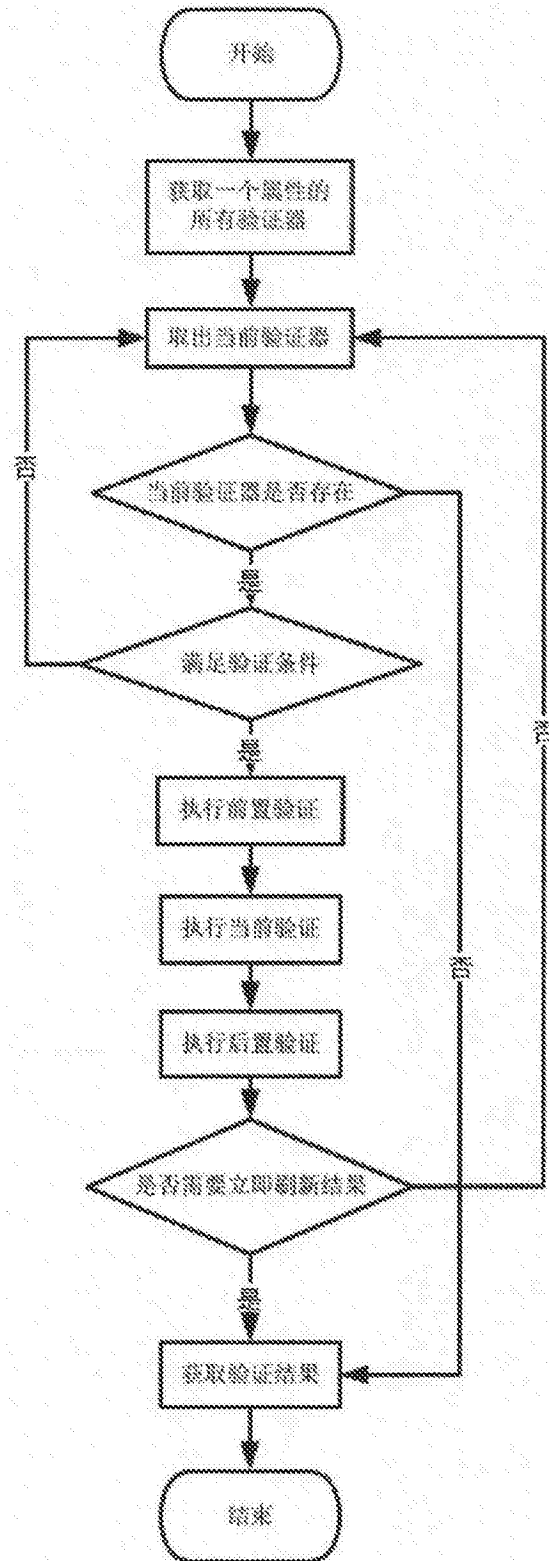


图1