校验工具——validator

用来做参数的验证。

*/\*\*  
 \* 数据校验工具类：  
 \* 请求传进来时完成对参数的处理  
 \*/***public class** BeanValidator {  
  
 */\*\*  
 \* 构建全局校验工厂  
 \*/* **private static** ValidatorFactory *validatorFactory* = Validation.*buildDefaultValidatorFactory*();  
  
 */\*\*  
 \* 普通的校验方法  
 \*  
 \** ***@param:*** *[t, groups]  
 \** ***@return:*** *java.util.Map*<*java.lang.String*,*java.lang.String*>  
 *\** ***@author:*** *nino  
 \** ***@date:*** *2019/5/26 10:19  
 \*/* **public static** <T> Map<String, String> validate(T t, Class... groups) {  
  
 *//获取校验实例* Validator validator = *validatorFactory*.getValidator();  
 *//使用校验器去校验  
 //校验不通过才会产生Set* Set validateResult = validator.validate(t, groups);  
  
 **if** (validateResult.isEmpty()) {  
 **return** Collections.*emptyMap*();  
 } **else** {  
 LinkedHashMap errors = Maps.*newLinkedHashMap*();  
 Iterator iterator = validateResult.iterator();  
 **while** (iterator.hasNext()) {  
 *//每次校验遍历出来都是一个ConstraintViolation* ConstraintViolation violation = (ConstraintViolation) iterator.next();  
 *//key是校验字段，value是错误信息* errors.put(violation.getPropertyPath().toString(), violation.getMessage());  
 }  
 **return** errors;  
 }  
 }  
  
 */\*\*  
 \* 传入List的校验  
 \*  
 \** ***@param:*** *[collection]  
 \** ***@return:*** *java.util.Map*<*java.lang.String*,*java.lang.String*>  
 *\** ***@author:*** *nino  
 \** ***@date:*** *2019/5/26 12:55  
 \*/* **public static** Map<String, String> validateList(Collection<?> collection) {  
  
 *//guava中的校验，非空判断* Preconditions.*checkNotNull*(collection);  
 Iterator iterator = collection.iterator();  
 Map errors;  
 **do** {  
 **if** (!iterator.hasNext()) {  
 **return** Collections.*emptyMap*();  
 }  
 Object object = iterator.next();  
 errors = *validate*(object, **new** Class[0]);  
 } **while** (errors.isEmpty());  
  
 **return** errors;  
 }  
  
 */\*\*  
 \* 进一步封装：对象类型的验证  
 \** ***@param:*** *[first, objects]  
 \** ***@return:*** *java.util.Map*<*java.lang.String*,*java.lang.String*>  
 *\** ***@author:*** *nino  
 \** ***@date:*** *2019/5/26 15:38  
 \*/* **public static** Map<String, String> validateObject(Object first, Object... objects) {  
 **if** (objects != **null** && objects.**length** > 0) {  
 **return** *validateList*(Lists.*asList*(first, objects));  
 } **else** {  
 **return** *validate*(first, **new** Class[0]);  
 }  
 }  
  
 */\*\*  
 \* 检查参数是否符合类型  
 \*  
 \** ***@param param*** *\** ***@throws*** *ParamException  
 \*/* **public static void** check(Object param) **throws** ParamException {  
 Map<String, String> map = BeanValidator.*validateObject*(param);  
 *//参数不符合则抛出异常  
 //map != null && map.entrySet().size() > 0* **if** (MapUtils.*isNotEmpty*(map)) {  
 **throw new** ParamException(map.toString());  
 }  
 }  
}

