使用 validation 自定义注解

作者：ellenbboe
原文链接：https://ld246.com/article/1575889341007
来源网站：链滴

第一步 在pom中引入

```
 <dependency>
     <groupId>org.springframework.boot</groupId>
     <artifactId>spring-boot-starter-validation</artifactId>
     <version>2.1.6.RELEASE</version>
   </dependency>
```

第二步 创建一个普通的类将至变成注解

```
@Target({ METHOD, FIELD, ANNOTATION_TYPE, CONSTRUCTOR, PARAMETER, TYPE_USE })
@Retention(RUNTIME)
@Documented
@Constraint(validatedBy = {IsMobileValidator.class})
public @interface IsMobile {
   String message() default "手机号码格式错误";
   Class<?>[] groups() default { };
   Class<? extends Payload>[] payload() default { };
}
```

步骤就是讲class替换成 @interface.然后将其他的复制出来就行,message() default 需要自己写

第三步 创建另一个类继承ConstraintValidator<>接口

```
public class IsMobileValidator implements ConstraintValidator<IsMobile,String> {
   public static final String REGEX_MOBILE = "^((17[0-9])|(14[0-9])|(13[0-9])|(15[^4,\\D])|(18[0,
-9]))\\d{8}$";
   @Override
   public void initialize(IsMobile constraintAnnotation) {
   }

   @Override
   public boolean isValid(String value, ConstraintValidatorContext context) {
      try {
         if(StringUtils.isEmpty(value))
         {
            return false;
         }
         return Pattern.matches(REGEX_MOBILE,value);
      } catch (Exception e) {
         return false;
      }

   }
}
```

在isValid下写判断逻辑

最后就可以使用@isMobile

```
   @IsMobile
   private String phone;
```

原文链接：使用 validation 自定义注解