

Dynamic validation for the input type

📌 USSD changes discussion with @Rupesh Kansara @Mayank Kansara @Omololu Fabelurin we need to create custom validator.

✅ Bean level data validation for spring boot.

Objective:

- create a normal bean level validator.
- create custom bean level validator.
- create Dynamic messaging.

📌 author @indranil banerjee

Creating Normal inbuild validation:

for request body whatever request body is being used you have to create inbuild validation shown in the picture

```
@Data
@AllArgsConstructor(staticName = "build")
@NoArgsConstructor
public class UserRequest {
    @NotNull(message = "username shouldn't be null")
    private String name;
    @Email(message = "invalid email address")
    private String email;
```

Creating Custom Validator:

- to create a custom validator, one has to create a custom annotation.

```
@Target({ElementType.FIELD, ElementType.PARAMETER})
@Retention(RetentionPolicy.RUNTIME)
@Documented
@Constraint(validatedBy = PhoneNumberCodeTypeValidator.class)
public @interface PhoneNumberCodeType {

    1 usage
    public String message() default "PhoneNumber should start with 91";

    no usages
    Class<?>[] groups() default {};

    no usages
    Class<? extends Payload>[] payload() default {};
}
```

- then one has to create a custom validator check it can be used for business case validation.

```

public class PhoneNumberCodeTypeValidator implements ConstraintValidator<PhoneNumberCodeType, String> {

    @Override
    public void initialize(PhoneNumberCodeType constraintAnnotation) {
        ConstraintValidator.super.initialize(constraintAnnotation);
    }

    @Override
    public boolean isValid(String phoneNumber, ConstraintValidatorContext constraintValidatorContext) {
        Pattern ptrn = Pattern.compile( regex: "(?:0091|\\+91|0)[7-9][0-9]{9}$");
        Matcher match = ptrn.matcher(phoneNumber);
        return (match.find() && match.group().equals(phoneNumber));
    }
}

```

- then one has to annotate the same with the validation.

```

@email(message = "invalid email address")
private String email;

@PhoneNumberCodeType(message = "{phoneNumber.not.correct}")
private String mobile;

```

Dynamic Messaging for the Validation:

of dynamic messaging create one **messages.properties** file and map that in the bean class path.

```

@Bean("testValidation")
public MessageSource messageSource(){
    ReloadableResourceBundleMessageSource messageSource= new ReloadableResourceBundleMessageSource();
    messageSource.setBasename("classpath:messages");
    messageSource.setDefaultEncoding("UTF-8");
    return messageSource;
}

no usages

@Bean
public LocalValidatorFactoryBean validator( @Qualifier("testValidation") MessageSource messageSource){
    LocalValidatorFactoryBean bean =new LocalValidatorFactoryBean();
    bean.setValidationMessageSource(messageSource);
    return bean;
}

no usages

@Bean
public LocaleResolver localeResolver(){
    SessionLocaleResolver local=new SessionLocaleResolver();
    local.setDefaultLocale(Locale.US);
    return local;
}

```

then create a custom message in your **messages.properties** file and use that in your validator.

```
phoneNumber.not.correct=Please provide correct indian phone number
```

Activating The validation:

annotate your required bean which need validation with

```
no usages
@PostMapping("/save")
public ResponseEntity<User> saveUser(@RequestBody @Valid UserRequest userRequest){
    return new ResponseEntity<>(service.saveUser(userRequest), HttpStatus.CREATED);
}

no usages
@GetMapping("/fetchAll")
public ResponseEntity<List<User>> getAllUsers() { return ResponseEntity.ok(service.get
```

Test Result for Custom Validator:

ST http://localhost:8084/users/save Send 400 Bad Request 13.2 s 55 B 33 Minutes Ago

Parameters JSON Auth Headers 2 Docs Preview Headers 4 Cookies Timeline

```
1 {
2   "name": "indranil",
3   "email": "banerjee.7@gmail.com",
4   "mobile": "9836170979",
5   "gender": "M",
6   "age": "34",
7   "nationality": "India"
8 }
9 }
```

```
1 {
2   "mobile": "Please provide correct indian phone number"
3 }
```

Source Code:



validation-ex...ain.zip
09 Apr 2024, 05:17 PM

Latest Code:

[indranb13/validation-exception-handling-main](#)