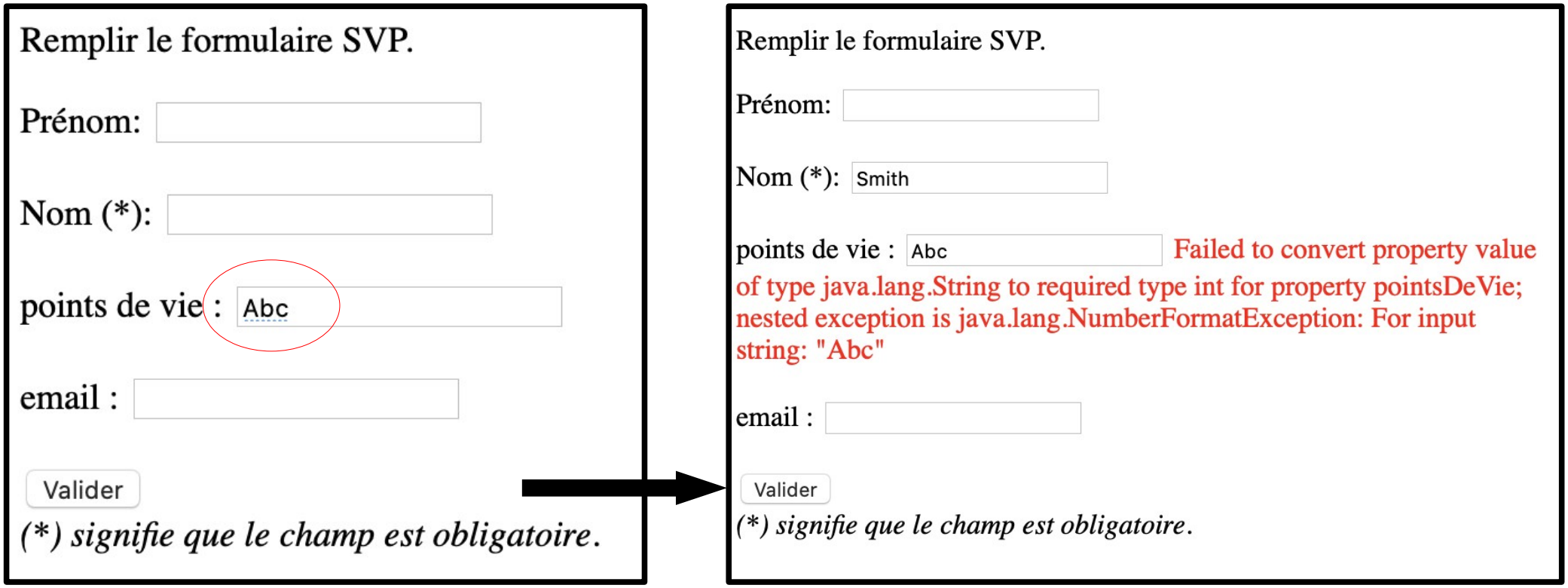
Spring MVC - Formulaires

Validation d'un champ prévu pour contenir nombre dans lequel on saisi autre chose

Cas d'échec de validation non géré



Si on fournit autre chose qu'un nombre , spring ne pouvant "caster" la saisie en nombre , il se produit une NumberFormatException et un message déplaisant est remonté à l'utilisateur .
Nous devons gérer le cas présent autrement

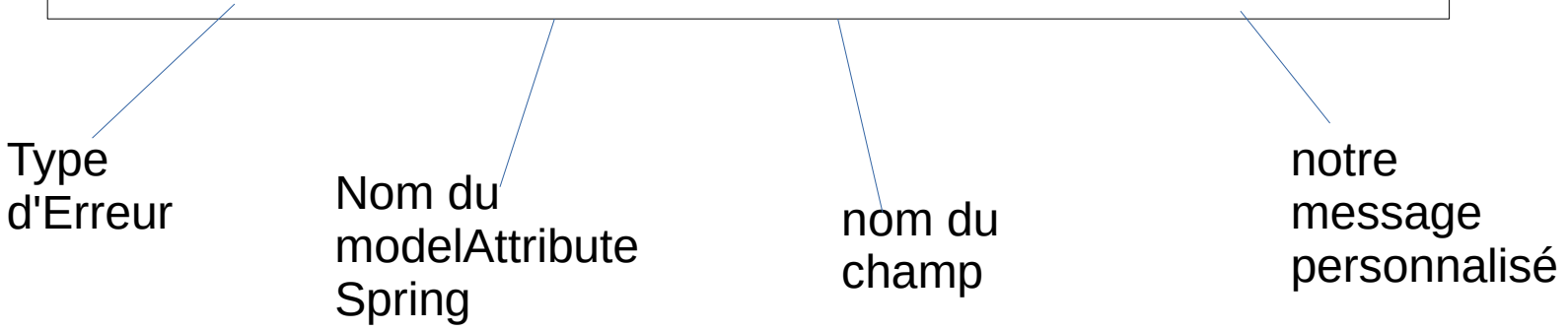
message.properties

- La solution est de spécifier un message d'erreur , pour remplacer celui affiché par défaut par la NumberFormatException

NB

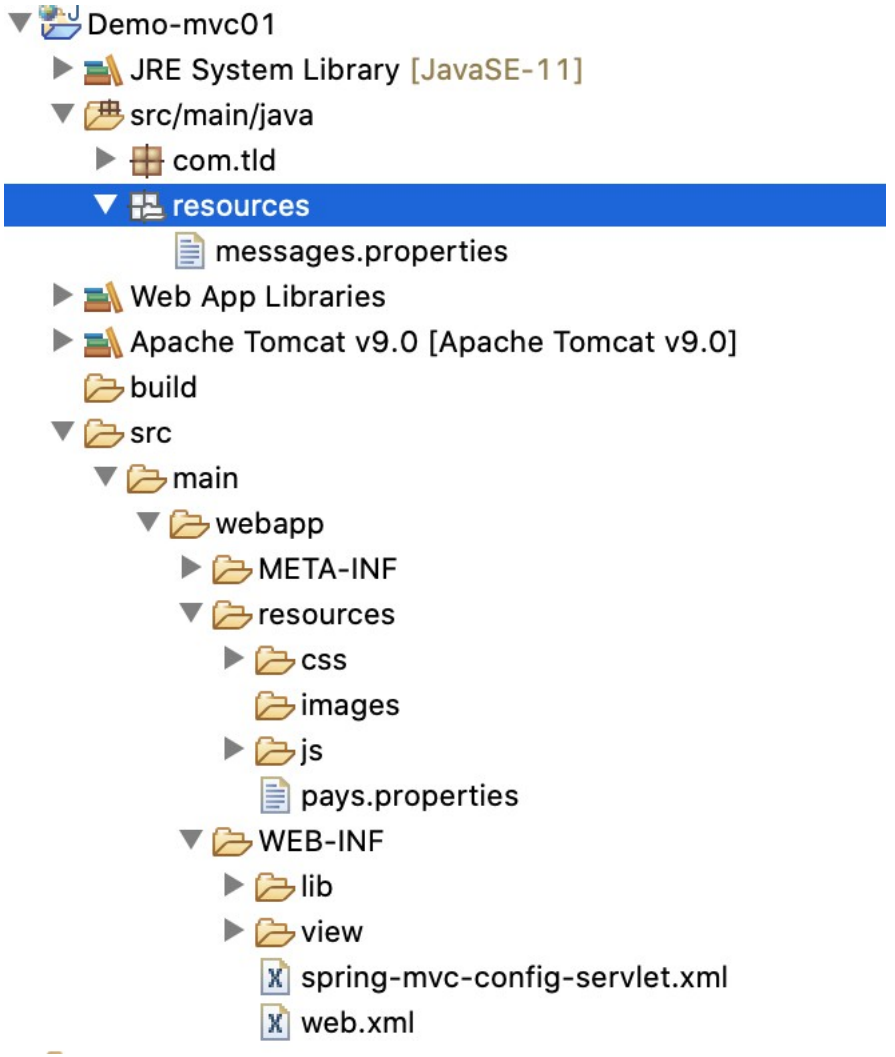
- Créons un fichier src/resources/message.properties

```
typeMismatch.personnage.pointsDeVie=nombre invalide
```



src/resources.....

NB



Message.properties

- Charger le fichier message.properties dans la configuration de spring : WebContent/WEB-INF/spring-mvc-config-servlet.xml

```
...etc

    <!--  Load custom message resources  -->
    <bean  id="messageSource"
        class="org.springframework.context.support.ResourceBundleMessageSource">
            <property name="basenames" value="resources/messages"/>
    </bean>

</beans>
```

Détail de la validation

- On peut monitorer la validation plus précisément en imprimant le BindingResult

```
@RequestMapping("/processForm")
public String processForm(
@Valid  @ModelAttribute("personnage") Personnage
unPersonnage,
BindingResult leBindingResult){
System.out.println("Nom: |" + unPersonnage.getNom()+"|");
System.out.println("Binding result: "+leBindingResult);

 if(leBindingResult.hasErrors()) {
 System.out.println("il y a des erreurs retour au
formulaire ");
 return "personnage-form";}
 else  {
 System.out.println("zéro  erreur de validation ");
 return "personnage-vue";

 }
}
```

Ré-exécutons le cas de test

Remplir le formulaire SVP.

Prénom:

Nom (*):

points de vie : Abc

email :

Valider

(*) signifie que le champ est obligatoire.

Remplir le formulaire SVP.

Prénom:

Nom (*): Smith

points de vie : Abc nombre invalide

email :

Valider

(*) signifie que le champ est obligatoire.

Markers | Servers | Properties | Data Source Explorer | Snippets | Console

Tomcat v9.0 Server at localhost [Apache Tomcat] /Library/Java/JavaVirtualMachines/openjdk-11.0.2.jdk/Contents/Home/bin/java (25 juil. 2020 à 23:27:01)

```
Nom: |Smith|
Binding result: org.springframework.validation.BeanPropertyBindingResult: 1 errors
Field err in object 'personnage' on field 'pointsDeVie': rejected value [Abc]; codes [typeMismatch.personnage.pointsDeVie,typeMismatch.pointsDeVie,typ
il y a des erreurs retour au formulaire
```

PersonnageControleur.java

```
System.out.println("Binding result: "+leBindingResult);
```

Cumul des règles de validation

- Pour terminer avec le champ numérique pointsDeVie, il est possible de cumuler les règles comme suit :

```
@NotNull(message="champ requis")
@Min(value=0, message=" doit être supérieur ou égal à 0")
@Max(value=10,message="doit être inférieur ou égal à 10")
Private Integer pointsDeVie ;

//getter & setter à mettre à jour
```