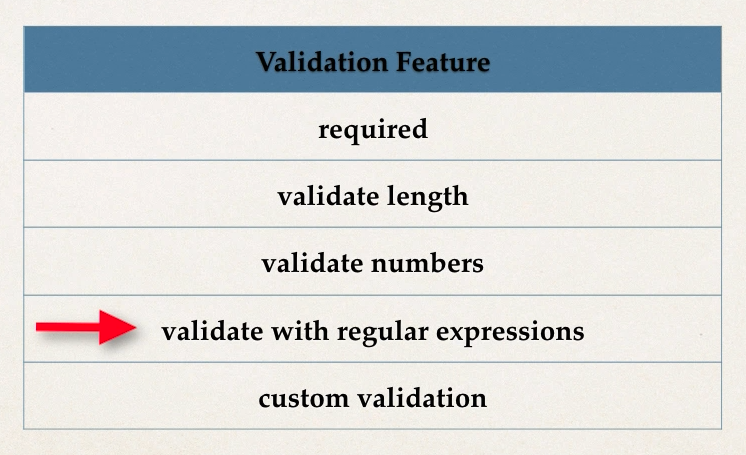
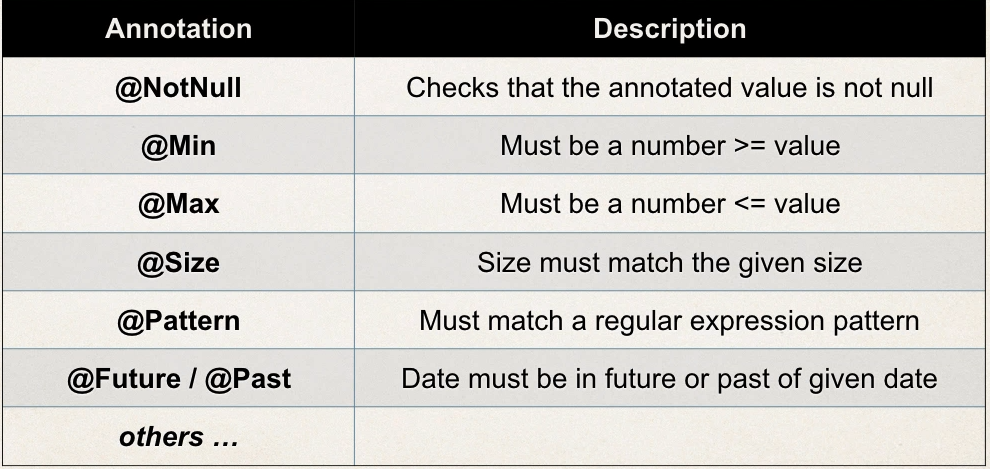


Java has a standard Bean Validation API. This is available for both server and client-side apps.



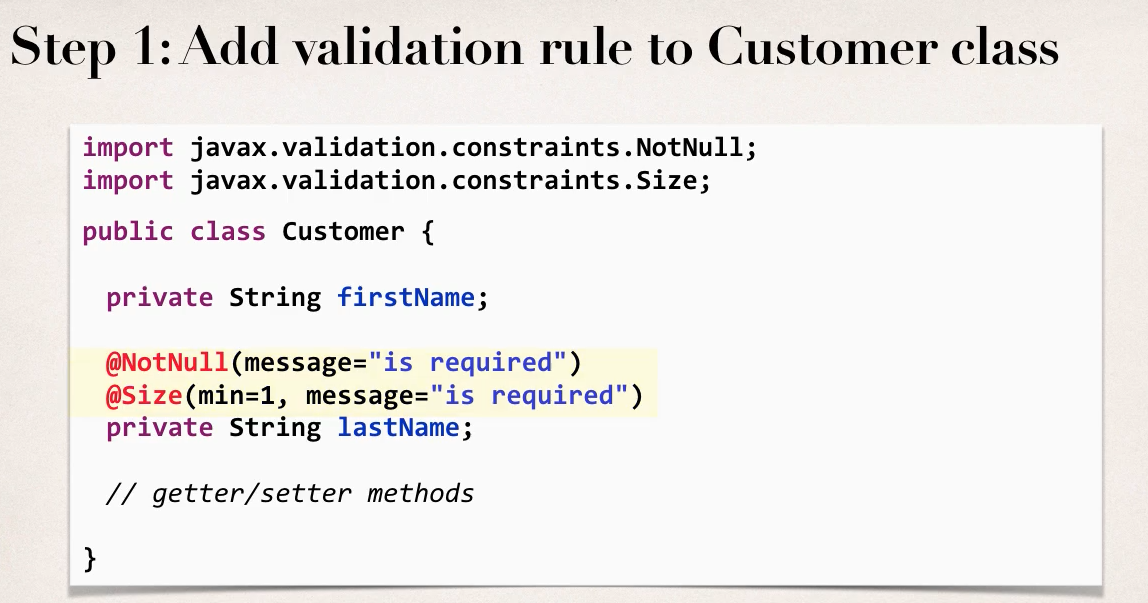


1. **Setting up Development Environment:** We will use hibernet validator. We can use this maven repo; the steps are explained in the official site. Basically we need 3 repo: hibernate-validator, jakarta.el, hibernate-validator-cdi.

<https://hibernate.org/validator/documentation/getting-started/>

I haven’t been able to run it correctly.

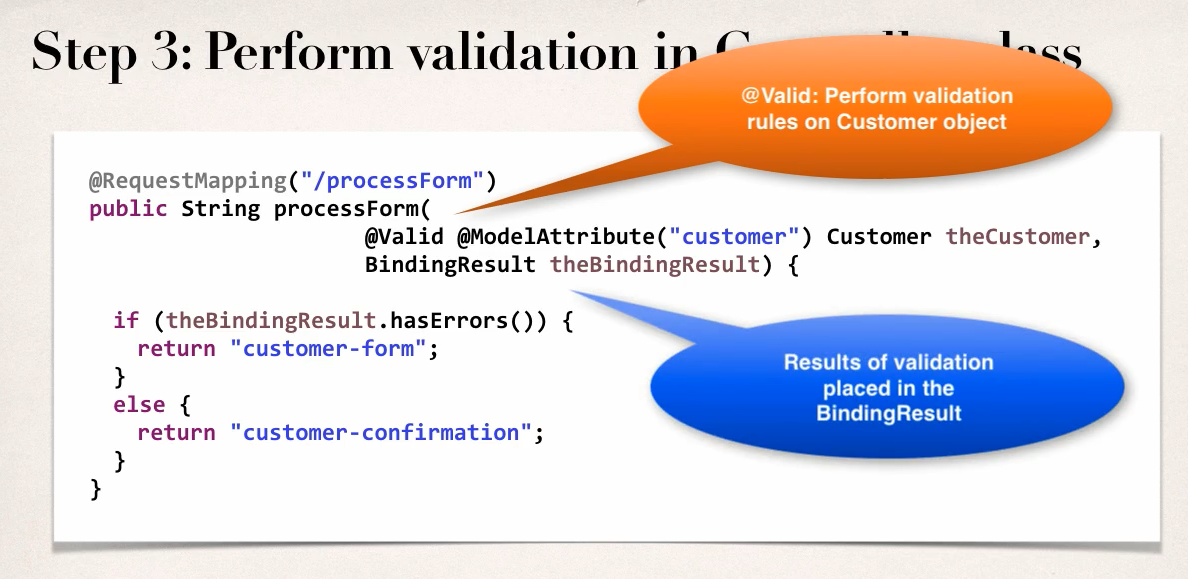
1. **Add Validation rule to Customer class:** Pretty self-explanatory. Just use annotation to apply validations. Here we define the rules for validations.



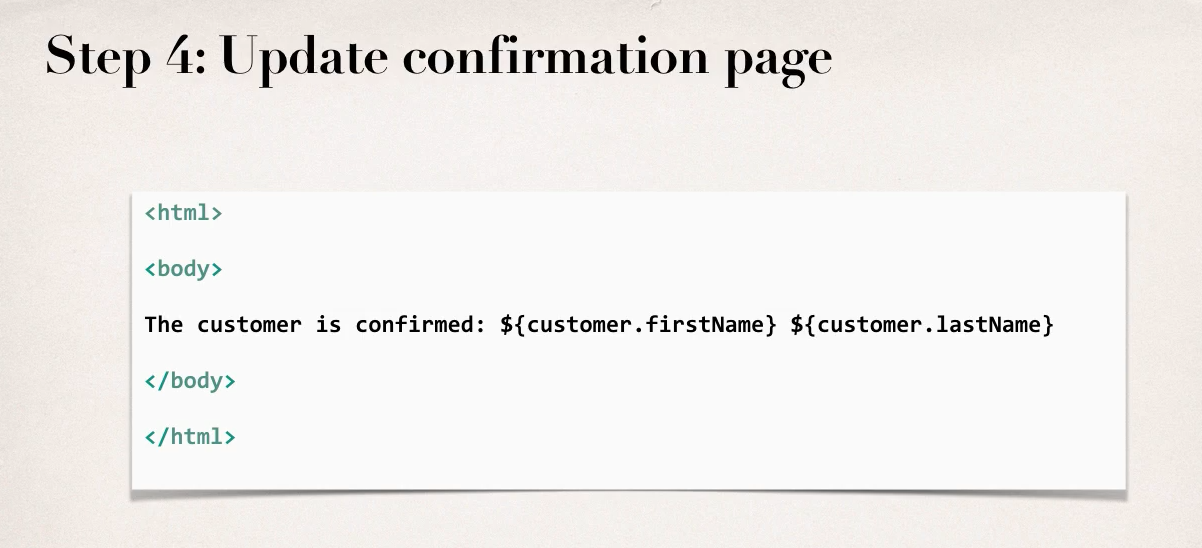
1. **Display Error message on HTML form**:The definition of error is on the form:errors tag.



1. **Perform Validation in the controller class:** By adding @Valid, we are adding validation to the customer class, and we are binding the result of the validation in the BindingResult object.

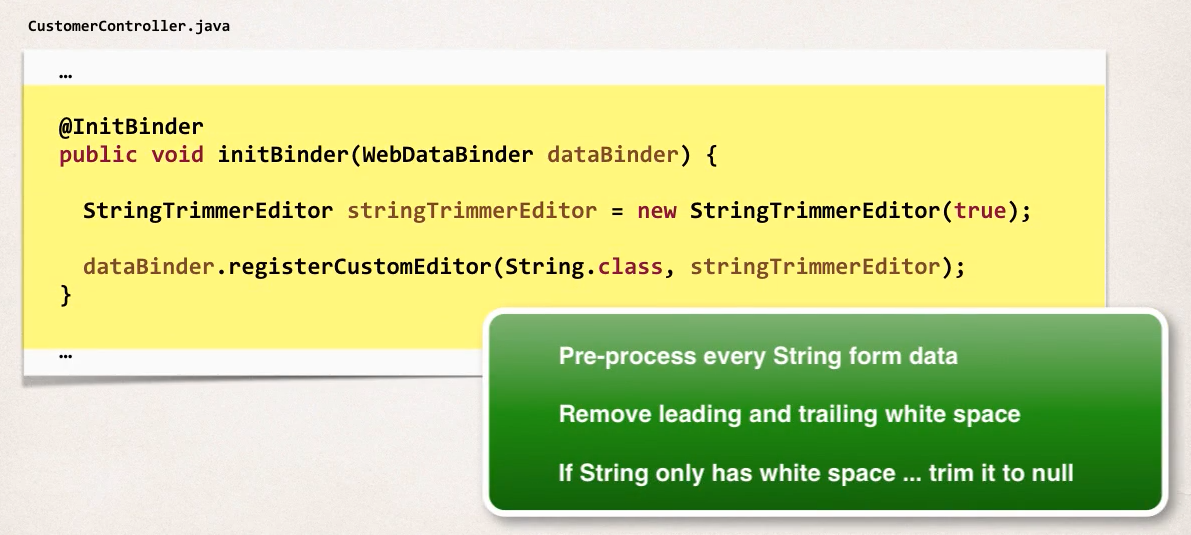


1. **Update confirmation Page:** Nothing special here, just showing the result.

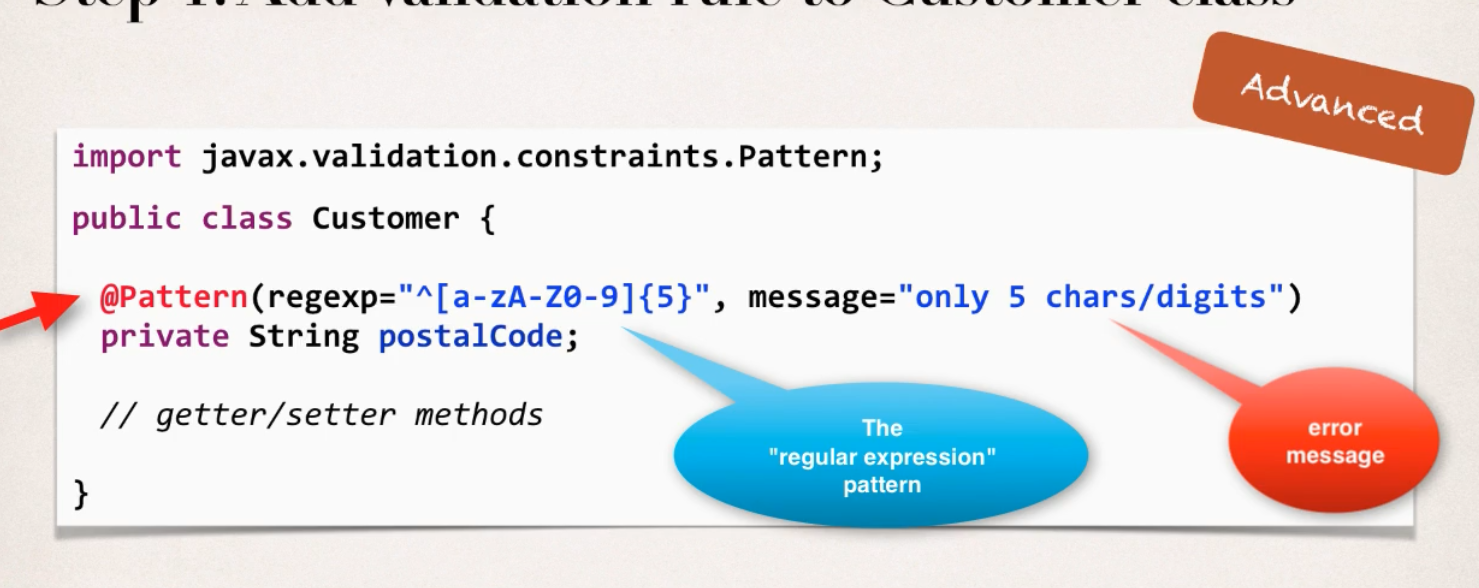


1. **Add pre-processor:** Even with all these validations, the whitespaces aren’t trimmed, so the validations won’t work as intended.

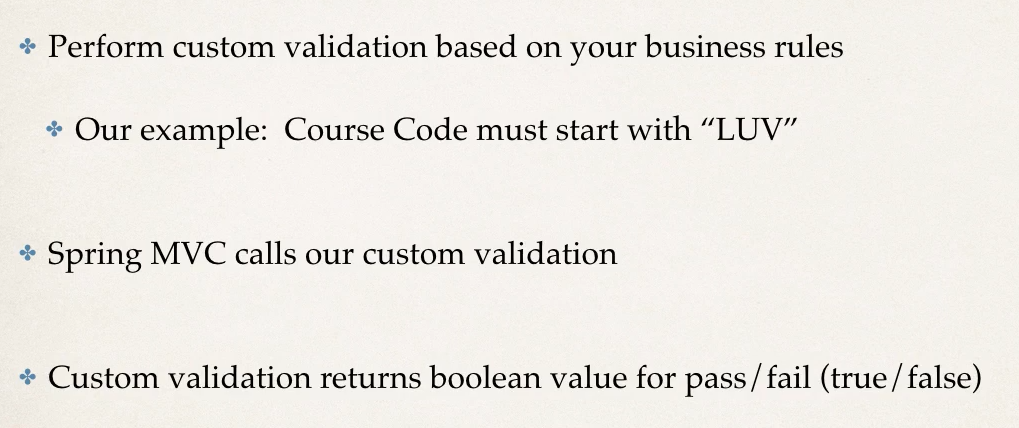
To solve this, we can use @**InitBinder** annotations in a method, this basically works like a middleware in node, each request will first go through this pre-processor method.



We can use regular expressions as well:



We can use our own custom validation rule:



For the rest watch the video.