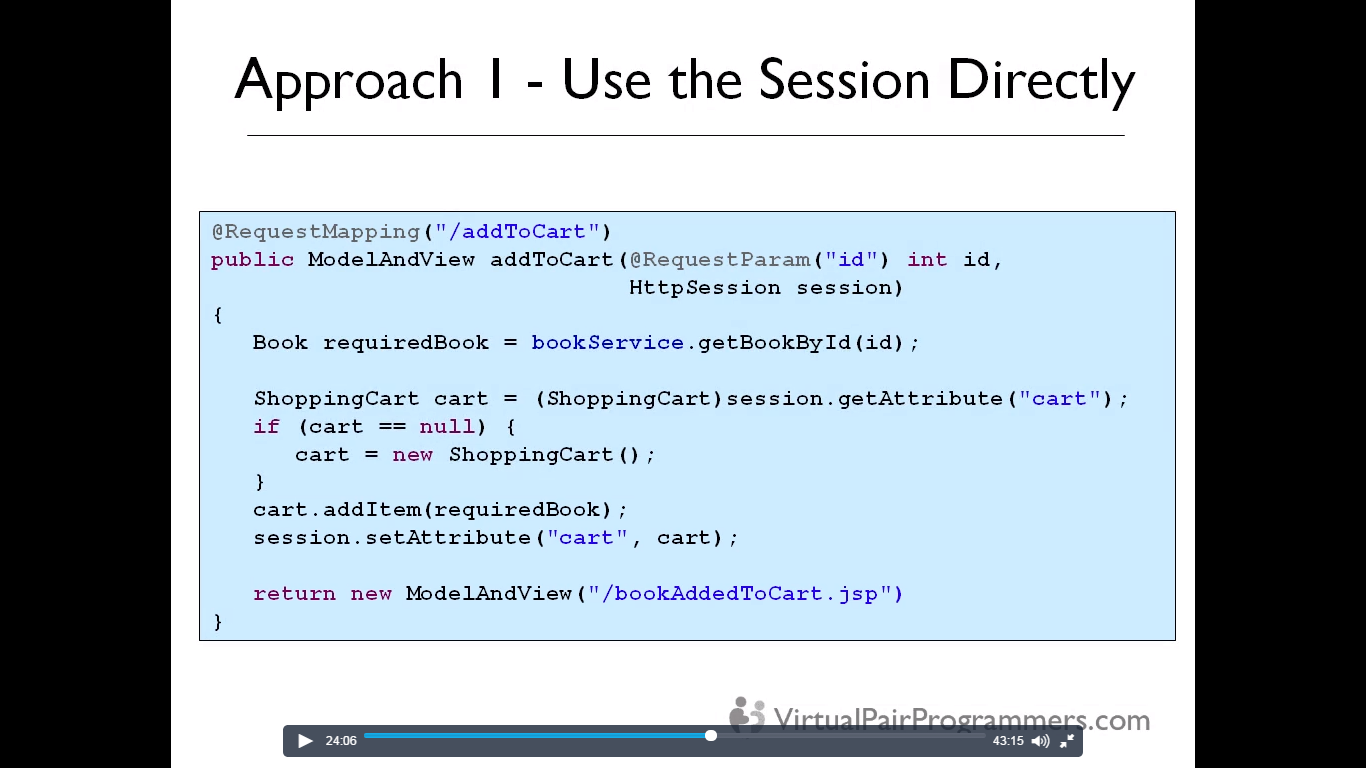
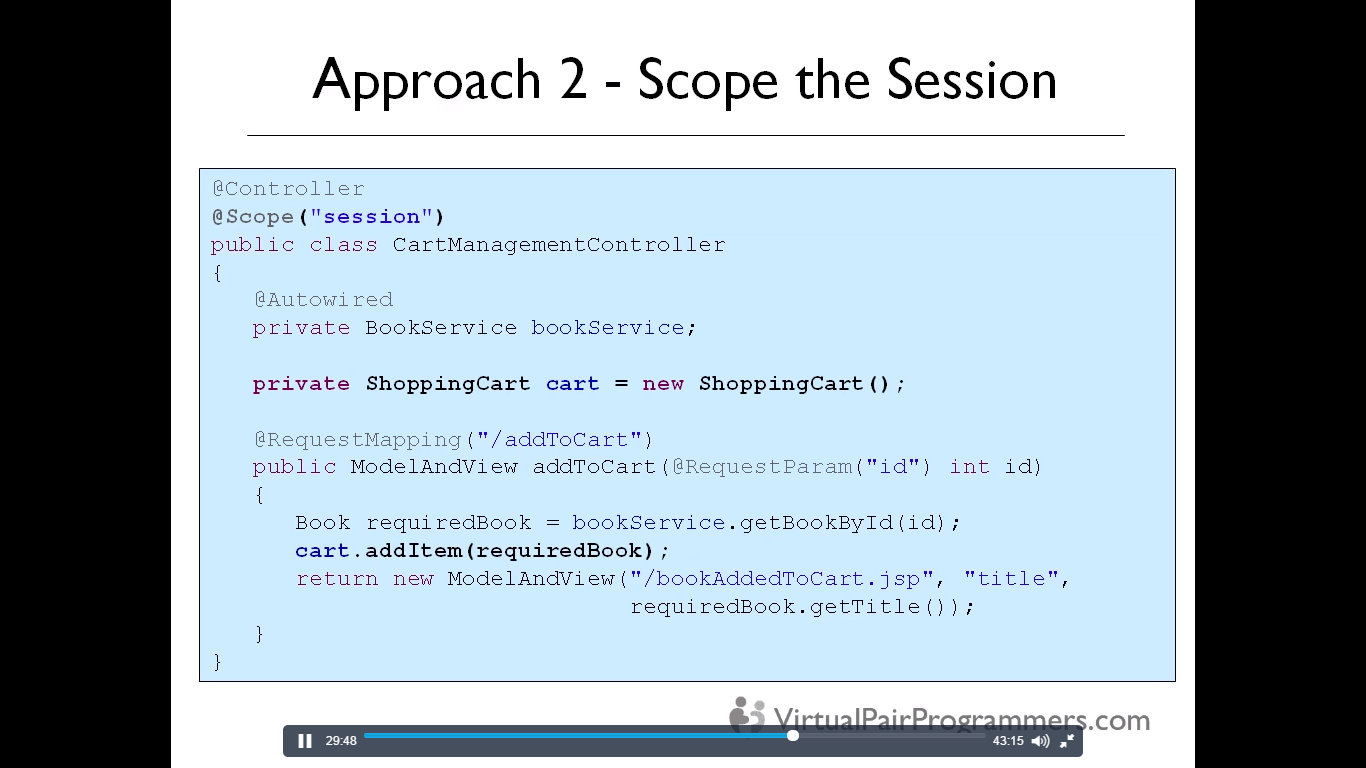
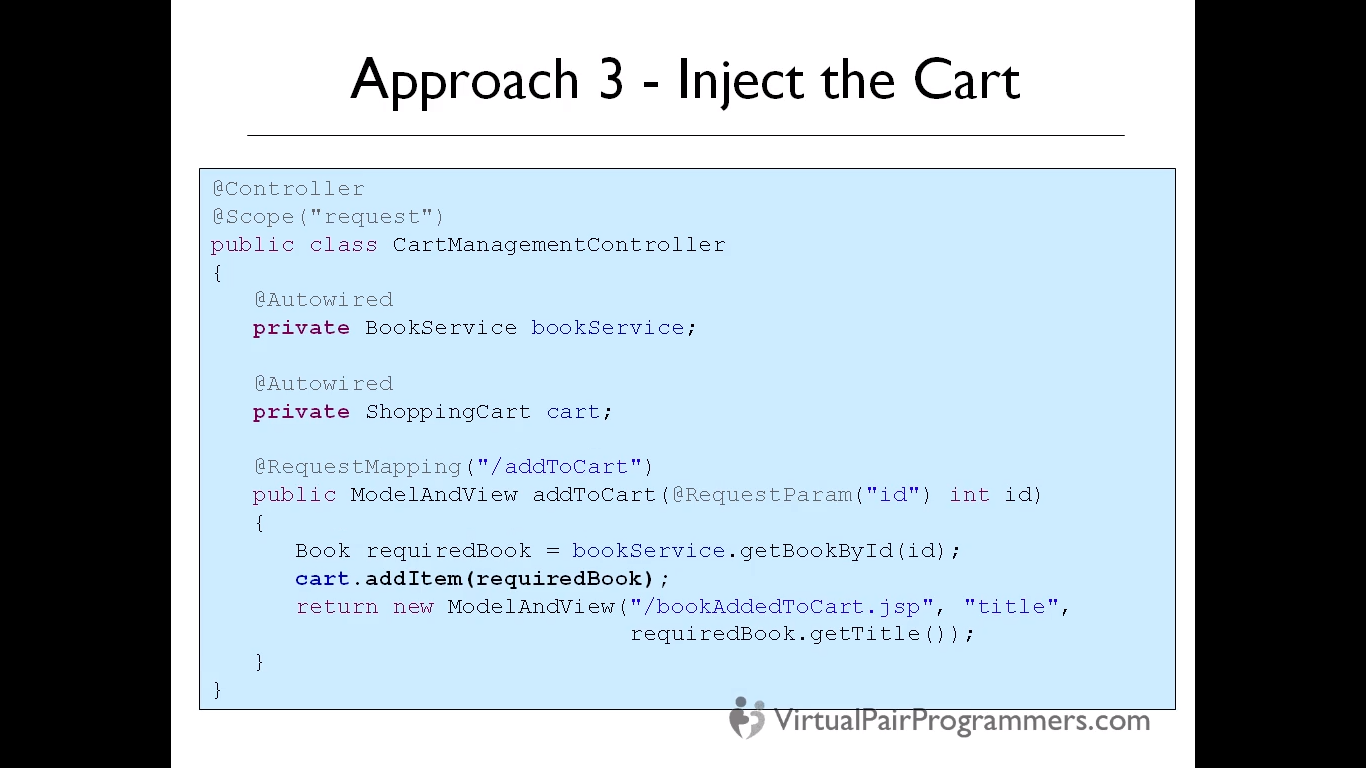
Session access in controller – First approach simple HttpSession object



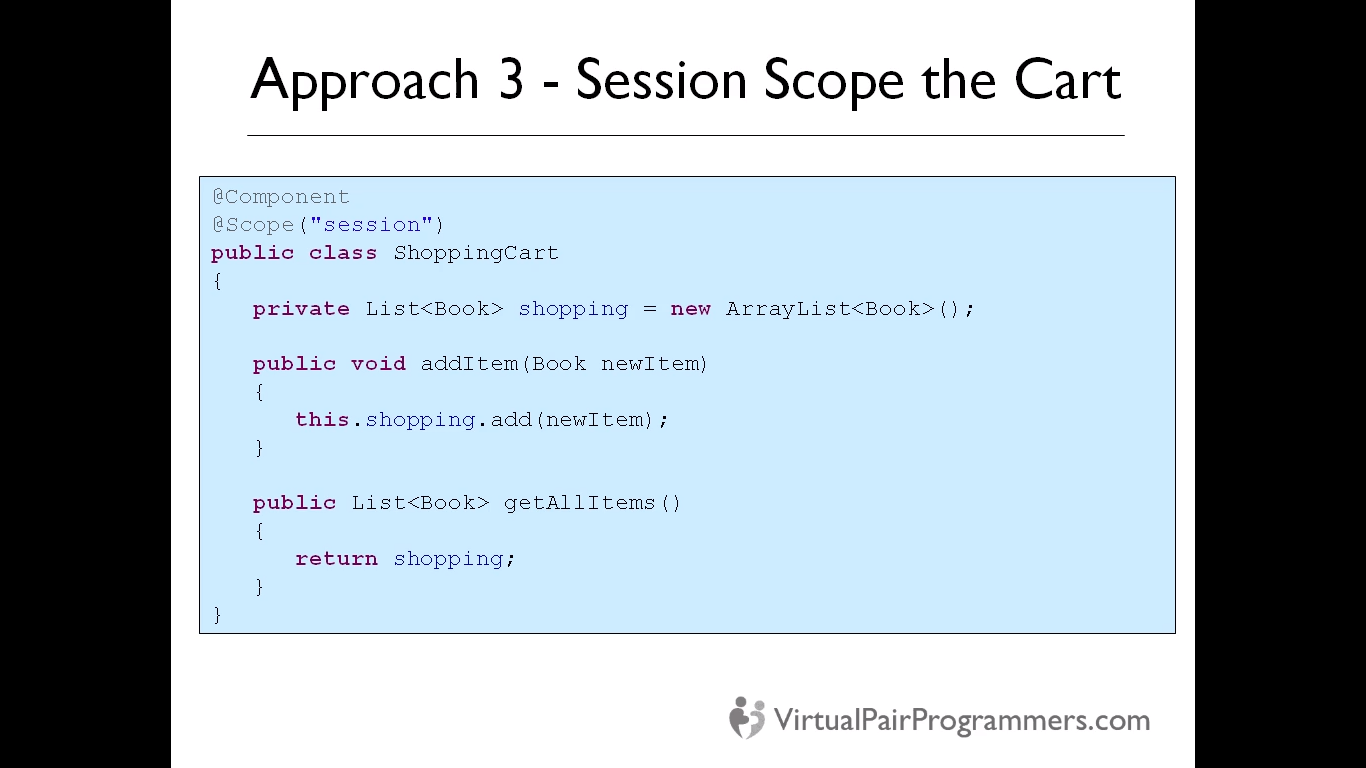
In the second approach, different controller will created for each individual user’s session rather than singleton controller which is by default spring beans. Shopping cart is added as private instance variable.



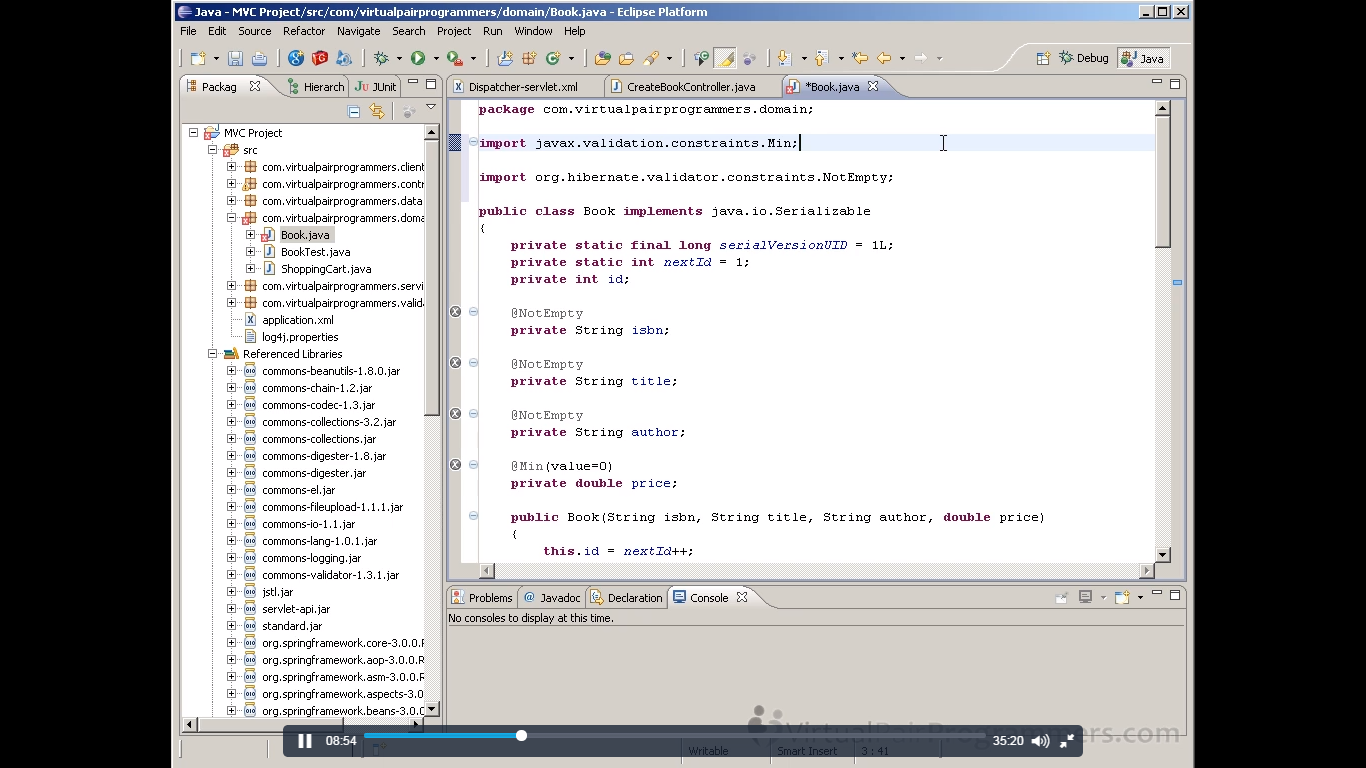
The drawback of this second approach is that the controller is stored in session HttpSession object and if controller is too big with lots of references and objects, it is not feasible.



Scope = request, the spring will create a new controller for each request from the user. The Shopping cart is autowired and ShoppingCart object is scope = session.

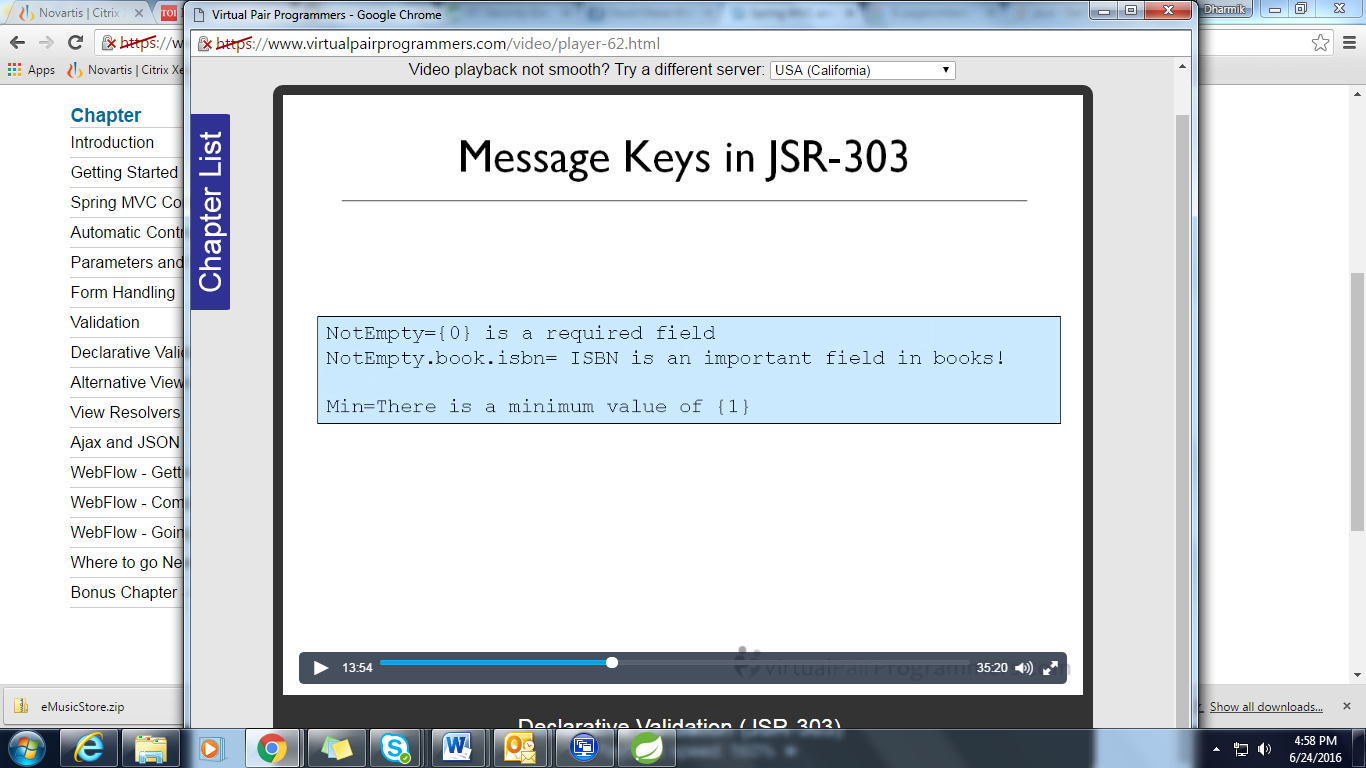


Shopping cart is now a spring bean @component annotation is used. This also has problem if controller is very heavy to instantiate as for each request it will be created.

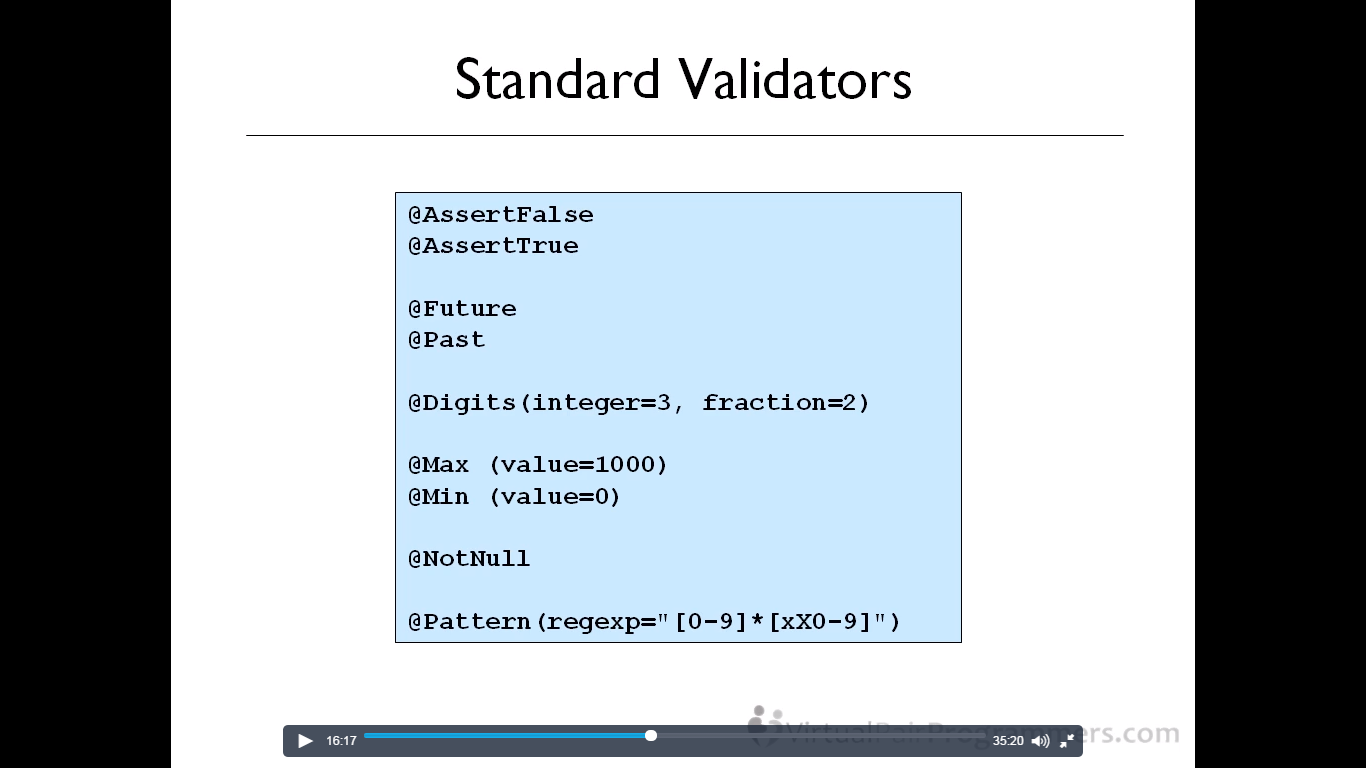


For activating validator, we need two jars. Validator jar from actual java and Hibernate validator which implements standards of jsr 303. For e.g @Min comes from javax.validator but @NotEmpty comes from hibernate validator and @NotNull is not effective in comparing empty strings.

<**mvc: annotation-driven** /> should be used to enable and use @Valid annotations.

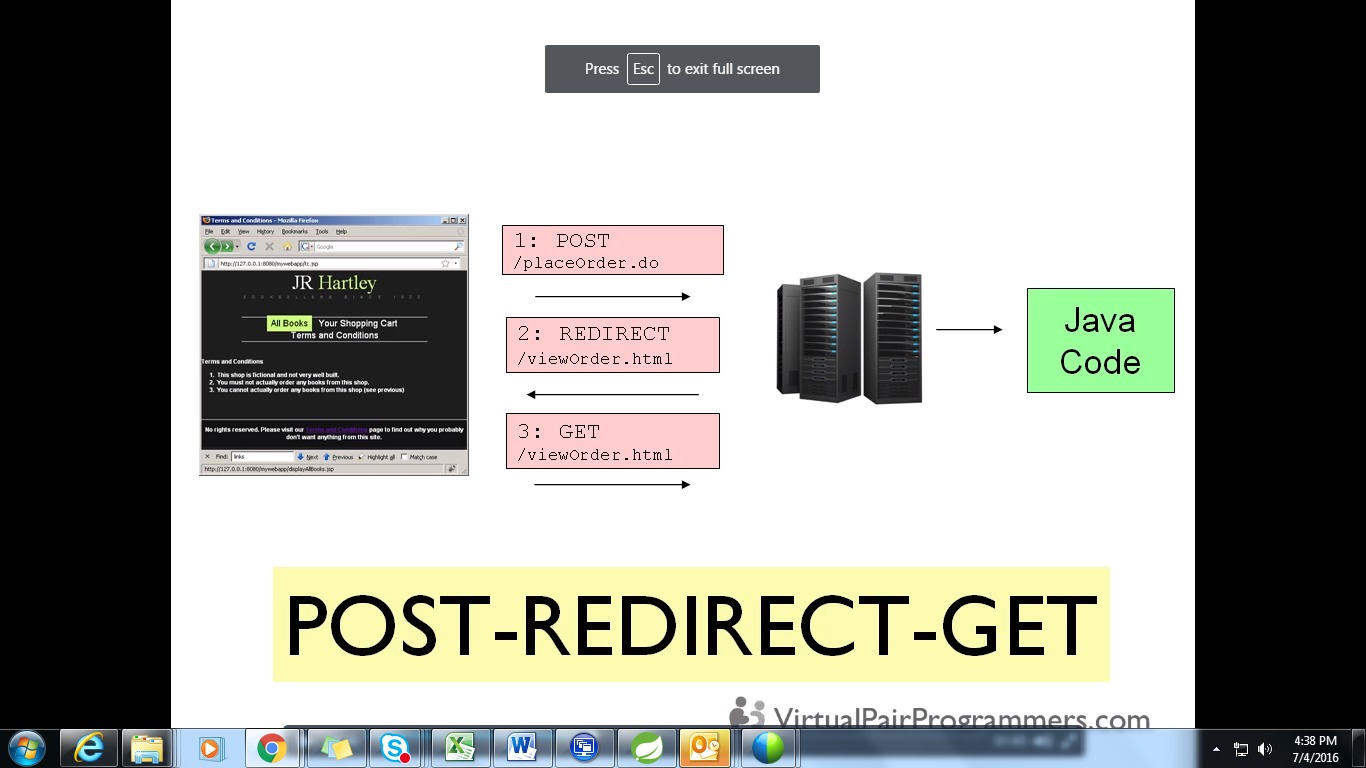


The messages will be picked up when we use above annotations. For {0} , attributes name will be passed and for {1} the min value will be passed from annotations.



@AssertFalse and true will be used if the value is true or false. @Future and @Past is used to test date with respect to current dates.

Webflow-



This method is used to prevent adverse effect from refreshing the browsers. Spring will redirect to different page after placing order. So again refreshing will be to fetch vieworder page only.