**JSR (Java Specification Requests) API AKA Bean Validation 2.0**

* **@NotNull** - Checks that the annotated value is not null. Unfortunately it doesn't check for empty string values
* **@Pattern** - Checks if the annotated string matches the regular expression given. We used it to ensure that the last name and first name properties have valid string values
* **@Past** and **@PastOrPresent** validate that a date value is in the past or the past including the present; can be applied to date types including those added in Java 8.
* **@Future**and**@FutureOrPresent** validate that a date value is in the future, or in the future including the present.
* **@Min** or **@DecimalMin**- The annotated element must be a number whose value must be greater or equal to the specified minimum
* **@Max** or **@DecimalMax** - The annotated element must be a number whose value must be lower or equal to the specified maximum
* **@NotBlank** - Checks that the annotated string is not null and the trimmed length is greater than 0. This annotation is not in JSR 303
* **@Email** - Checks whether the specified string is a valid email address. This annotation is also not in JSR 303
* **@Positive** and **@PositiveOrZero** apply to numeric values and validate that they are strictly positive, or positive including 0.
* **@Negative** and **@NegativeOrZero** apply to numeric values and validate that they are strictly negative, or negative including 0.
* **@NotEmpty** validates that the property is not null or empty; can be applied to String, Collection, Map or Array values.
* **@AssertTrue** validates that the annotated property value is true.
* **@Size**validates that the annotated property value has a size between the attributes min and max; can be applied to String, Collection, Map, and array properties.

They all take a message attribute:

@NotNull(message = "Last name is compulsory")

Some of them have other attributes, like:

@Min(value = 18, message = "Age must be greater than or equal to 18")

The most complicated one is the Pattern which takes a reg expression:

@Pattern(regexp=[.+@.+\\.[a-z]+](mailto:.+@.+\\.%5ba-z%5d+), message = “whatever”)

Size actually, has two (only works for strings):

@Size(min=2, max=240, message = “whatever you want to say”)

@Range is for numbers

https://docs.jboss.org/hibernate/validator/5.2/reference/en-US/html/ch02.html#table-custom-constraints