



CS544 EA

Applications

Validation

Validation Overview

- User input is often not reliable
 - Regardless of the mistake being honest / dishonest
 - Can be a real problem to system functionality
- Data Validation is critical for any serious system
 - Can be done by specifying constraints
 - Data is generally stored in Entities
 - Hibernate Validator has become a Java Standard

About Validation

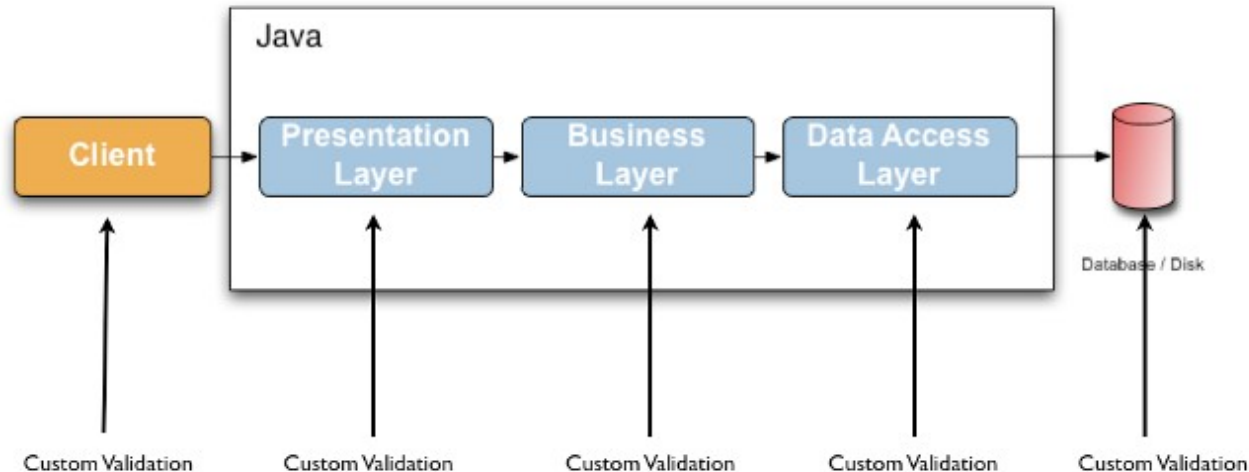
- Validation is the act of ensuring that your data is what you expect it to be
- Ensuring that the program operates on clean, correct and useful data
- Important for:
 - Consistency, reliability, security

Hibernate Bean Validation

- Validating data is a common task that occurs throughout the application layers
 - Presentation Layer (user input)
 - Business Layer (input from clients)
 - Data Access Layer (check before persisting)

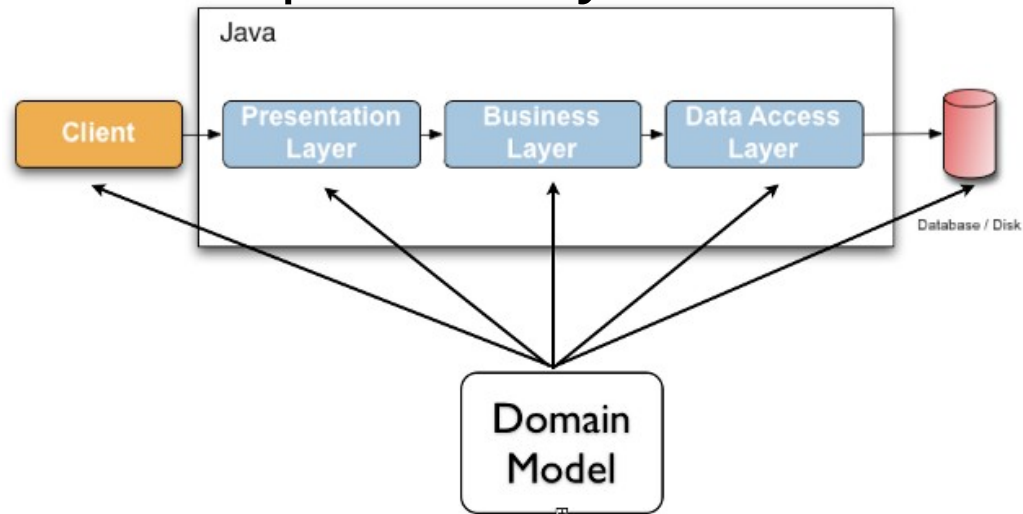
Validation Logic

- Same validation logic could be implemented in each layer
 - Time consuming
 - Error prone (violates DRY)



Validation in Domain

- Validation logic can be added to the domain
 - But code would clutter domain classes
 - Such code is really just metadata about each class
 - Metadata is best expressed by annotations!



Code Validation VS Annotations

Code

```
@Entity
public class Car {

    ...

    private int year;

    public void setYear(int year) {
        if (year > 1940 && year < 2015){
            this.year = year;
        } else {
            throw new IllegalArgumentException
                ("Invalid year for a Car");
        }
    }
}
```

Annotations

```
@Entity
public class Car {

    ...

    @Range(min = 1940, max = 2015)
    private int year;
}
```