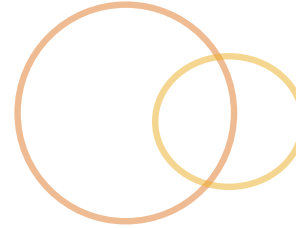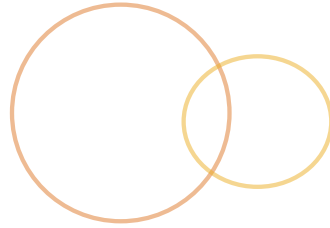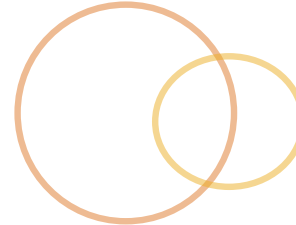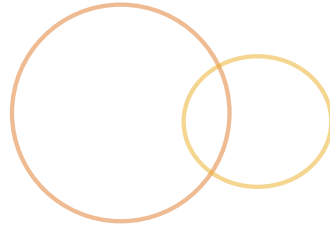# Validation Using JSR-303

Validation

- Three ways to do validation
  - Use the JSR-303 annotations
  - Write your own
  - A mix of the previous two
- Using JSR-303, you simply use one of the annotations provided
- For writing your own, you implement `ConstraintValidator` and you define a annotation for it
- Configure a `DataBinder` if you want more control and/or don't want to use annotations

JSR-303

- Spring 3.0 and above support JSR-303. This JSR is all about validation via annotation
- Primarily found in the `javax.validation.constraints`
- `@NotNull`
- `@Min(<minimum value allowed>`
- `@Max(<maximum value allowed>)`
- `@Future`
- `@Size`

JAR Dependencies

◉ Needs to be configured

  ◉ One part is the validation engine

  ◉ One part is the API to get the JSR-303 annotations

```xml
<dependency>
  <groupId>org.hibernate</groupId>
  <artifactId>hibernate-validator</artifactId>
  <version>5.0.1.Final</version>
</dependency>

<dependency>
  <groupId>javax.validation</groupId>
  <artifactId>validation-api</artifactId>
  <version>1.1.0.Final</version>
</dependency>
```

Preparing for Validation

◎ To use validation, add the following bean into your configuration file

```
@Bean
 public Validator setup303Validation(){
    return new LocalValidatorFactoryBean();
 }
```

◎ This turns on validation

- ◎ There is an expectation that you have a provider loaded
- ◎ Comes from `org.springframework.validation.beanvalidation`
- ◎ Implements what is needed for both Spring and Java's validation mechanisms

Modify Your Model Class

```java
public class Book {
  @NotNull
  @Pattern(regexp="^(97(8|9))?\d{9}(\d|X)$")
  private String ISBN;

  @NotNull
  @Size(min=5, max=24)
  private String author;

  @NotNull
  private String title;

  @Past
  private Date dueDate;
...
}
```

Modify Your Spring Bean

```java
public class LibraryServiceImpl implements LibraryService{
  private BookRepository repository;

  public LibraryServiceImpl(BookRepository repository){
    this repository = repository;
  }


  public void addBook(@Valid Book b) {
    return repository.addBook(b);
  }
}
```

Lab 8 – Validation