

Recap of Spring Concepts That Are Used in This Course



Bryan Hansen
DIRECTOR OF SOFTWARE DEVELOPMENT
@bh5k

A close-up photograph showing a group of people's hands stacked together in a circle. The hands belong to individuals of various ages and skin tones, suggesting a diverse community. The hands are positioned palm-up, with fingers interlaced. The background is blurred, showing what appears to be a colorful mural or artwork on a wall.

Spring

Types of Injection

Setter

Constructor

```
@Bean(name="customerService")
public CustomerService getCustomerService() {
    CustomerServiceImpl customerService = new CustomerServiceImpl();
    customerService.setCustomerRepository(getCustomerRepository());
    return customerService;
}
```

```
@Bean(name = "customerRepository")
public CustomerRepository getCustomerRepository() {
    return new HibernateCustomerRepositoryImpl();
}
```

Setter Injection

Simple as a method call

“Mystery” of injection goes away

Setter Injection simply calling a setter

Setter Pojo

```
public static void main(String [] args) {  
  
    CustomerRepository repository = new CustomerRepositoryImpl();  
  
    CustomerServiceImpl service = new CustomerServiceImpl();  
    service.setCustomerServiceRepository(repository);  
  
    //do something with the service now  
  
}
```

```
<bean id="customerRepository"
      class="com.pluralsight.conference.repository.CustomerRepositoryImpl"/>

<bean id="customerService"
      class="com.pluralsight.conference.service.CustomerServiceImpl">
    <property name="repository" value="customerRepository"/>
</bean>
```

XML and Spring Configuration

Bean for customerRepository

Bean for customerService

Property for “wiring” the repository into the service.

```
@Bean(name = customerRepository)  
...  
@Autowired  
public void setCustomerRepository(CustomerRepository  
customerRepository) {  
    this.customerRepository = customerRepository;  
}
```

Autowiring Setters

Bean defined as `customerRepository`

Setter autowired with `customerRepository` bean

Utilizes a component scanner

```
@Bean(name="customerService")
public CustomerService getCustomerService() {
    CustomerServiceImpl customerService =
        new CustomerServiceImpl(getCustomerRepository());
    return customerService;
}
```

```
@Bean(name = "customerRepository")
public CustomerRepository getCustomerRepository() {
    return new HibernateCustomerRepositoryImpl();
}
```

Constructor Injection

Constructor with args in order that you want

```
<bean id="customerRepository"
      class="com.pluralsight.conference.repository.CustomerRepositoryImpl"/>

<bean id="customerService"
      class="com.pluralsight.conference.service.CustomerServiceI
mpl">
    <constructor-arg index="0" value="customerRepository"/>
</bean>
```

Spring XML Config Constructor Injection

Switch from property to constructor arg

```
@Bean(name = customerRepository)  
...  
@Autowired  
public CustomerServiceImpl(CustomerRepository  
customerRepository) {  
    this.customerRepository = customerRepository;  
}
```

Autowiring Constructors

Almost identical to autowiring setters

Context Files



Spring Configuration
XML
Annotations
Java Config
Now AppConfig!!!

Summary

Setter Injection
Constructor Injection
POJOs
XML Configuration
Autowiring
Context Files