# Spring Configuration Using Java





**Bryan Hansen**Director of Software Development



## No applicationContext.xml

Too much XML

Namespaces helped

**Java Configuration** 

**Everything available** 

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

### @Configuration

applicationContext.xml replaced by @Configuration @Configuration at class level Spring Beans defined by @Bean @Bean at method level

```
@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**

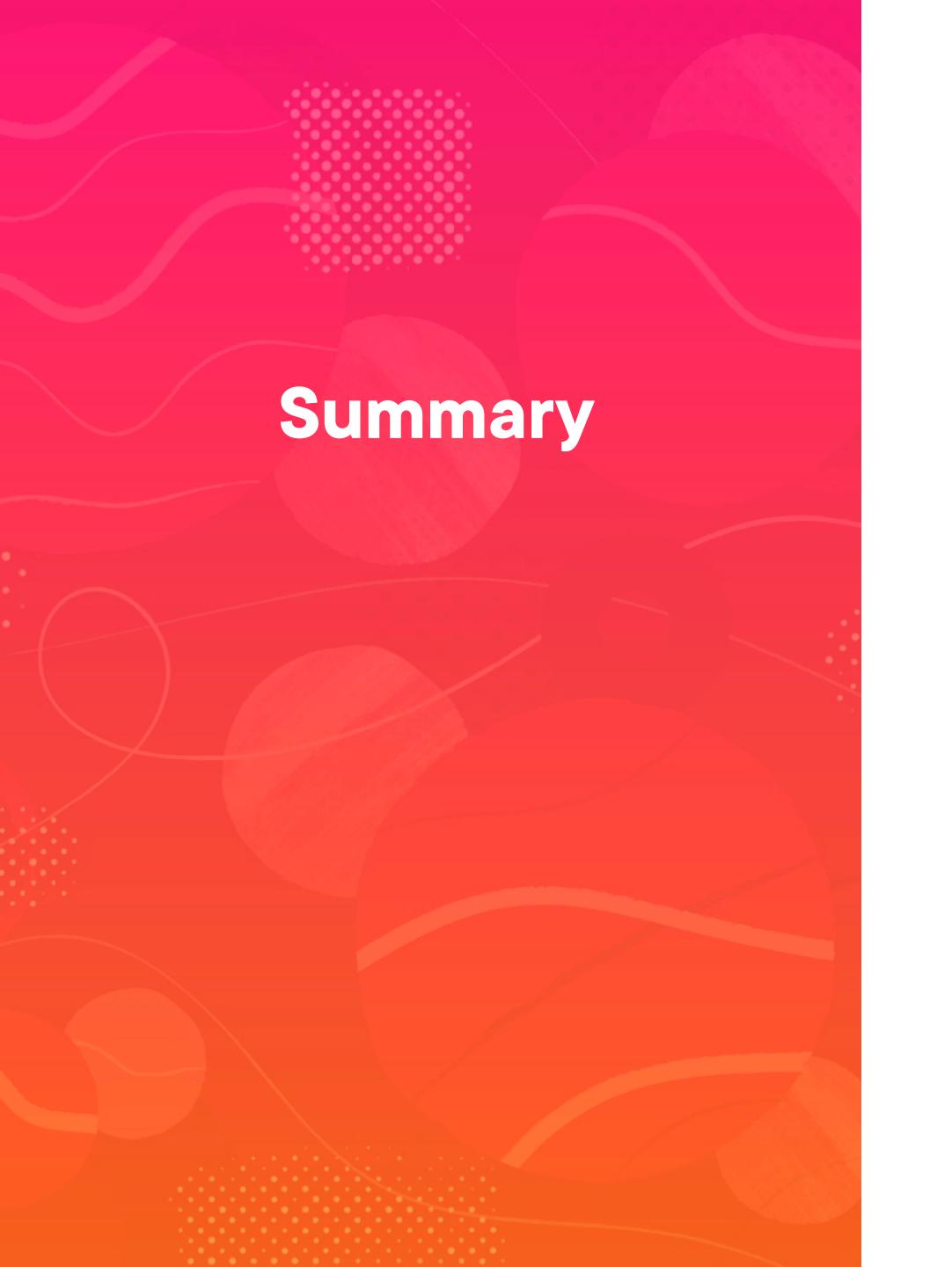
Simple as a method call "Mystery" of injection goes away Simply calling a setter

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

### **Constructor Injection**

Just like setter injection

Becoming the preferred way



AppConfig

@Configuration

@Bean

Setter and Constructor Injection