

Spring Configuration with Annotations



What are Java Annotations?

- Special labels/markers added to Java classes
- Provide meta-data about the class
- Processed at compile time or run-time for special processing

Boot
Color: Silver
Style: Jewel
Code: 1460
SKU: 10072090
Size US: 8
Size UK: 6

Annotation Example

- We've seen annotations already ...

```
public class TrackCoach implements Coach {  
  
    @Override  
    public String getDailyWorkout() {  
        return "Run a hard 5k";  
    }  
    ...  
}
```

Why Spring Configuration with Annotations?

- XML configuration can be verbose
- Configure your Spring beans with Annotations
- Annotations minimizes the XML configuration

Scanning for Component Classes

- Spring will scan your Java classes for special annotations
- Automatically register the beans in the Spring container

Development Process

1. Enable component scanning in Spring config file
2. Add the @Component Annotation to your Java classes
3. Retrieve bean from Spring container

Step-By-Step

Step 1: Enable component scanning in Spring config file

```
<beans ...>  
  
    <context:component-scan base-package="com.luv2code.springdemo" />  
  
</beans>
```

Step 2: Add the @Component Annotation to your Java classes

```
@Component("thatSillyCoach")
public class TennisCoach implements Coach {

    @Override
    public String getDailyWorkout() {
        return "Practice your backhand volley";
    }

}
```

Step 3: Retrieve bean from Spring container

- Same coding as before ... nothing changes.

```
Coach theCoach = context.getBean("thatSillyCoach", Coach.class);
```