



## Spring Core – Annotations

Presented by



# Annotations

Annotations are instructions to the compiler

Meta data / data about code

@Override

Checks that the method is an override.

Ex:

```
@Override
public String toString() {
    return "Welcome !";
}
```

@SuppressWarnings

Instructs the compiler to suppress the [compile time](#)



# @Autowired

**@Autowired** annotation is used to inject a bean automatically without using **ref** attribute

Normal configuration:

```
<bean id="emp" class="Employee">
    <property name="firstname" value="Kumar">
    <property name="address" ref="address">
</bean>
<bean id="address" class="Address">
    <property name="street" value="Road No. 1">
</bean>
```

# @Autowired...

## With @Autowired

```
<bean id="emp" class="Employee">  
  <property name="firstname" value="Kumar">  
</bean>
```

```
<bean id="address" class="Address">  
  <property name="street" value="Road No. 1">  
</bean>
```

Employee class

```
@Autowired  
public void setAddress(Address address) {  
  this.address = address;  
}
```

**\*\* Example**

# @Component

**@Component** annotation marks a java class as a bean

So spring container can pick it up and pull it into the application

Called as component-scanning .

**No need of configuring it**

Ex :

```
@Component  
public class Employee {  
}
```

**\*\* Example**

# @Qualifier

## @Qualifier

To Select a specific component bean of same type with autowired dependency

```
<bean id="profile" class="Profile">  
</bean>
```

In the Profile class :

```
@Autowired  
@Qualifier("student1")
```

```
<!-- Definition for student1 bean -->  
<bean id="student1" class="Student">  
  <property name="name" value="Ram" />  
  <property name="age" value="11"/>  
</bean>
```

```
<!-- Definition for student2 bean -->  
<bean id="student2" class="Student">  
  <property name="name" value="Syam" />  
  <property name="age" value="2"/>  
</bean>
```

\*\*Example

