

# Defining Beans with Java Code



# Defining Beans in Spring

# Defining Beans in Spring

## Full XML Config

```
<!-- define the dependency -->
<bean id="myFortuneService"
      class="com.Luv2code.springdemo.HappyFortuneService">
</bean>

<bean id="myCoach"
      class="com.Luv2code.springdemo.TrackCoach">

    <!-- set up constructor injection -->
    <constructor-arg ref="myFortuneService" />
</bean>

<bean id="myCricketCoach"
      class="com.Luv2code.springdemo.CricketCoach">

    <!-- set up setter injection -->
    <property name="fortuneService" ref="myFortuneService" />
```

# Defining Beans in Spring

## Full XML Config

```
<!-- define the dependency -->
<bean id="myFortuneService"
      class="com.Luv2code.springdemo.HappyFortuneService">
</bean>

<bean id="myCoach"
      class="com.Luv2code.springdemo.TrackCoach">

    <!-- set up constructor injection -->
    <constructor-arg ref="myFortuneService" />
</bean>

<bean id="myCricketCoach"
      class="com.Luv2code.springdemo.CricketCoach">

    <!-- set up setter injection -->
    <property name="fortuneService" ref="myFortuneService" />

```

## Java Configuration Class

```
package com.luv2code.springdemo;

import org.springframework.context.annotation.ComponentScan;
import org.springframework.context.annotation.Configuration;

@Configuration
public class SportConfig {
```

# Defining Beans in Spring

## Full XML Config

```
<!-- define the dependency -->
<bean id="myFortuneService"
      class="com.Luv2code.springdemo.HappyFortuneService">
</bean>

<bean id="myCoach"
      class="com.Luv2code.springdemo.TrackCoach">

    <!-- set up constructor injection -->
    <constructor-arg ref="myFortuneService" />
</bean>

<bean id="myCricketCoach"
      class="com.Luv2code.springdemo.CricketCoach">

    <!-- set up setter injection -->
    <property name="fortuneService" ref="myFortuneService" />

```

## Java Configuration Class

```
package com.luv2code.springdemo;

import org.springframework.context.annotation.ComponentScan;
import org.springframework.context.annotation.Configuration;

@Configuration
public class SportConfig {
```

No XML!

# Our New Coach ...

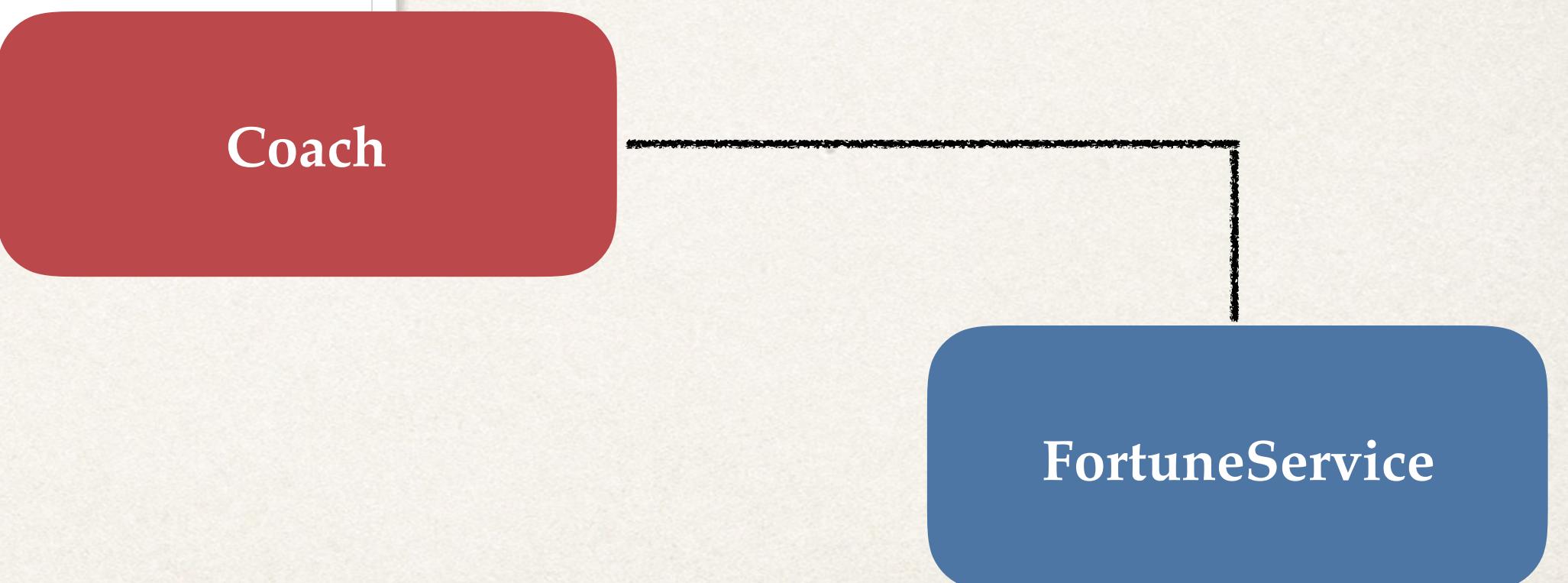
```
public class SwimCoach implements Coach {  
    ...  
}
```

*No special  
annotations*

# Our New Coach ...

```
public class SwimCoach implements Coach {  
    ...  
}
```

No special  
annotations



# Development Process

*Step-By-Step*

# Development Process

*Step-By-Step*

1. Define method to expose bean

# Development Process

*Step-By-Step*

1. Define method to expose bean
2. Inject bean dependencies

# Development Process

*Step-By-Step*

1. Define method to expose bean
2. Inject bean dependencies
3. Read Spring Java configuration class

# Development Process

*Step-By-Step*

1. Define method to expose bean
2. Inject bean dependencies
3. Read Spring Java configuration class
4. Retrieve bean from Spring container

# Step 1: Define method to expose bean

```
@Configuration  
public class SportConfig {  
  
    @Bean  
    public Coach swimCoach() {  
        SwimCoach mySwimCoach = new SwimCoach();  
  
        return mySwimCoach;  
    }  
  
}
```

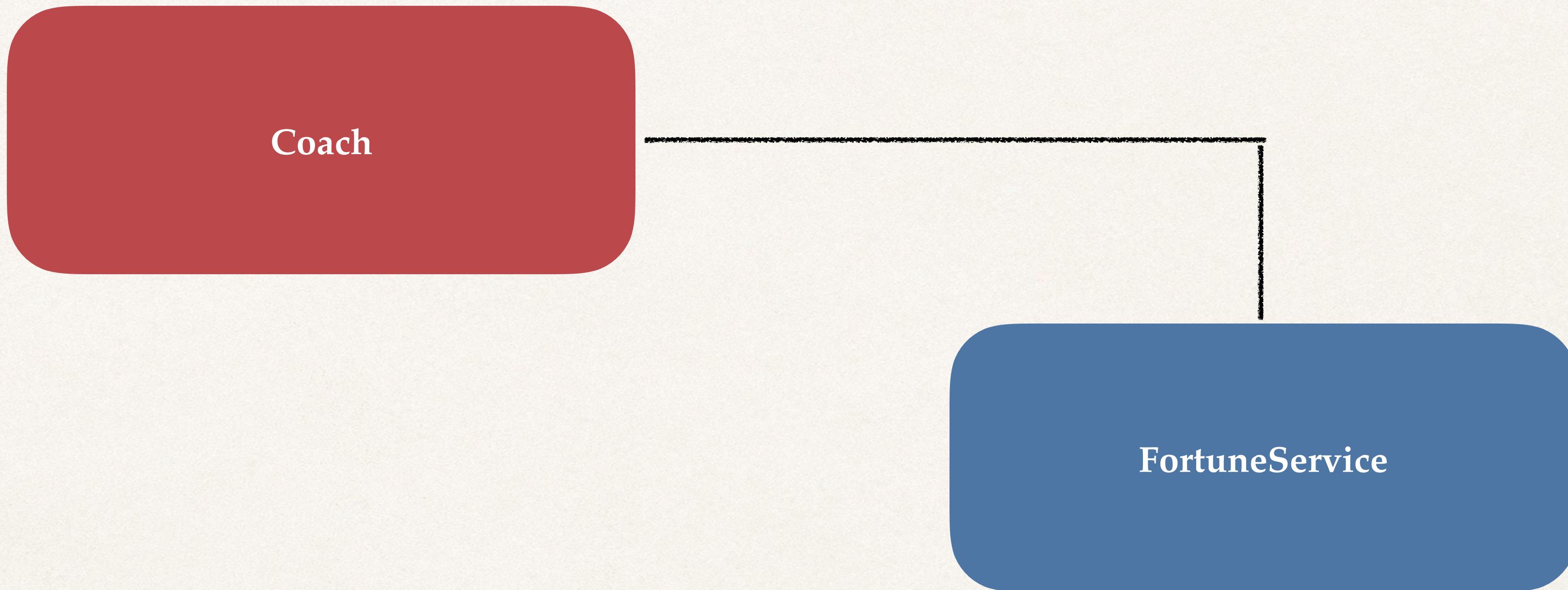
# Step 1: Define method to expose bean

```
@Configuration  
public class SportConfig {  
  
    @Bean  
    public Coach swimCoach() {  
        SwimCoach mySwimCoach = new SwimCoach();  
  
        return mySwimCoach;  
    }  
  
}
```

No component scan

# What about our dependencies?

# What about our dependencies?



# Step 2: Inject bean dependencies

```
@Configuration  
public class SportConfig {  
  
    @Bean  
    public FortuneService happyFortuneService() {  
        return new HappyFortuneService();  
    }  
  
    @Bean  
    public Coach swimCoach() {  
        SwimCoach mySwimCoach = new SwimCoach( happyFortuneService() );  
  
        return mySwimCoach;  
    }  
}
```

# Step 3: Read Spring Java configuration class

```
AnnotationConfigApplicationContext context =  
    new AnnotationConfigApplicationContext(SportConfig.class);
```

# Step 4: Retrieve bean from Spring container

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

# What about our dependencies?

# What about our dependencies?

