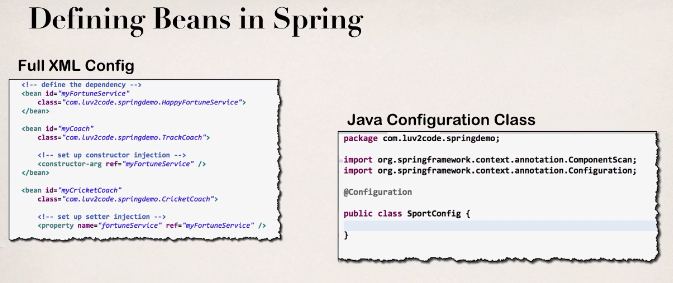
**Defining Spring Beans with Java Code**

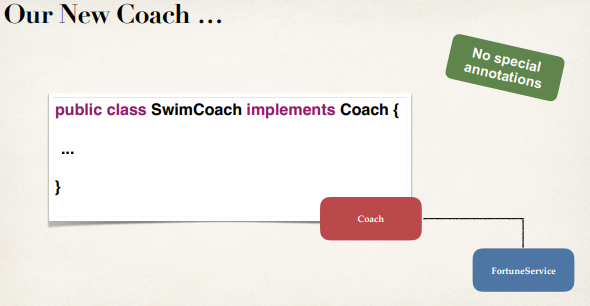
**Defining Beans in Spring**:

Now we define beans with Java code. We can also define beans by using XML. We will list each bean out separately. Here we define beans using Java configuration classes.



**Our new Coach**:

Now we introduce here our new coach



We have a Coach and we also have a FortuneService implementation. Here we will learn how to inject that FortuneService using just raw Java code.

**Development Process**:

1. Define method to expose bean
2. Inject bean dependencies
3. Read Spring Java configuration class (in our main program)
4. Retrieve bean from Spring container

**Step 1: Define method to expose bean**:

In our configuration class SportConfig, we use a new annotation call @Bean. This basically defines a bean. Here we will provide a method called “swimCoach()”. Internally inside of this method we actually create a new instance of this bean. After that return that instance.

The name of this method here “swimCoach”, that’s the actual bean ID that will be registered by the Spring container.

In this example we are not using component scan ((@ComponentScan). In this configuration we are defining each bean individually within this configuration class.

@Configuration

public class SportConfig {

@Bean

public Coach swimCoach() {

SwimCoach mySwimCoach = new SwimCoach();

return mySwimCoach;

}

}

**Note**:

No component scan (@ComponentScan)

**Step 2: Inject bean dependencies**:

Here we actually inject the bean dependencies, we make use the @Bean, and we define the bean here HappyFortuneService.

@Configuration

public class SportConfig {

@Bean

public FortuneService happyFortuneService() {

return new HappyFortuneService();

}

@Bean

public Coach swimCoach(FortuneService fortuneService) {

SwimCoach mySwimCoach = new SwimCoach( happyFortuneService() );

return mySwimCoach;

}

}

**Step 3: Read Spring Java configuration class**:

AnnotationConfigApplicationContext context = new

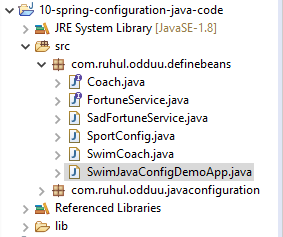
AnnotationConfigApplicationContext(SportConfig.class);

**Step 4: Retrieve bean from Spring container**:

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

**Example**:

**Project Structure**:



**File: Coach.java**:

**package** com.ruhul.odduu.definebeans;

**public** **interface** Coach {

**public** String getDailyWorout();

**public** String getDailyFortune();

}

**File: FortuneService.java**:

**package** com.ruhul.odduu.definebeans;

**public** **interface** FortuneService {

**public** String getFortune();

}

**File: SadFortuneService.java**:

**package** com.ruhul.odduu.definebeans;

**public** **class** SadFortuneService **implements** FortuneService {

@Override

**public** String getFortune() {

**return** "Today is a sad day";

}

}

**File: SwimCoach.java**:

**package** com.ruhul.odduu.definebeans;

**public** **class** SwimCoach **implements** Coach {

**private** FortuneService fortuneService;

// constructor

**public** SwimCoach(FortuneService fortuneService) {

**this**.fortuneService = fortuneService;

}

@Override

**public** String getDailyWorout() {

**return** "Swim 1000 meters as a warm up";

}

@Override

**public** String getDailyFortune() {

**return** fortuneService.getFortune();

}

}

**File: SportConfig.java**:

**package** com.ruhul.odduu.definebeans;

**import** org.springframework.context.annotation.Bean;

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

@Configuration

**public** **class** SportConfig {

// define bean for our SadFortuneService

// the method name("sadFortuneService") will be the bean ID

@Bean

**public** FortuneService sadFortuneService() {

**return** **new** SadFortuneService();

}

// define bean for our SwimCoach and inject dependency

@Bean

**public** Coach swimCoach() {

**return** **new** SwimCoach(sadFortuneService());

}

}

**File: SwimJavaConfigDemoApp.java**:

**package** com.ruhul.odduu.definebeans;

**import** org.springframework.context.annotation.AnnotationConfigApplicationContext;

**public** **class** SwimJavaConfigDemoApp {

**public** **static** **void** main(String[] args) {

// read spring configuration java class

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

// retrieve bean from spring container

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

// call a method on the bean

System.***out***.println(theCoach.getDailyWorout());

// call a method to the daily fortune

System.***out***.println(theCoach.getDailyFortune());

// close the context

context.close();

}

}

/\*

**Output**:

Swim 1000 meters as a warm up

Today is a sad day

\*/

Defining Spring Beans with Java Code