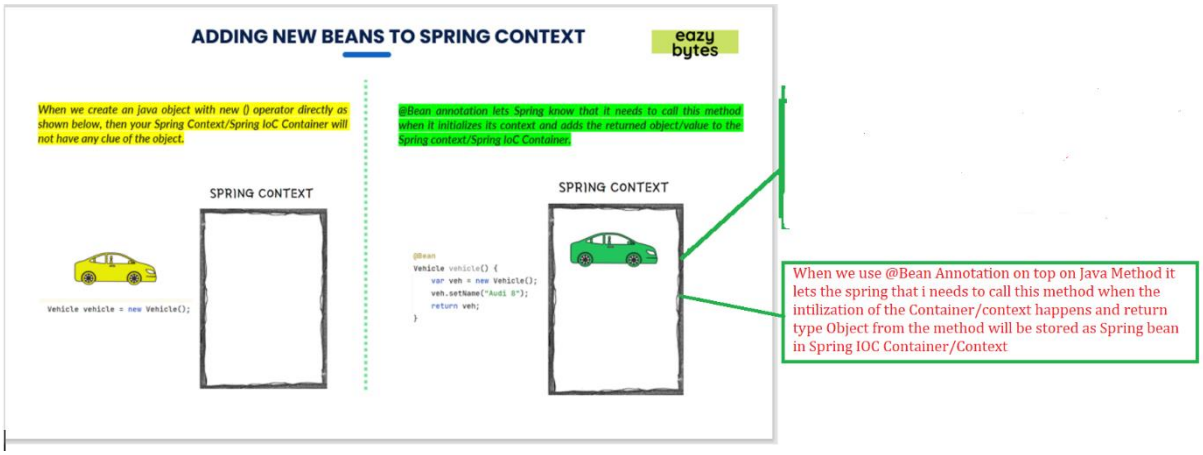
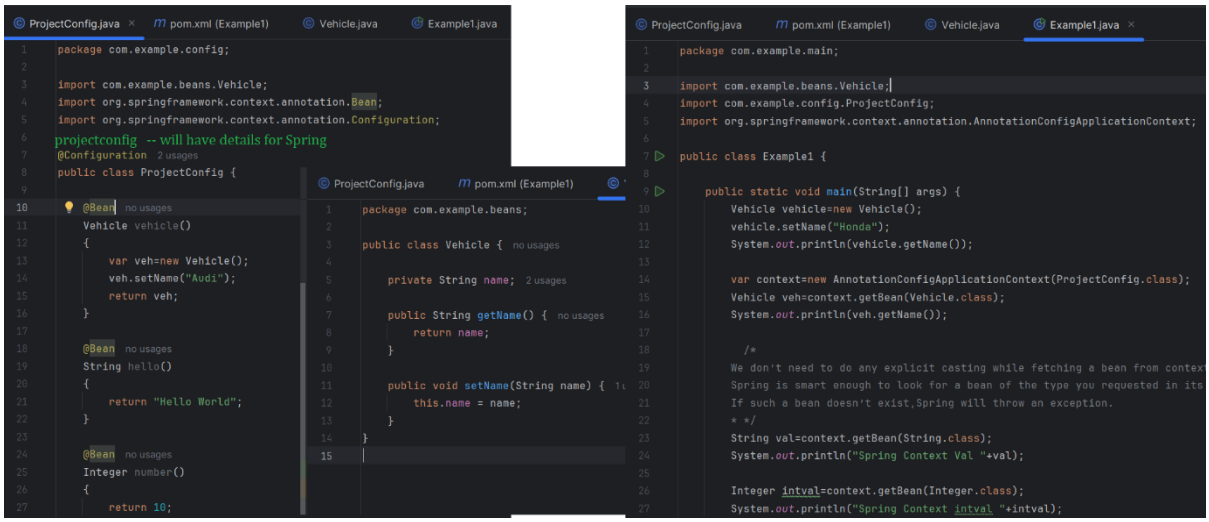


@Configuration

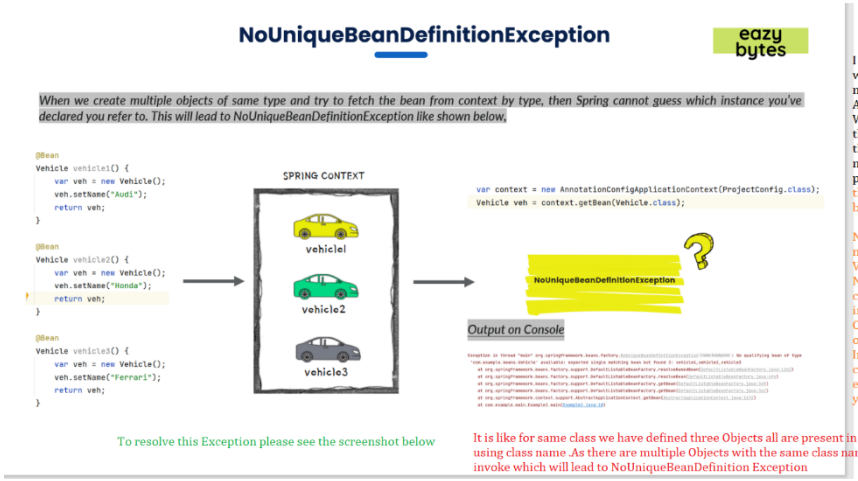
```
/*
Spring @Configuration annotation is part of the spring core framework.
Spring Configuration annotation indicates that the class has @Bean definition
methods. So Spring container can process the class and generate Spring Beans
to be used in the application.
* */
no usages
```



Configuration files will have details which class has to considered as Bean and few other details.

The name of the method will be considered as bean name.

16.Understanding NoUniqueBeanDefinition in Spring

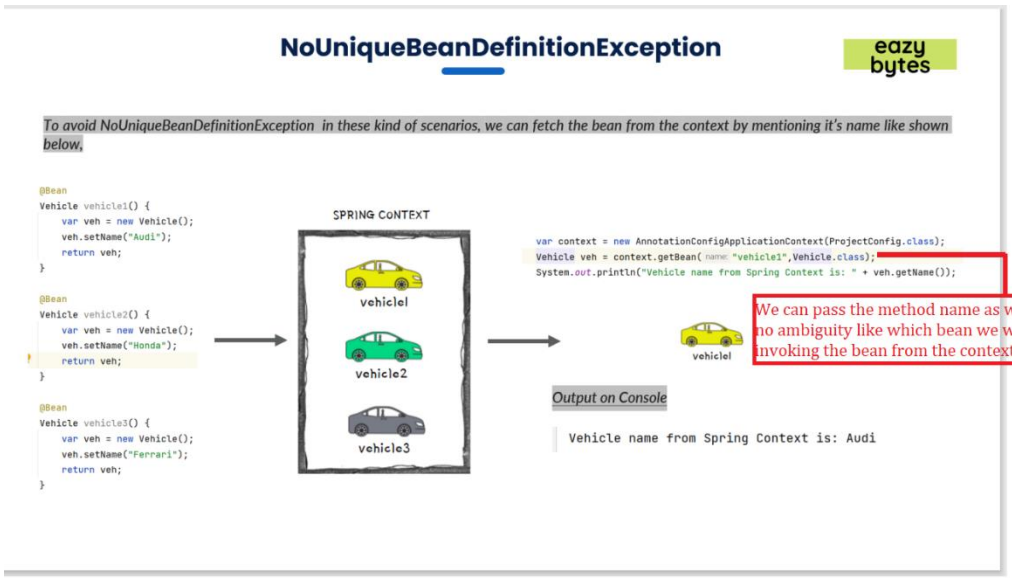


I have created three beans of same data type, which is vehicle with the help of three different methods and inside these three methods I have set different different names for my vehicle like Audi, Honda and Ferrari.
With these being declaration, my spring context will load all those beans during startup and you can see all these beans with the name vehicle one, vehicle two and vehicle three is being maintained by the spring IOC container. Like we are discussing previously,whatever method name that you have mentioned, the same name will be considered as a name of your spring bean.

Now when we try to get the details of the bean using get bean method, you are passing the data type of the bean, which is Vehicle(Name of the Class).
Now at this point of time, my spring has an ambiguity or confusion because there are three beans available inside the spring context with the same data type(With Same ClassName) and the developer is not providing any details other than data type of the bean.
In these kind of scenarios, my spring IOC container will have a confusion and with that confusion it is going to throw an exception saying that NoUniqueBeanDefinition exception and you will get an exception

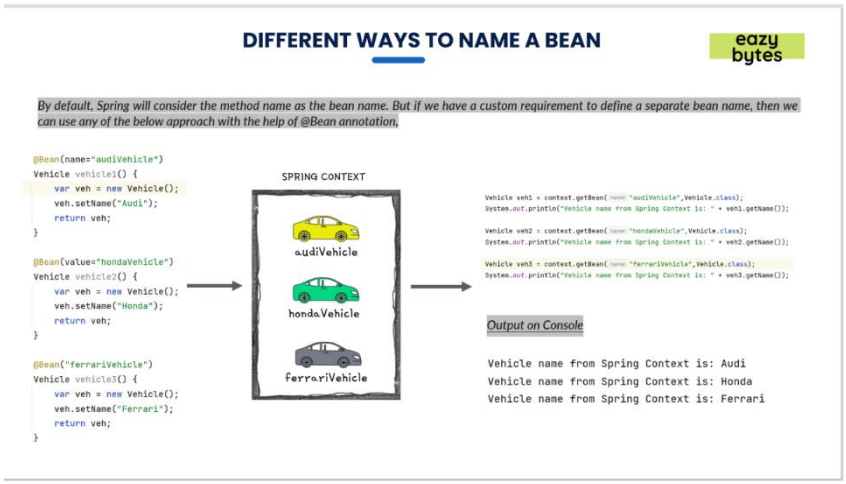
It is like for same class we have defined three Objects all are present in the context and when trying to get them we are getting using class name .As there are multiple Objects with the same class name Spring will have ambiguity which Object/bean to invoke which will lead to NoUniqueBeanDefinition Exception

No qualifying bean of type 'com.example.beans.Vehicle' available: expected single matching bean but found 3: vehicle1,vehicle2,vehicle3



By default whatever is the Java Method that name will be used as bean name.

17.Providing Custom Name to Bean :

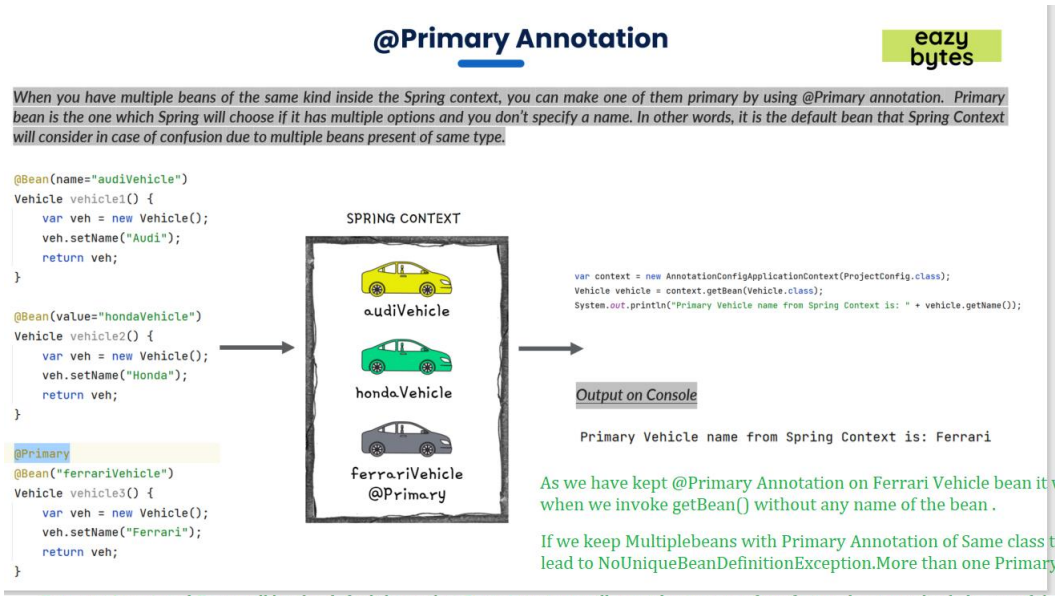


As we are having issue when we are trying to fetch with the class Type as for multiple class type there were too many multiple returns.We can go by passing the name in the getBean method or

else can Name the bean in the below way.

@Bean(name="Name of the Bean1" ----This name will be used as name for that Bean.
@Bean(value="Name of bean")
@Bean("Name of Bean"

18.

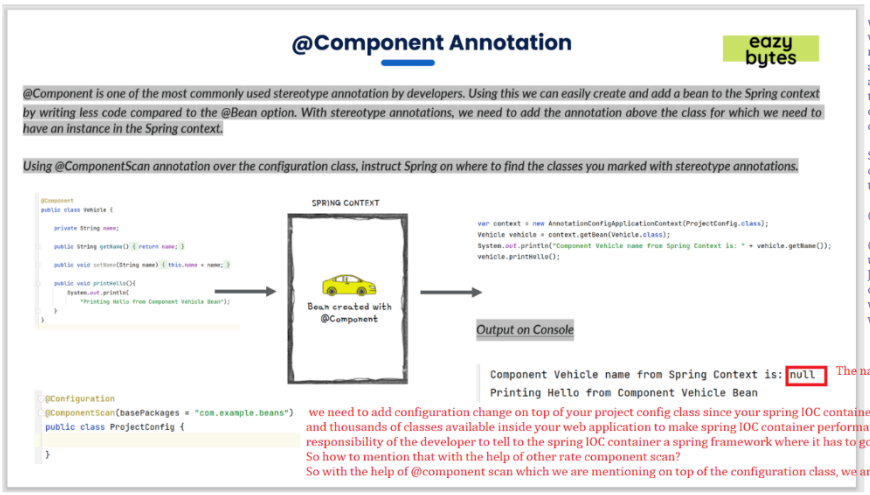


As we have kept @Primary Annotation on Ferrari Vehicle bean it will be considered when we invoke getBean() without any name of the bean .

If we keep Multiplebeans with Primary Annotation of Same class that also will again lead to NoUniqueBeanDefinitionException.More than one Primary Bean found

Primary Annotated Bean will be the default bean that Spring context will consider in case of confusion due to multiple beans of the same type.

19.Creating Beans using @Component Annotation:



with @Bean Annotation
we have to write a lot of code like you have to create a Java method on top of it .we need to annotate with @bean annotation and inside this Java method you need to create an object by yourself and initialize it with whatever values that you want inside it and the return
object from that method will act as a bean inside spring IOC container context.

So when going with @Bean annotation if the number of classes are more it becomes it becomes difficult do avoid this we can use @Component Annotation.

@Component--Stereo type Annotation.

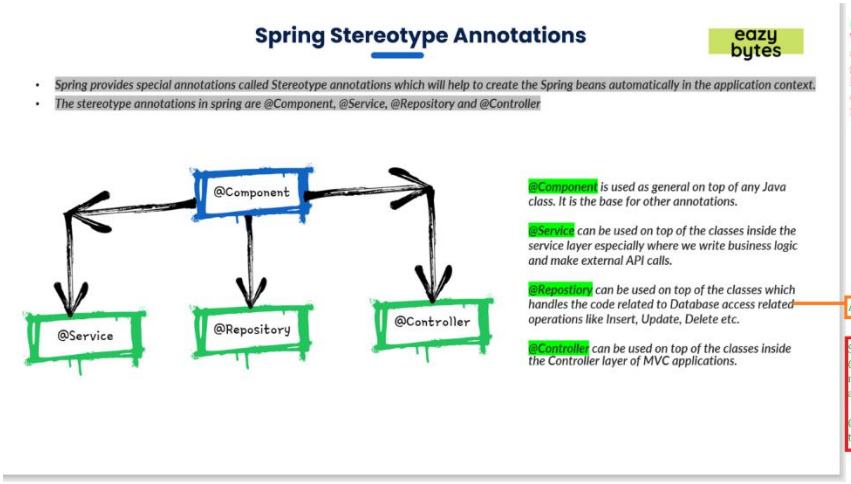
@component is a generic stereotype annotation that we can use on top of any Pojo Java class which will make that pojo Java class as a spring bean because spring IOC container during startup, whenever it sees its annotation, it
will create object by itself rather than looking for a code where we create it with the new operator

The name of the bean will be null

we need to add configuration change on top of your project config class since your spring IOC container can't go and scan all of your packages and thousands and thousands of classes available inside your web application to make spring IOC container performance better.So Spring team decided that it's a responsibility of the developer to tell to the spring IOC container a spring framework where it has to go and scan.
So how to mention that with the help of other rate component scan?
So with the help of @component scan which we are mentioning on top of the configuration class, we are instructing spring, go ahead and look for the Java

classes which are annotated with the stereotype annotations like @component. and based upon that, please go ahead and create the spring beans.

20.Stereo Type Annotations in Spring:



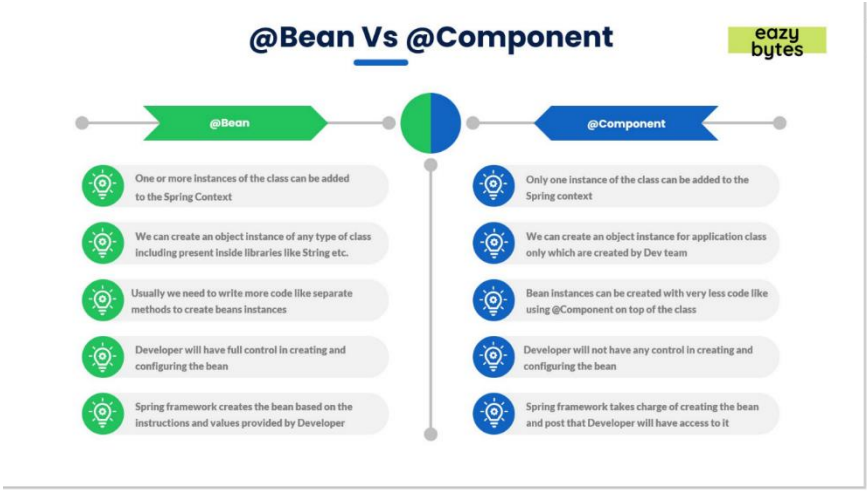
@Component
We will use @Component stereotype annotation in the scenarios where your particular Java class is trying to act as a general Java class, which means it doesn't have any significance. It is just a utility class or any other generic class which your enter application will use.
So in those scenarios, we can keep @Component

Any DAO Classes

So there is no mandate that you need to have @Service, @Repositor, @Controller.It's just that it will improve the reliability of your application and all these special annotations,

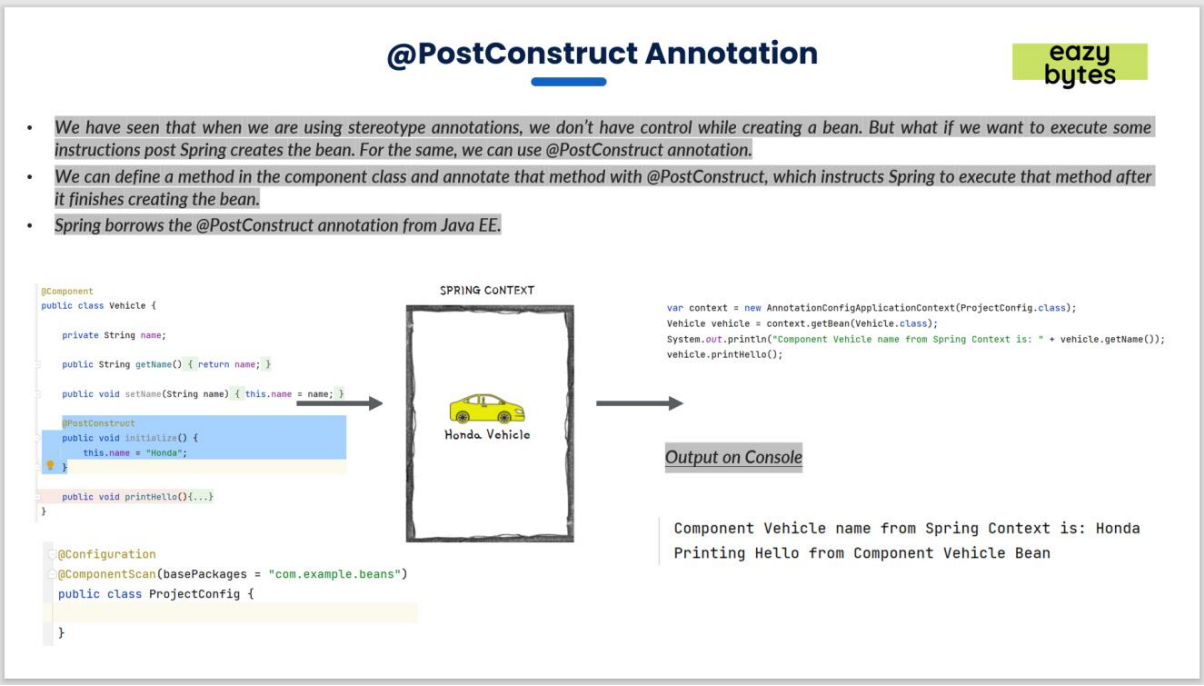
@Service, @Repository and @Controller are extended from the @Component annotation itself.

21. @Bean vs @Component



with the help of @component, we can only create bean object instances for the application classes that are defined by yourself only.

22. @PostConstruct Annotation:



We need to add the below dependency to the application(pom.xml):

<!-- https://mvnrepository.com/artifact/jakarta.annotation/jakarta.annotation-api -->

<dependency>

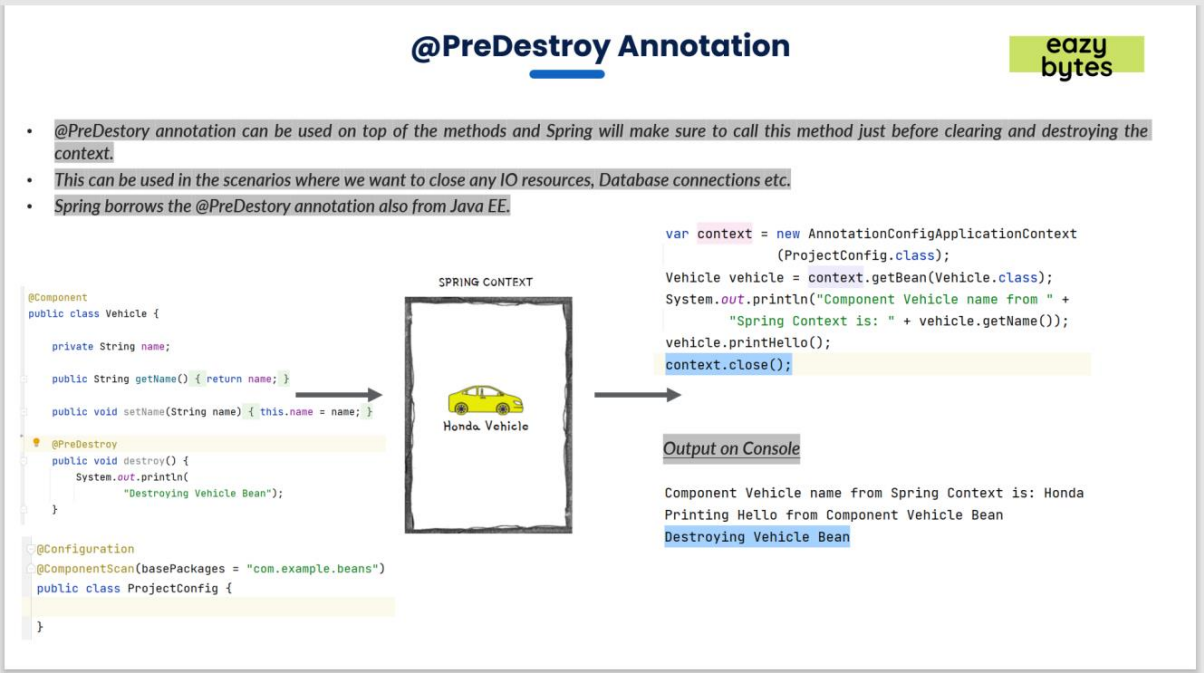
<groupId>jakarta.annotation</groupId>

<artifactId>jakarta.annotation-api</artifactId>

<version>2.1.1</version>

</dependency>

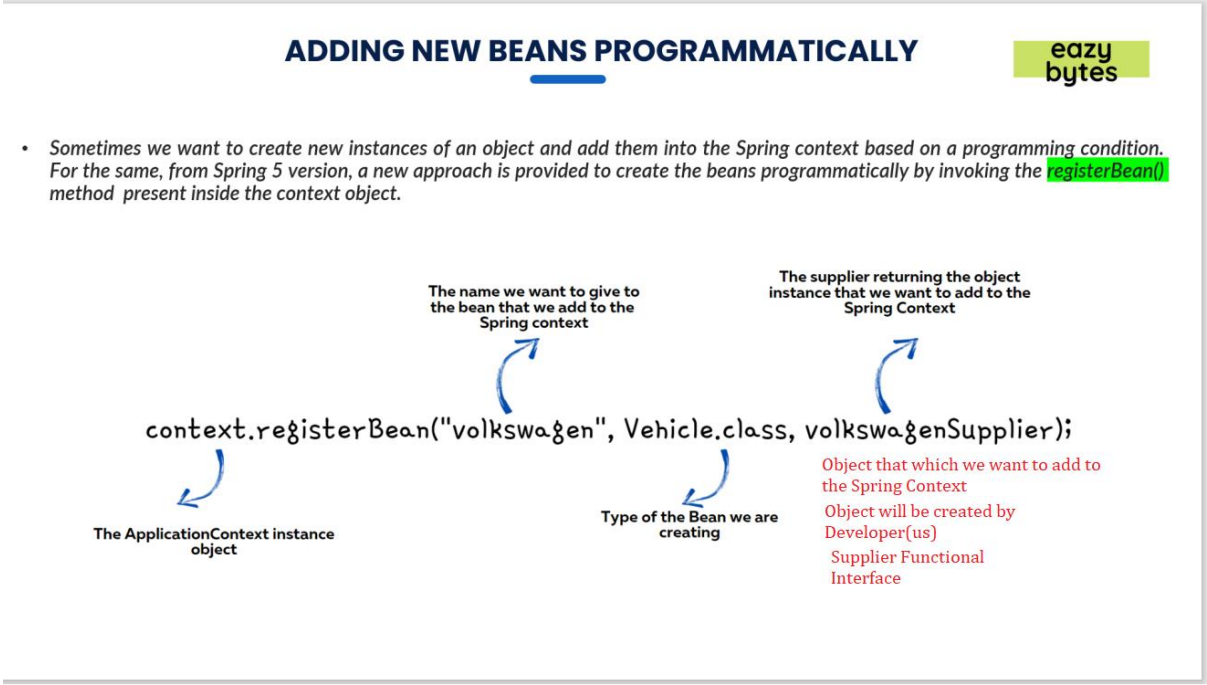
23. @PreDestroy Annotation:



23.Creating Beans Programmatically using registerBean() :

Helps developers in creating the beans based on some programmatic conditions.

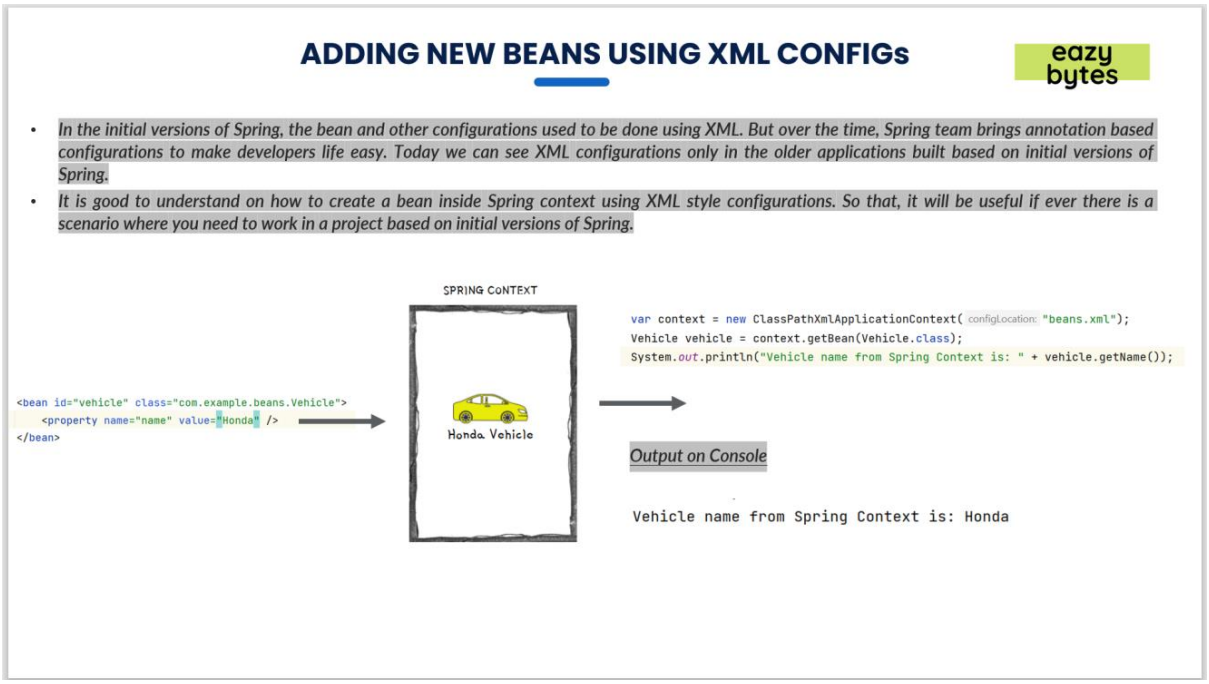
Supplier Interface that doesn't take any input but it returns an Object.

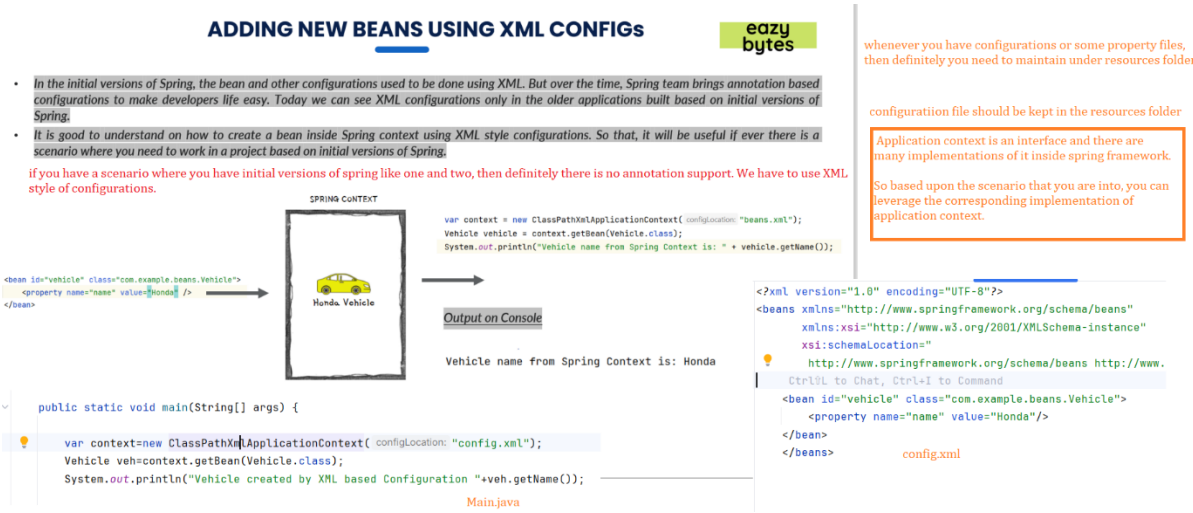



```
package com.example.main;

import com.example.beans.Vehicle;
import com.example.config.ProjectConfig;
import org.springframework.beans.factory.NoSuchBeanDefinitionException;
import org.springframework.context.annotation.AnnotationConfigApplicationContext;

import java.util.Random;
import java.util.function.Supplier;
```

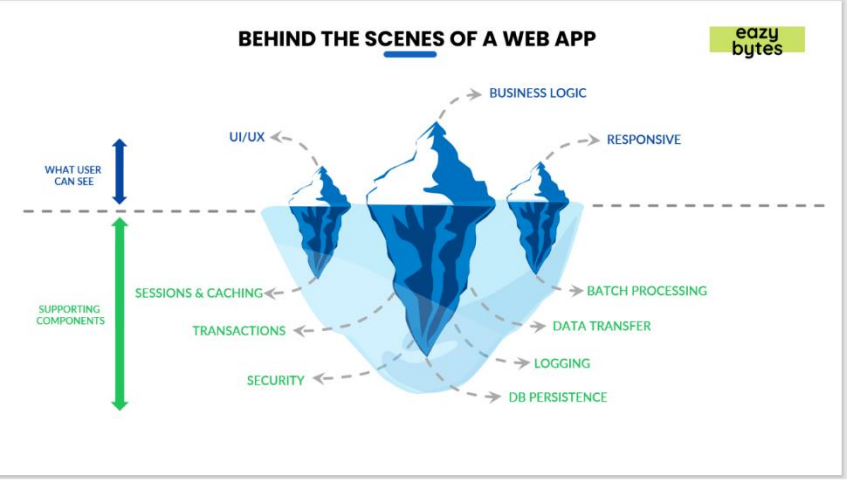
```
public class  Example8 {

    public static void main(String[] args) {

        var context=new AnnotationConfigApplicationContext(ProjectConfig.class);
        Vehicle shine=new Vehicle();
        shine.setName("Shine");
        Supplier<Vehicle> shineSupplier=() ->shine;

        Supplier<Vehicle> javaSupplier=() ->{
            Vehicle javaSup=new Vehicle();
            javaSup.setName("Java_Bike");
            return javaSup;
        };
        Vehicle shineBike= null;
        Vehicle javaBike = null;
        Random rnd=new Random();
        int randomNumber=rnd.nextInt(10);
        if(randomNumber%2==0)
        {
            context.registerBean("Java Bike", Vehicle.class,javaSupplier);
        }
        else
        {
            context.registerBean("Shine Bike", Vehicle.class,shineSupplier);
        }
        Vehicle veh=context.getBean(Vehicle.class);
        System.out.println("Random number "+randomNumber);
        System.out.println("Vehicle from the Spring Container "+veh.getName());

        try {
            shineBike = context.getBean("Java Bike",Vehicle.class);
        }catch (NoSuchBeanDefinitionException noSuchBeanDefinitionException){
            System.out.println("Error while creating Volkswagen vehicle");
        }
        try {
            javaBike = context.getBean("Shine Bike",Vehicle.class);
        }catch (NoSuchBeanDefinitionException noSuchBeanDefinitionException){
            System.out.println("Error while creating Audi vehicle");
        }

        if(null != shineBike){
            System.out.println("Programming Vehicle name from Spring Context is: " +
shineBike.getName());
        }else{
            System.out.println("Programming Vehicle name from Spring Context is: " +
javaBike.getName());
        } context.close();
    }
}
```

25.Creating Beans using XML Configurations:





26.Why should we use frameworks:



As a developer, if I start focusing on building all the supporting components, it will take years and years to build a small web application.
So to avoid such scenarios, we have frameworks where they will provide all the supporting components

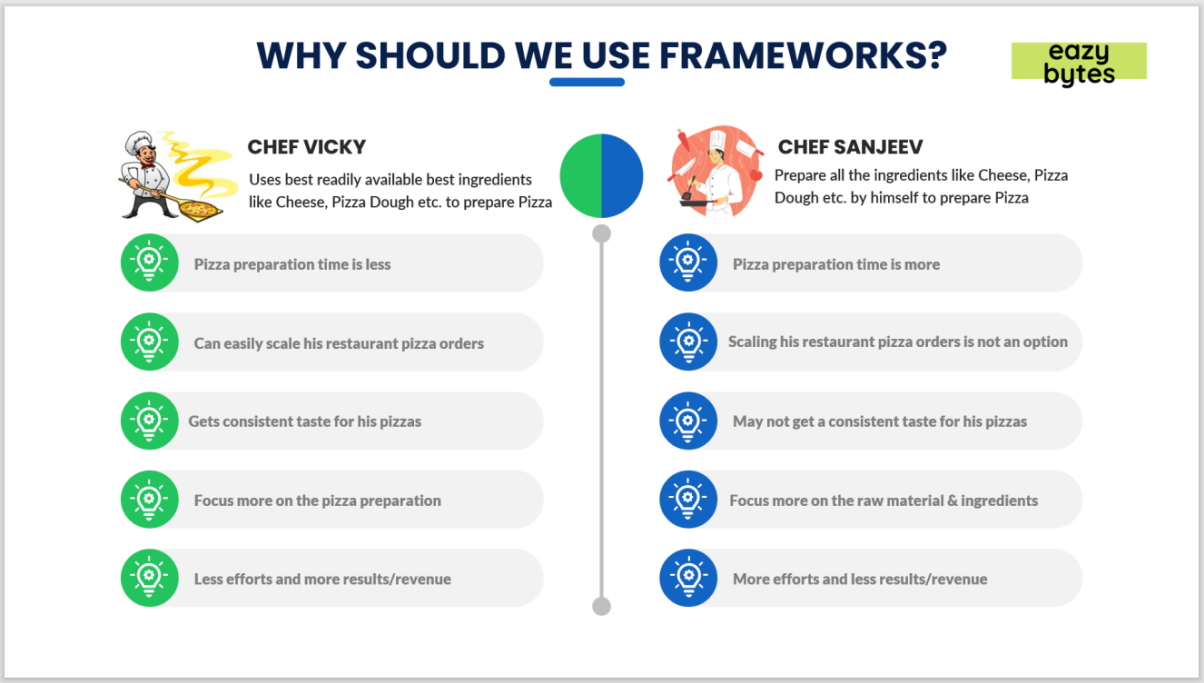
like you want to implement caching or security or DB persistence logging. The scenario can be anything.

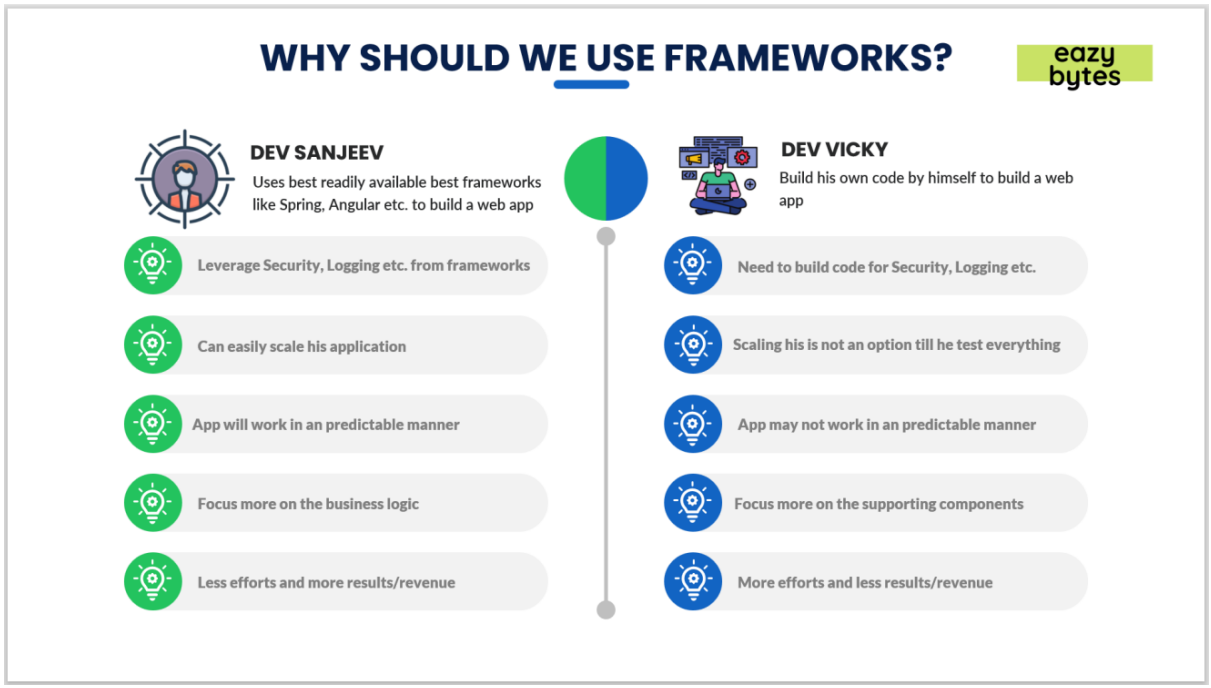
You can leverage these frameworks and they will provide all the common supporting components to the

developers.

And it's a duty of the developer to understand that framework and to configure that framework inside

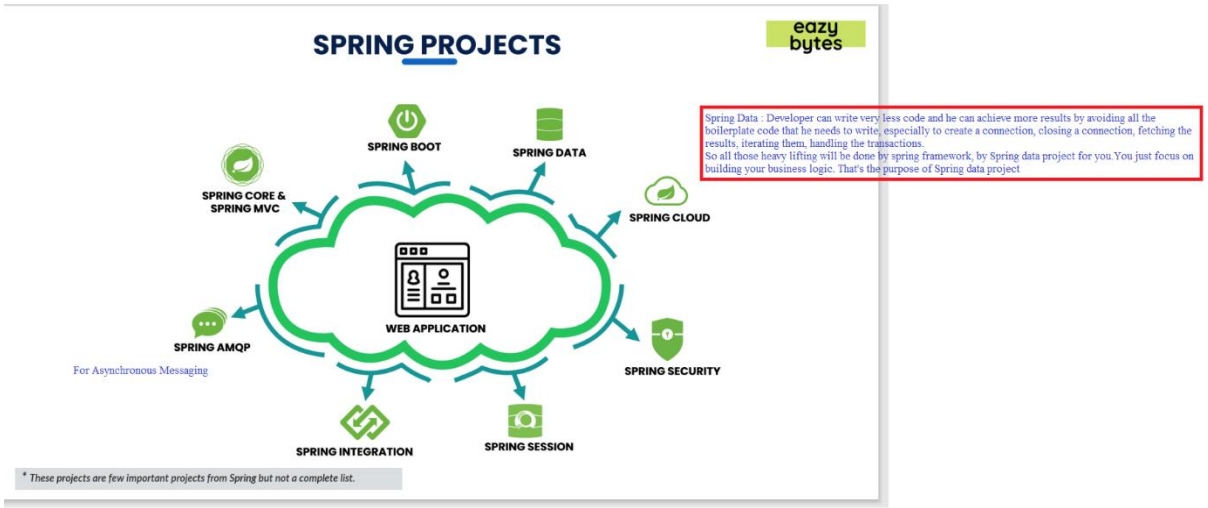
your web application.

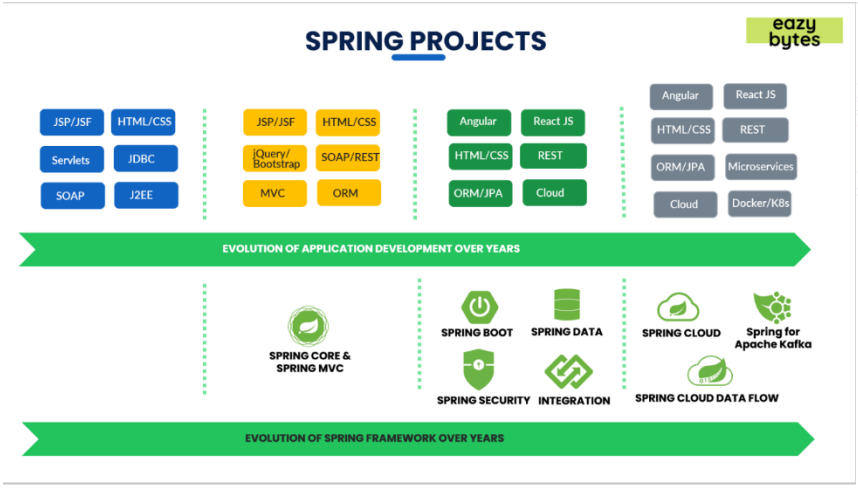




27.Introduction to Spring Projects:

There are many Spring based Projects





the official website of spring is spring.io, and spring is being maintained by the VMware.vVMware is an organization who maintains and regularly introduces various projects or new features orvupgrades to the spring framework.

Spring has many projects and based upon the need that you have and based upon the scenario that you are in, you can choose a specific project and try to adopt that inside your web application.

When we create multiple objects of same class type and try to fetch the bean from context by class type, then Spring cannot guess which instance you've declared you refer to. This will lead to NoUniqueBeanDefinition Exception.

Question 9:

Why should we use frameworks like Spring for web development inside our projects?

- ○ **Because everyone is using and I need to follow the same**
- ○ **Frameworks makes my life super easy and I don't need to write any code. Simply deploy the code & monitor it**
- ◉ **With the help of Frameworks, we can just focus on business logic, since frameworks take care of the common non functional requirements like logging, security etc. Also when frameworks used, it makes our application predictable and easily scalable.**