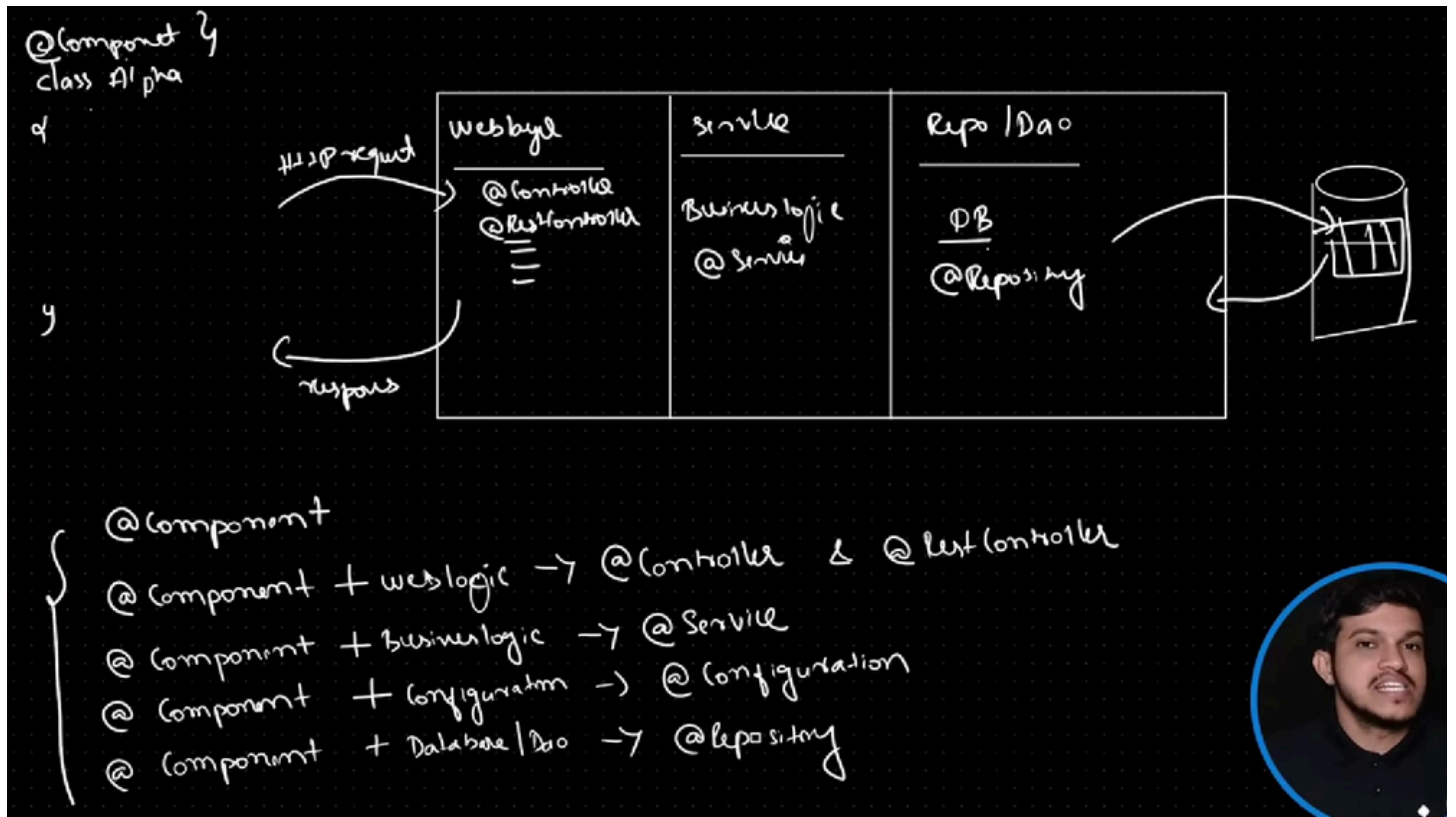


SPRING CORE:

- Annotation Approach:



- We don't have to manage a xml file now.

```

public class Password
{
    String algo;

    public Password(String algo)
    {
        this.algo=algo;
        System.out.println("Password Bean created");
    }

    public String aboutAlgo()
    {
        return "Algo used is "+ algo;
    }
}

```

@Configuration

```

public class Config {

    public Config()
    {
        System.out.println("Config Bean Created");
    }

    public Password createPass()
    {
        return new Password("SHA");
    }
}

```

- Here, we are manually creating Password class's object, and we want Spring to accept it.
- Because Spring won't create its object due to having no annotation of Component in Password class.

```

@Configuration
public class Config {

    public Config()
    {
        System.out.println("Config Bean Created");
    }

    @Bean
    public Password createPass()
    {
        return new Password("SHA");
    }
}

```

- When we manually create object of a class then we have to put the `@Bean` annotation where it is creating an object.

```

@Component("al")
public class Alpha
{
    public Alpha()
    {
        System.out.println("Alpha Bean created");
    }
}

```

- Here “al” is the id of bean created of this Alpha class or default it would be camelcase of class name.
- We used id when we were specifying our Spring framework through xml approach (<bean id="" class="" />).

No XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xmlns:context="http://www.springframework.org/schema/context"
       xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.spr

<context:annotation-config/>
<context:component-scan base-package="com.tshaped"></context:component-scan>

</beans>
```

- Previously we used almost no xml but still used a little code (in image above) for specifying which package to take care of.
- But now, we can delete our xml file.
- After deletion, we put `@ComponentScan(basePackages = {"com.tshaped"})` in our configuration class.
- If we are using `ApplicationContext` class for IOC container then we would have to change its signature too.