

1. If the classes annotated with `@Configuration` , `@Component` (`@Repository` , `@Service`) , `@Controller`  
Are in different packages then how will the `SpringWebApplicationContext` know about them !!!

Use `@ComponentScan` annotation .

2. RULE -- property name MUST start with small case  
Setter name      `public void setPropertyName(PropertyType s)`
3. If there is bean declared in `beans.xml` --- how to use it . `@ImportResource`

#### IN A SINGLE APPLICATION -----

##### 100 beans

- 30 beans are defined in `beans.xml`
- 10 beans in `config.xml`
- 20 beans in `MyJavaConfig` class
- 20 beans in `MyJavaConfig2` class
- 10 beans are annotated as `@Component`
- 5 beans are annotated as `@Service`
- 5 beans are annotated as `@Repository`

---

#### Question

If we want to use the beans created by the container  
In our controller ? Use `@Autowired` = the bean created by container is made available to us

**Rule** --- If a class is annotated as `@Component`,`@Service`,`@Repository`,`@Configuration`,`@Controller`  
or it is a bean in Java Config  
or it is a bean in `beans.xml`

We want that the class MUST be managed by the container ----- i.e. Instantiated and DI

Then

**WE must never create the object of that class using "new"**

We can use interface `ApplicationContextAware` ---- get the context and `getBean` , OR we use `Autowired`

---

Session Management ----- Spring MVC!!!

Ask the container to pass the `HttpServletRequest` to the MAPPING method !!!

---

Spring MVC -----

View ? 1. **JSP**

Another option to use for the view part = 2. **ThymeLeaf**

**Thymeleaf** = it is a spring component that can dynamically generate xml and html !!!

1. Add the thymeleaf-starter dependency in POM
2. Add a first.html in src/main/resources/templates / first.html
3. We write a controller , add a mapping method that returns "first.html"

**ArrayList<String> al =...**

<b>for( String s: al)</b>	<b>th:each= "s:\${al}"</b>
<b>{</b> <b>    Sysout(s)</b> <b>}</b>	<b>&lt;td&gt; &lt; span th:text="\${s}"&gt;&lt;/span&gt;</b>