



IMP Methods of ApplicationContext

Lab49: Files required

1. Lab49.java	2. Hai.java
3. Hello.java	4. JLCAppConfig.java

1. Lab49.java

```
package com.coursecube.spring;

import org.springframework.context.ApplicationContext;
import org.springframework.context.annotation.AnnotationConfigApplicationContext;
/*
 * @Author : Srinivas Dande
 * @Company : CourseCube
 * @Website : www.coursecube.com
 */
public class Lab49 {
    public static void main(String[] args) {

        ApplicationContext ctx = new AnnotationConfigApplicationContext(JLCAppConfig.class);
        System.out.println("-----Now Spring Container is Ready-----");

        System.out.println("1. Get the bean with Bean Name--");
        Hello hello1 = (Hello) ctx.getBean("myhello");
        hello1.show();

        System.out.println("2. Get the bean with Bean Type--");
        Hello hello2 = ctx.getBean(Hello.class);
        hello2.show();

        System.out.println("3. Get the bean with Bean Name and Bean Type--");
        Hello hello3 = ctx.getBean("myhello", Hello.class);
        hello3.show();
        System.out.println("-----");
        System.out.println("4. " + ctx.getBeanDefinitionCount());

        System.out.println("5. List of Beans-----");
        String bnames[] = ctx.getBeanDefinitionNames();
        for (String bname : bnames) {
            System.out.println(bname);
        }
        System.out.println("-----");
        System.out.println("6. " + ctx);
        System.out.println("7. " + ctx.getDisplayName());
        System.out.println("8. " + ctx.getId());
        System.out.println("9. " + ctx.getStartupDate());
        System.out.println("10. " + ctx.getType("myhello"));
```



```
System.out.println("11A. " + ctx.containsBean("myhello"));
System.out.println("11B. " + ctx.containsBean("myhello1"));

System.out.println("12A. " + ctx.isSingleton("myhai"));
System.out.println("12B. " + ctx.isSingleton("myhello"));

System.out.println("13A. " + ctx.isPrototype("myhai"));
System.out.println("13B. " + ctx.isPrototype("myhello"));
}
}
```

2. Hai.java

```
package com.coursecube.spring;

/*
 * @Author : Srinivas Dande
 * @Company : CourseCube
 * @Website : www.coursecube.com
 */
public class Hai {
    String str; //Dependency
    public void setStr(String str) {
        this.str = str;
    }
    public String toString() {
        return str;
    }
}
```

3. Hello.java

```
package com.coursecube.spring;

import org.springframework.beans.factory.annotation.Autowired;

/*
 * @Author : Srinivas Dande
 * @Company : CourseCube
 * @Website : www.coursecube.com
 */
public class Hello {

    @Autowired
    Hai hai;
    public void show() {
        System.out.println("Hello-show()");
        System.out.println(hai);
    }
}
```



4. JLCAppConfig.java

```
package com.coursecube.spring;

import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
/*
 * @Author : Srinivas Dande
 * @Company : CourseCube
 * @Website : www.coursecube.com
 */

@Configuration
public class JLCAppConfig {

    @Bean(name = "myhai")
    public Hai createHai() {
        Hai hai = new Hai();
        hai.setStr("I am Bean - Hai");
        return hai;
    }

    @Bean(name = "myhello")
    public Hello createHello() {
        return new Hello();
    }

}
```

Interview Questions – Part 2

Q161) What is the Advantage of Externalizing Bean Properties?

Ans:

Q162) What is the Bean to Configure to access Externalized Bean Properties?

Ans:

Q163) How to access the Externalized Bean Properties from Property Files?

Ans:

Q164) What is the use @Value Annotation?

Ans:

Q165) How the Properties from Environment Object?

Ans:



Q166) How to Implement Internationalization(I18N)?

Ans:

Q167) What are the Naming Conventions to follow when you write Multiple Message Bundles?

Ans:

Q168) What is the Bean to Configure to access Messages from Message Bundles?

Ans:

Q169) How to access Messages from Message Bundles?

Ans:

Q170) What is the use of MessageSource?

Ans:

Q171) What is the use of PropertyEditors?

Ans:

Q172) What is the Super for Custom PropertyEditor class?

Ans:

Q173) What is the method to Override in Custom PropertyEditor class?

Ans:

Q174) What are the Steps to write Custom PropertyEditors?

Ans:

Q175) How to Register Custom PropertyEditor with Container?

Ans:

Q176) What are the standard events provided by the Spring Container?

Ans:

Q177) When exactly ContextStartedEvent will be published by the Spring Container?

Ans:

Q178) When exactly ContextRefreshedEvent will be published by the Spring Container?

Ans:

Q179) When exactly ContextStoppedEvent will be published by the Spring Container?

Ans:

Q180) When exactly ContextClosedEvent will be published by the Spring Container?

Ans:



Q181) Can I refresh() method explicitly on ConfigurableApplicationContext?

Ans:

Q182) What are the Ways Available to Write the Listeners in Spring?

Ans:

Q183) What are the Steps to Write the Listeners with ApplicationListener interface?

Ans:

Q184) What are the Steps to Write the Listeners with @EventListener annotation?

Ans:

Q185) How to Register the Listeners with Spring Container?

Ans:

Q186) Can I write the One Listener to Handle Multiple Events?

Ans:

Q187) Can I write the Multile Listeners to Handle One Event?

Ans:

Q188) Can I write the Different Listeners to Handle Dedicated Events?

Ans:

Q189) What are the Ways Available to Write the Application Events or Custom Events?

Ans:

Q190) What are the Steps to Write the Custom Events with ApplicationEvent class?

Ans:

Q191) Can I write the Custom Events without ApplicationEvent class?

Ans:

Q192) What are the Steps to Write the Custom Events without ApplicationEvent class?

Ans:

Q193) How to publish Application Events?

Ans:

Q194) What is the Use of ApplicationEventPublisher?

Ans:

Q195) How Can I access the Required Bean Object from Spring container?

Ans:

Q196) How Can I get the List of Bean Names running in the Spring container?

Ans:



Q197) How Can I Total Number of Beans running in the Spring container?

Ans:

Q198) How Can I check whether Bean with the given Bean Name is running in the Spring container?

Ans:

Q199) How Can I check type of Bean with the given Bean Name?

Ans:

Q200) How Can I check whether Bean is Singleton?

Ans:

Q201) How Can I check whether Bean is Prototype?

Ans:

Q202) What are the stereotype annotations?

Ans:

- @Component
- @Controller
- @Service
- @Repository

Q203) What is the use of stereotype annotations?

Ans:

Q204) When Can I use @Controller annotation?

Ans:

Q205) When Can I use @Service annotation?

Ans:

Q206) When Can I use @Repository annotation?

Ans:

Q207) When Can I use @Component annotation?

Ans:

Q208) When Can I use @ComponentScan annotation?

Ans:

Q209) What is the Need for Injecting ApplicationContext into the Bean?

Ans:

Q210) What are the Annotations Used in IOC so far?

Ans: