What is ApplicationContext Interface and what are its Implementation Classes

Description

Introduction

The ApplicationContextÂ is the central interface within a Spring application for providing configuration information to the application.Â There are two interfaces which represent the Spring IOC container. First is the BeanFactory. BeanFactory is the root interface which provides basic functionality for managing spring beans. And another interface is ApplicationContext which is the sub-interface of BeanFactory. ApplicationContext provides all the features of BeanFactory plus some additional features.

Most of the times ApplicationContext is the default spring container.

Below are some of the important features of ApplicationContext.

- supports internationalization
- publishing events
- resolving messages
- application layer specific contexts.

ApplicationContext Interface Implementation Classes

1. FileSystemXMLApplicationContext

FileSystemXMLApplicationContext is used to load an XML-based spring configuration file from the file system or from URLs. Below example shows the process of creating the spring container and loading the beans for XML-based configuration.

```
String path = "C:/Spring-pet-project/src/main/resources/spring-servlet.xml";

ApplicationContext appContext = new FileSystemXmlApplicationContext(path);
EmployeeService employeeService = appContext.getBean("employeeService", Employ
```

2. ClassPathXmlApplicationContext

If we want to load XML configuration from the classpath then we can use ClassPathXmlApplicationContext.

```
ApplicationContext appContext = new ClassPathXmlApplicationContext("spring-ser
EmployeeService employeeService = appContext.getBean("employeeService", Employ
```

3. XmlWebApplicationContext

If we want to use XML based configuration in our web application then we can use XmlWebApplicationContext class. The XML file is present in /WEB-INF/

```
public class MyXmlWebApplicationInitializer implements WebApplicationInitializ

  public void onStartup(ServletContext container) throws ServletException {
    XmlWebApplicationContext context = new XmlWebApplicationContext();
    context.setConfigLocation("/WEB-INF/spring/applicationContext.xml");
    context.setServletContext(container);

    // Servlet configuration
  }
}
```

4. AnnotationConfigApplicationContext

AnnotationConfigApplicationContext was introduced in Spring 3.0. It takes classes annotated with @Component, @Configuration and JSR-330 metadata as input. Below example shows the usage of AnnotationConfigApplicationContext container with Java based configuration.

```
ApplicationContext context = new AnnotationConfigApplicationContext(EmployeeCo
EmployeeService employeeService = context.getBean(EmployeeService.class);
```

5. AnnotationConfigWebApplicationContext

AnnotationConfigWebApplicationContext is a web variant of AnnotationConfigApplicationContext. We can use this class when we configure Spring’s ContextLoaderListener servlet listener or a Spring MVC DispatcherServlet in a web.xml file.

```
public class MyWebApplicationInitializer implements WebApplicationInitializer

  public void onStartup(ServletContext container) throws ServletException {
    AnnotationConfigWebApplicationContext context = new AnnotationConfigWebApp
    context.register(EmployeeConfig.class);
    context.setServletContext(container);

    // servlet configuration
  }
```

```
}
```

Category

1. Core Spring

Date Created
December 27, 2022
Author
kk-ravi144gmail-com