IoC Container

Summary

It is the service that provides the basic functions of a framework - Inversion of Control (IoC) Container function. It improves flexibility and expansion by setting the dependence on other object that it refers to outside the source code, rather than hard coding inside the source code when it creates the object.

Main Concept

Inversion of Control (IoC)

IoC is the abbreviation of Inversion of Control. It means "inverse control". Let's examine what is the phenomenon of reversing the control. The application developed generally has the control for the work of creating the Java object and connecting mutual dependence when developing application based on Java in the past. When using Servlet and EJB, however, the control is transferred to Servlet Container and EJB Container so that Container assumes full charge of Life Cycle of object. Like this, reversal of control in IoC means that the control for all objects has changed from creation of object to management of life cycle.

Related Document

- Inversion of Control written by Martin Fowler
- Korean translation of Inversion of Control

Dependency Injection

Refers to automatic connection of dependence relations between classes by container based on Bean Definition information.Because the container automatically connects the dependence, the developer does not need to get involved in dependence using the container API, reducing subordination to the container API. Developers only need to add information requiring dependence at the empty setting file (store management file).

Related Document

- Inversion of Control Containers and the Dependency Injection pattern written by Martin Fowler
- Korean translation of Inversion of Control Containers and the Dependency Injection pattern by Choi Byeom Kyun (Blog Java Can Do It)

Open Source Used

- IoC Container uses Spring Framework without modification.

Description

This IoC Container uses the function of Spring Framework without modification. This guide document translates and summarizes The Spring Framework - Reference Documentation. If detailed description is required for Spring Framework IoC Container, refer to The Spring Framework - Reference Documentation original document and Spring Framework API.

IoC Container of Spring Framework

The org.springframework.beans and org.springframework.context packages provide the foundation for IoC Container of Spring Framework. The BeanFactoryinterface provides more advanced setting mechanism for managing objects.

The ApplicationContextinterface (sub-interface of BeanFactoryinterface), created based on the BeanFactoryinterface, provides the context function specialized to application layer such as Spring AOP, message resource processing (used at internationalization), event propagation, WebSpplicationContextfor web application, in addition to the function provided by BeanFactory.

In summary, BeanFactory provides the setting function for framework and basic function, while ApplicationContext additionally provides functions more suitable to Enterprise environment. Since ApplicationContext is the perfect superset ofBeanFacatory, description on the behavior and function of BeanFactory also applies to ApplicationContext.

This document is largely divided into 2 parts. The first part describes the basic principle applied to both BeanFactory andApplicationContextand the second part describes the features applied to ApplicationContext only.

- Basics
  Explains the basic concept required to explain IoC Container and how to use it.

- Dependencies
  Explains how to use and set Dependency Injection that is the core function of IoC Container,.

- Bean scope
  Explains how to create and apply Bean managed by IoC Container.

- Customizing the nature of a bean
  Explain how to change the property of Bean including management of lifecycle of Bean, reference of Container where Bean belongs to.

- Bean definition inheritance
  Explains definition inheritance of Bean.

- Container extension points
  Explain how to expand the function of IoC Container.

- The ApplicationContext
  Explains the function provided by ApplicationContext only.

- Annotation-based configuration
  Explain how to define Bean based on Java Annotation.

- Classpath scanning for managed components
  Explain how to set the Java Annotation foundation for base Bean to be inserted by Dependency Injection.

Reference

- *The Spring Framework* - Reference Documentation / Chapter 3. The IoC container
- *Inversion of Control*
- *Inversion of Control Containers and the Dependency Injection pattern*
- Korean edition of *Inversion of Control Containers and the Dependency Injection pattern*(translated by Choi Byeom Kyun (Blog Java Can Do It))