

Part 1 – Spring Framework Reference Documentation

Chapter 1. The IoC Container [<https://docs.spring.io/spring/docs/5.2.x/spring-framework-reference/core.html> (Links to an external site.) (Links to an external site.)]

This chapter covers the Spring Framework implementation of the Inversion of Control (IoC) principle. IoC is also known as dependency injection (DI). It is a process whereby objects define their dependencies (that is, the other objects they work with) only through constructor arguments, arguments to a factory method, or properties that are set on the object instance after it is constructed or returned from a factory method. The container then injects those dependencies when it creates the bean. This process is fundamentally the inverse (hence the name, Inversion of Control) of the bean itself controlling the instantiation or location of its dependencies by using direct construction of classes or a mechanism such as the Service Locator pattern.

Part 2 – Reading Assignment

https://www.safaribooksonline.com/library/view/spring-in-action/9781617291203/kindle_split_009.html (Links to an external site.) (Links to an external site.)

https://www.safaribooksonline.com/library/view/spring-in-action/9781617291203/kindle_split_010.html (Links to an external site.) (Links to an external site.)

Part 3 – Programming Assignment

Redo HW3-Part4 using Spring MVC. Use IoC to create instances.

Use BeanNameUrlHandlerMapping for the controllers.

If you used Spring MVC in HW3, then convert the controllers to Annotated controllers.

If you used Spring Annotated controllers in HW3, then convert use BeanNameUrlHandler Mapping..

<https://docs.spring.io/spring-framework/docs/current/javadoc-api/org/springframework/web/servlet/handler/BeanNameUrlHandlerMapping.html> (Links to an external site.) (Links to an external site.)

Part 4 – Programming Assignment

Redo HW3-Part5 using Spring MVC.

Use IoC to create instances.

Use SimpleUrlHandlerMapping for the controllers.

<https://docs.spring.io/spring-framework/docs/current/javadoc-api/org.springframework.web.servlet.handler.SimpleUrlHandlerMapping.html> (Links to an external site.) (Links to an external site.)

Part 5 – Programming Assignment

Create a utility class that has a get method returning a string (it could be any random message). Use Spring's IoC container to inject this using Autowired to inject this. Use annotated controller. Do not use new operator.

Run the controller from different browsers, and print the address of the Utility class instance to see if it is the same object shared by all user requests.

Part 6 – Programming Assignment

Redo Part 5, but inject the instance to the handler method.

Try this from different browsers to see if a new object is created for each request.