Daniel Cender

CST-341-O500

March 18, 2020

Activity 3: SpringBeans and DI

**Screenshots assignment3a:**

1. Screenshot of the OrdersBusinessService executing from Controller.

A screenshot of a social media post

Description automatically generated

1. Screenshot of the User Controller code with injection code.

A screenshot of a social media post

Description automatically generated

1. Screenshot of the SpringBean Configuration file.

A screenshot of a social media post

Description automatically generated

1. Screenshot of the AnotherOrdersBusinessService executing from Controller.

A screenshot of a social media post

Description automatically generated

**Screenshots assignment3b:**

1. Screenshot of lifecycle with README.txt.

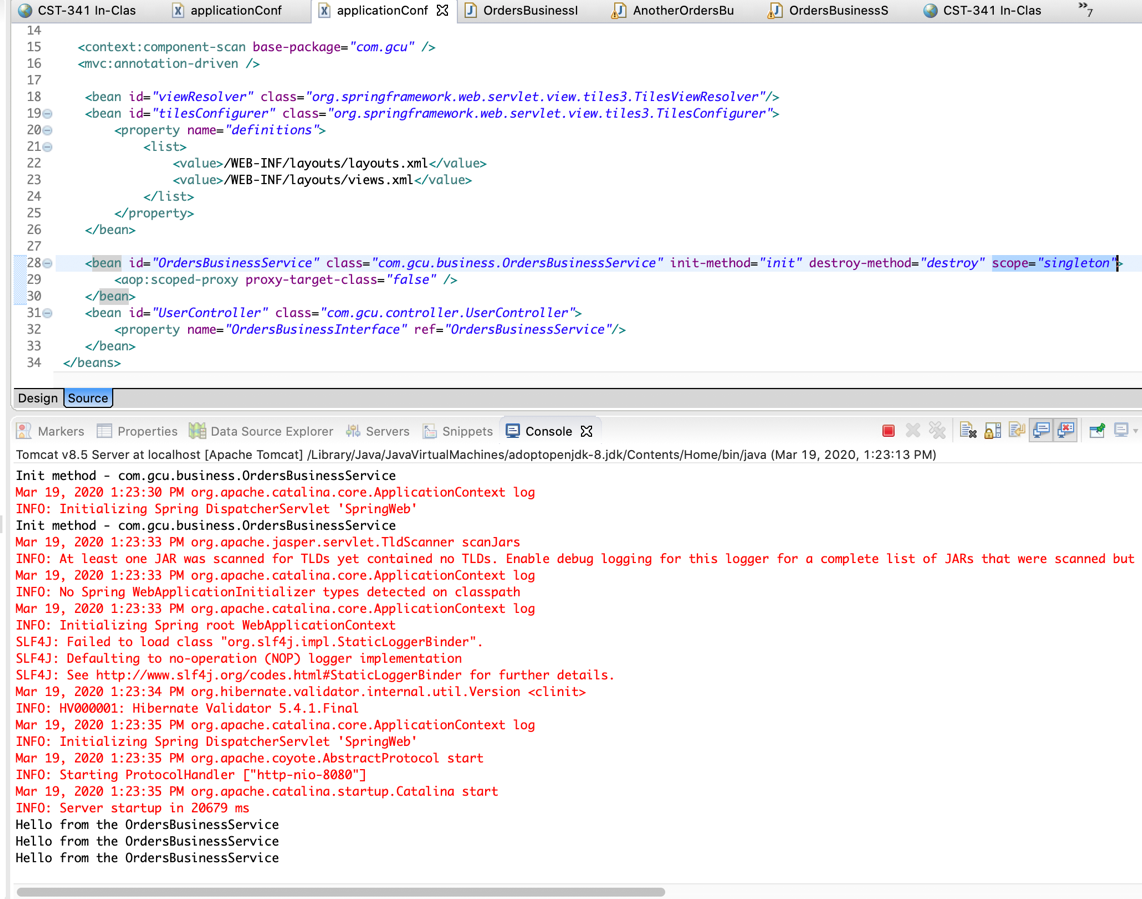
A screenshot of a social media post

Description automatically generated

The SpringBean’s init method was called twice. The first time instantiated the bean into the WebApplicationContext, and the second call initialized it into the state of the FrameworkServlet (UserController). Every submission of the test form after the first request will use the already-instantiated Bean.

1. Screenshots of scopes with README.txt.

**With scope as “singleton”:**



This scope declaration limits the instantiation of a Bean to one per IoC container. This is the default scope setting, so it matches the previous test.

**With scope as “prototype”:**

A screenshot of a social media post

Description automatically generated

The “prototype” scope allows the Bean to have any number of instances. Here, the Bean is injected into the IoC container at runtime, so each form submission includes a completely different instance of the SpringBean.

**With scope as “request”:**

A screenshot of a social media post

Description automatically generated

The “request” scope limits an instance of a SpringBean to a request within a web-aware SpringApplicationContext. As seen here, for each POST request made to the FrameworkServlet, a new instance of the Bean is created.

**With scope as “session”:**

A screenshot of a social media post

Description automatically generated

This scope is similar to “request”, except that the Bean continues to exist between requests, as long as the HTTP session state holds. Again, this can only be used in a web-aware SpringApplicationContext.

**Screenshots assignment3c:**

1. Screenshot of auto-wire.

A screenshot of a social media post

Description automatically generated

1. Screenshots of scopes using Java Annotations.

A screenshot of a social media post

Description automatically generated

**With annotated “singleton” scope:**

**A screenshot of a social media post

Description automatically generated**

**With annotated “prototype” scope:**

**A screenshot of a social media post

Description automatically generated**

**With annotated “request” scope:**

**A screenshot of a social media post

Description automatically generated**

**With annotated “session” scope:**

**A screenshot of a social media post

Description automatically generated**

**Screenshots assignment3d:**

1. Screenshot of Spring based REST Service.

A screenshot of a social media post

Description automatically generated

**Incompleted from here onward – Jersey Servlet Install Moved Entirely to Maven Dependencies**