CS544 EA

Spring

Dependency Injection Features

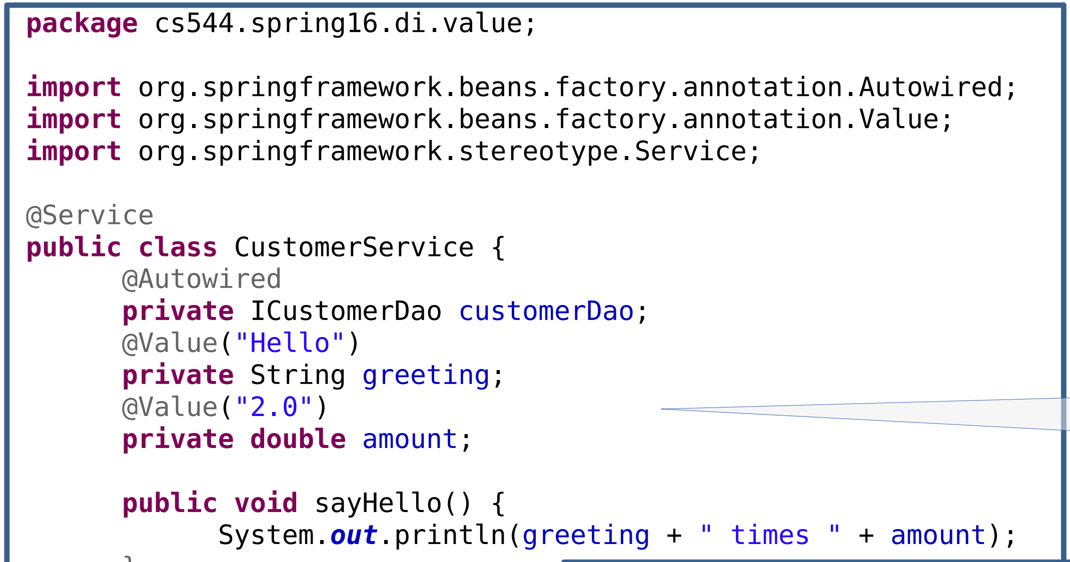
DI Features

- We will look at:
  - Injecting Values
  - Injecting Collections
  - Inheritance and Dependency injection

Injecting Values

Strings and Primitives

- Spring can also **inject a value**



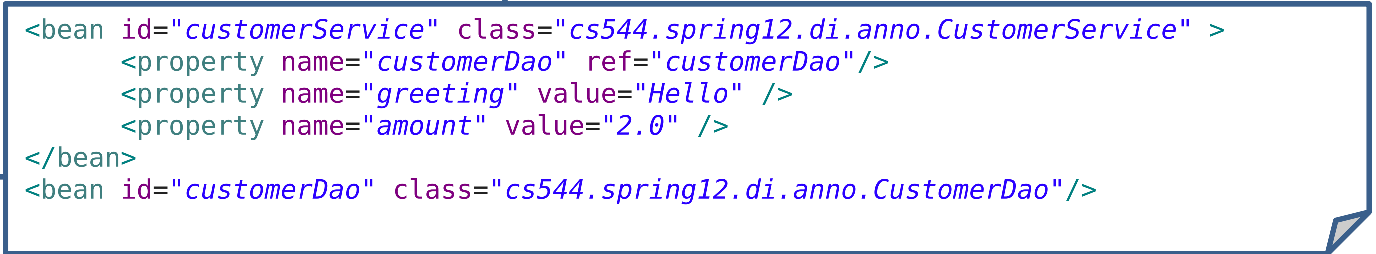
```java
package cs544.spring16.di.value;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.stereotype.Service;

@Service
public class CustomerService {
	@Autowired
	private ICustomerDao customerDao;
	@Value("Hello")
	private String greeting;
	@Value("2.0")
	private double amount;

	public void sayHello() {
		System.out.println(greeting + " times " + amount);
	}
}
```

Although specified as a String Spring will cast to needed type



```xml
<bean id="customerService" class="cs544.spring12.di.anno.CustomerService" >
	<property name="customerDao" ref="customerDao"/>
	<property name="greeting" value="Hello" />
	<property name="amount" value="2.0" />
</bean>
<bean id="customerDao" class="cs544.spring12.di.anno.CustomerDao"/>
```

SPEL

- Spring Expression Language
  - Can be used inside @Value
  - Looks a lot like the JSF / JSP expression language

```
#{ systemProperties['user.language'] }

#{ customerService.greeting }
```