CS544 EA

Spring:

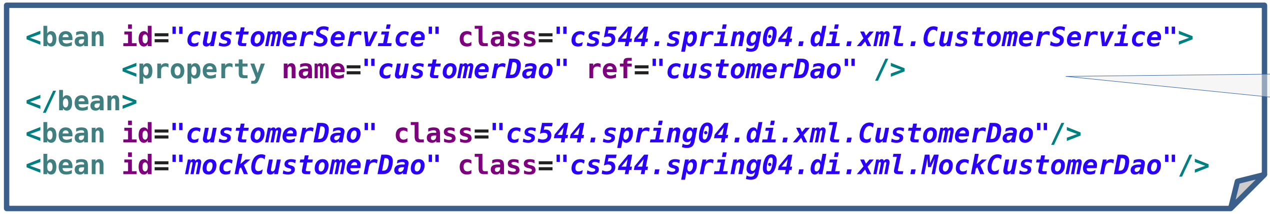
Dependency Injection

Dependency Injection

- **Dependency Injection** is used to gain greater flexibility, loose coupling. Possible to change the connections without chancing the code.
  - Code is clean, associations not hard-coded (no new)
  - Unit testing becomes easier (inject mock)
  - AOP / Interceptors become possible (inject proxy)

How Does DI work?

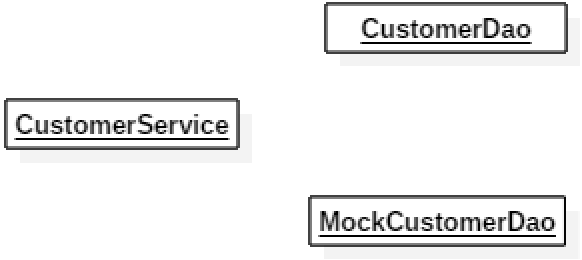
- Spring creates all the specified beans (config)



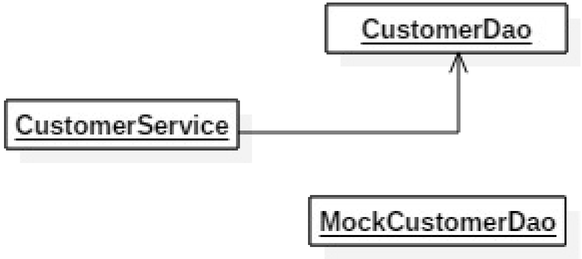
```
<bean id="customerService" class="cs544.spring04.di.xml.CustomerService">
      <property name="customerDao" ref="customerDao" />
</bean>
<bean id="customerDao" class="cs544.spring04.di.xml.CustomerDao"/>
<bean id="mockCustomerDao" class="cs544.spring04.di.xml.MockCustomerDao"/>
```

Specifies DI

- And then sets the customerDao property inside the CustomerService class to the CustomerDao object



Create beans



Inject CustomerDao

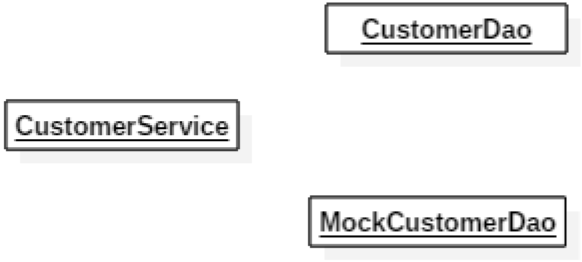
Change the Wiring

- Spring creates all the specified beans (config)

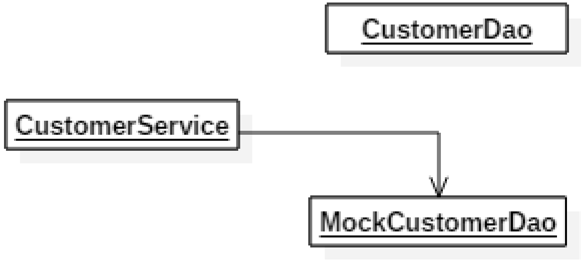
```
<bean id="customerService" class="cs544.spring04.di.xml.CustomerService">
      <property name="customerDao" ref="mockCustomerDao" />
</bean>
<bean id="customerDao" class="cs544.spring04.di.xml.CustomerDao"/>
<bean id="mockCustomerDao" class="cs544.spring04.di.xml.MockCustomerDao"/>
```

Inject Mock instead

- And then sets the customerDao property inside the CustomerService class to the MockCustomerDao object



Create beans



Inject MockCustomerDao

Dependency Injection

- We will look at:
  - XML and Dependency Injection
  - Annotations and Dependency Injection
  - Java Config and Dependency Injection