

Bean Scopes



Bean Scopes

- Scope refers to the lifecycle of a bean
- How long does the bean live?
- How many instances are created?
- How is the bean shared?

Default Scope: Singleton

```
<beans ...>

<bean id="myCoach"
      class="com.luv2code.springdemo.TrackCoach">
    ...
</bean>

</beans>
```

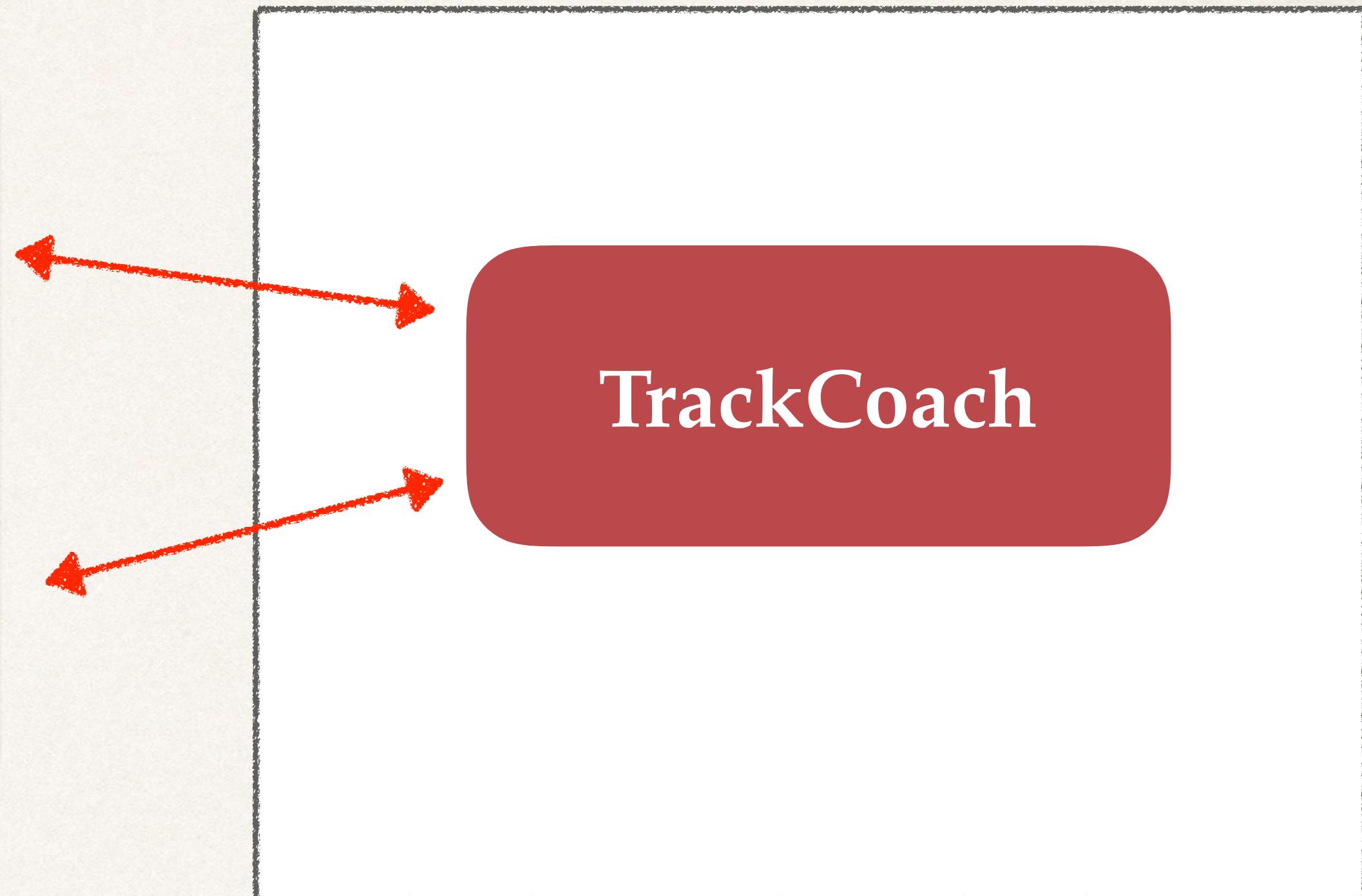
What Is a Singleton?

- Spring Container creates only one instance of the bean, by default
- It is cached in memory
- All requests for the bean
 - will return a SHARED reference to the SAME bean

What is a Singleton?

Spring

```
Coach theCoach = context.getBean("myCoach", Coach.class);  
...  
Coach alphaCoach = context.getBean("myCoach", Coach.class);
```



Explicitly Specify Bean Scope

```
<beans ... >

<bean id="myCoach"
      class="com.luv2code.springdemo.TrackCoach"
      scope="singleton"
```

Additional Spring Bean Scopes

Scope	Description
singleton	Create a single shared instance of the bean. Default scope.
prototype	Creates a new bean instance for each container request.
request	Scoped to an HTTP web request. Only used for web apps.
session	Scoped to an HTTP web session. Only used for web apps.
global-session	Scoped to a global HTTP web session. Only used for web apps.

Prototype Scope Example

Prototype scope: new object for each request

```
<beans ... >

<bean id="myCoach"
      class="com.luv2code.springdemo.TrackCoach"
      scope="prototype">
    ...
</bean>

</beans>
```

Prototype Scope Example

Prototype scope: new object for each request

Spring

```
Coach theCoach = context.getBean("myCoach", Coach.class);
```

...

```
Coach alphaCoach = context.getBean("myCoach", Coach.class);
```

