

Professional Skills
Agile Fundamentals
Jira
Git
Databases Introduction
Java Beginner
Maven
Testing (Foundation)
Java Intermediate
HTML
CSS
Javascript
Spring Boot
<div><div></div>Introduction to Spring Boot</div>
<div><div></div>Multi-Tier Architecture</div>
<div><div></div>Beans</div>
<div><div></div>Bean Scopes</div>
<div><div></div>Bean Validation</div>
<div><div></div>Dependency Injection</div>
<div><div></div>Components</div>
<div><div></div>Configuration</div>
<div><div></div>Connecting to a Database</div>
<div><div></div>Entities</div>
<div><div></div>Postman</div>
<div><div></div>Controllers</div>
<div><div></div>Services</div>
<div><div></div>Repositories</div>
<div><div></div>Custom Queries</div>
<div><div></div>Data Transfer Objects</div>
<div><div></div>Lombok</div>
<div><div></div>Custom Exceptions</div>
<div><div></div>Swagger</div>
<div><div></div>Profiles</div>
<div><div></div>Pre-Populating Databases for Testing</div>
<div><div></div>Unit testing with Mockito</div>

Bean Scopes

Contents

- [Overview](#)
 - [Singleton](#)
 - [Prototype](#)
- [Tutorial](#)
- [Exercises](#)

Overview

The **scope** of a bean determines when a bean is created and how long it lasts for.

Singleton

The default bean scope - **singleton** beans - are created once and then re-used forever.

Typically, these will be *stateless* and therefore will not store any information internally.

Prototype

Unlike singleton beans, every class that depends on a **prototype** bean will depend on a *separate instance* of that bean.

Every time a prototype bean is *injected*, a new instance of that bean is created.

These beans are typically *stateful*, and thus *will* store information internally.

Tutorial

Here, we've set up a configuration class containing our **greeting** bean:

```
@Configuration
public class AppConfig {

    @Bean
    public String greeting() {
        return "Hello my friends";
    }
}
```

By default, this **greeting** bean will default to the **singleton** scope - there will only ever be one of this bean created each time the app is run.

We can change this bean to be **prototype** scoped by using the **@Scope** annotation:

```
@Configuration
public class AppConfig {

    @Bean
    @Scope("prototype")
    public String greeting() {
        return "Hello my friends";
    }
}
```

Now, a new **greeting** will be created every time this bean is accessed.

<div><div></div><div>Testing</div></div>
Selenium
Sonarqube
Advanced Testing (Theory)
Cucumber
MongoDB
Express
NodeJS
React
Express-Testing
Networking
Security
Cloud Fundamentals
AWS Foundations
AWS Intermediate
Linux
DevOps
Jenkins Introduction
Jenkins Pipeline
Markdown
IDE Cheatsheet

Exercises

Using the code you created within the [Beans module](#), do the following:

1. Create a bean called `farewell` which is prototype scoped.
2. Call both beans. What differences can you see between the `greeting` and `farewell` beans?