spring AOP

tutorialspoint

SIMPLY EASY LEARNING

www.tutorialspoint.com

https://www.facebook.com/tutorialspointindia https://twitter.com/tutorialspoint

About the Tutorial

One of the key components of Spring Framework is the Aspect Oriented Programming (AOP) framework. Aspect Oriented Programming entails breaking down program logic into distinct parts called so-called concerns.

This tutorial will take you through simple and practical approaches while learning AOP framework provided by Spring.

Audience

This tutorial has been prepared for beginners to help them understand the basic to advanced concepts related to AOP framework of Spring.

Prerequisites

Before you start practicing various types of examples given in this tutorial, we assume that you are already aware about computer programs and computer programming languages.

Copyright & Disclaimer

Table of Contents

Spring AOP Basics

1. Spring AOP – Overview

One of the key components of Spring Framework is the **Aspect Oriented Programming (AOP)** framework. Aspect Oriented Programming entails breaking down program logic into distinct parts called **so-called concerns**. The functions that span multiple points of an application are called **cross-cutting concerns.** These cross-cutting concerns are conceptually separate from the application's business logic. There are various common good examples of aspects such as logging, auditing, declarative transactions, security, caching, etc.

The key unit of modularity in OOP is the class, whereas in AOP the unit of modularity is the aspect. Dependency Injection helps you decouple your application objects from each other, while AOP helps you decouple cross-cutting concerns from the objects that they affect. AOP is like triggers in programming languages such as Perl, .NET, Java, and others.

Spring AOP module lets interceptors intercept an application. For example, when a method is executed, you can add extra functionality before or after the method execution.

2. Spring AOP – Environment Setup

This chapter takes you through the process of setting up Spring AOP on Windows and Linux based systems. Spring AOP can be easily installed and integrated with your current Java environment and MAVEN by following a few simple steps without any complex setup procedures. User administration is required while installation.

System Requirements

JDK	Java SE 2 JDK 1.5 or above
Memory	1 GB RAM (recommended)
Disk Space	No minimum requirement
Operating System Version	Windows XP or above, Linux

Let us now look at the steps to install Spring AOP.

Step 1: Verify your Java Installation

First of all, you need to have Java Software Development Kit (SDK) installed on your system. To verify this, execute any of the following two commands depending on the platform you are working on.

If the Java installation has been done properly, then it will display the current version and specification of your Java installation. A sample output is given in the following table.

Platform	Command	Sample Output
Windows	Open command console and type: **\>java -version**	Java version "1.7.0_60" Java (TM) SE Run Time Environment (build 1.7.0_60-b19) Java Hotspot (TM) 64-bit Server VM (build 24.60-b09,mixed mode)
Linux	Open command terminal and type: **$java -version**	java version "1.7.0_25" Open JDK Runtime Environment (rhel-2.3.10.4.el6_4-x86_64) Open JDK 64-Bit Server VM (build 23.7-b01, mixed mode)

We assume the readers of this tutorial have Java SDK version 1.7.0_60 installed on their system.

In case you do not have Java SDK, download its current version from http://www.oracle.com/technetwork/java/javase/downloads/index.html and have it installed.

Step 2: Set your Java Environment

Set the environment variable JAVA_HOME to point to the base directory location where Java is installed on your machine. For example,

Platform	Description
Windows	Set JAVA_HOME to C:\ProgramFiles\java\jdk1.7.0_60
Linux	Export JAVA_HOME=/usr/local/java-current

Append the full path of Java compiler location to the System Path.

Platform	Description
Windows	Append the String "C:\Program Files\Java\jdk1.7.0_60\bin" to the end of the system variable PAT
Linux	Export PATH=$PATH:$JAVA_HOME/bin/

Execute the command **java -version** from the command prompt as explained above.

Step 3: Download Maven Archive

Download Maven 3.3.3 from http://maven.apache.org/download.cgi

OS	Archive Name
Windows	apache-maven-3.3.3-bin.zip
Linux	apache-maven-3.3.3-bin.tar.gz
Mac	apache-maven-3.3.3-bin.tar.gz

Step 4: Extract the Maven Archive

Extract the archive, to the directory you wish to install Maven 3.3.3. The subdirectory apache-maven-3.3.3 will be created from the archive.

OS	Location (can be different based on your installation)
Windows	C:\Program Files\Apache Software Foundation\apache-maven-3.3.3
Linux	/usr/local/apache-maven
Mac	/usr/local/apache-maven

Step 5: Set Maven Environment Variables

Add M2_HOME, M2, MAVEN_OPTS to environment variables.

OS	Output
Windows	Set the environment variables using system properties. *M2_HOME=C:\Program Files\Apache Software Foundation\apache-maven-3.3.3* *M2=%M2_HOME%\bin* *MAVEN_OPTS=-Xms256m -Xmx512m*
Linux	Open command terminal and set environment variables. *export M2_HOME=/usr/local/apache-maven/apache-maven-3.3.3* *export M2=$M2_HOME/bin* *export MAVEN_OPTS=-Xms256m -Xmx512m*
Mac	Open command terminal and set environment variables. *export M2_HOME=/usr/local/apache-maven/apache-maven-3.3.3* *export M2=$M2_HOME/bin* *export MAVEN_OPTS=-Xms256m -Xmx512m*

Step 6: Add Maven Bin Directory Location to System Path

Now append M2 variable to System Path.

OS	Output
Windows	Append the string ;%M2% to the end of the system variable, Path.
Linux	export PATH=$M2:$PATH
Mac	export PATH=$M2:$PATH

Step 7: Verify Maven installation

Now open console, and execute the following **mvn** command.

OS	Task	Command
Windows	Open Command Console	c:\> mvn --version
Linux	Open Command Terminal	$ mvn --version
Mac	Open Terminal	machine:< joseph$ mvn --version

Finally, verify the output of the above commands, which should be something as follows:

OS	Output
Windows	Apache Maven 3.3.3 (7994120775791599e205a5524ec3e0dfe41d4a06; 2015-04-22T17:27:37+05:30) Maven home: C:\Program Files\Apache Software Foundation\apache-maven-3.3.3 Java version: 1.7.0_75, vendor: Oracle Corporation Java home: C:\Program Files\Java\jdk1.7.0_75\jre Default locale: en_US, platform encoding: Cp1252
Linux	Apache Maven 3.3.3 (7994120775791599e205a5524ec3e0dfe41d4a06; 2015-04-22T17:27:37+05:30) Maven home: /usr/local/apache-maven/apache-maven-3.3.3 Java version: 1.7.0_75, vendor: Oracle Corporation Java home: /usr/local/java-current/jdk1.7.0_75/jre
Mac	Apache Maven 3.3.3 (7994120775791599e205a5524ec3e0dfe41d4a06; 2015-04-22T17:27:37+05:30) Maven home: /usr/local/apache-maven/apache-maven-3.3.3 Java version: 1.7.0_75, vendor: Oracle Corporation Java home: /Library/Java/Home/jdk1.7.0_75/jre

Step 8: Set Up Eclipse IDE

All the examples in this tutorial have been written using Eclipse IDE. So I would suggest you should have the latest version of Eclipse installed on your machine.

To install Eclipse IDE, download the latest Eclipse binaries from http://www.eclipse.org/downloads/. Once you download the installation, unpack the binary distribution into a convenient location. For example, in C:\eclipse on Windows, or /usr/local/eclipse on Linux/Unix and finally set PATH variable appropriately.

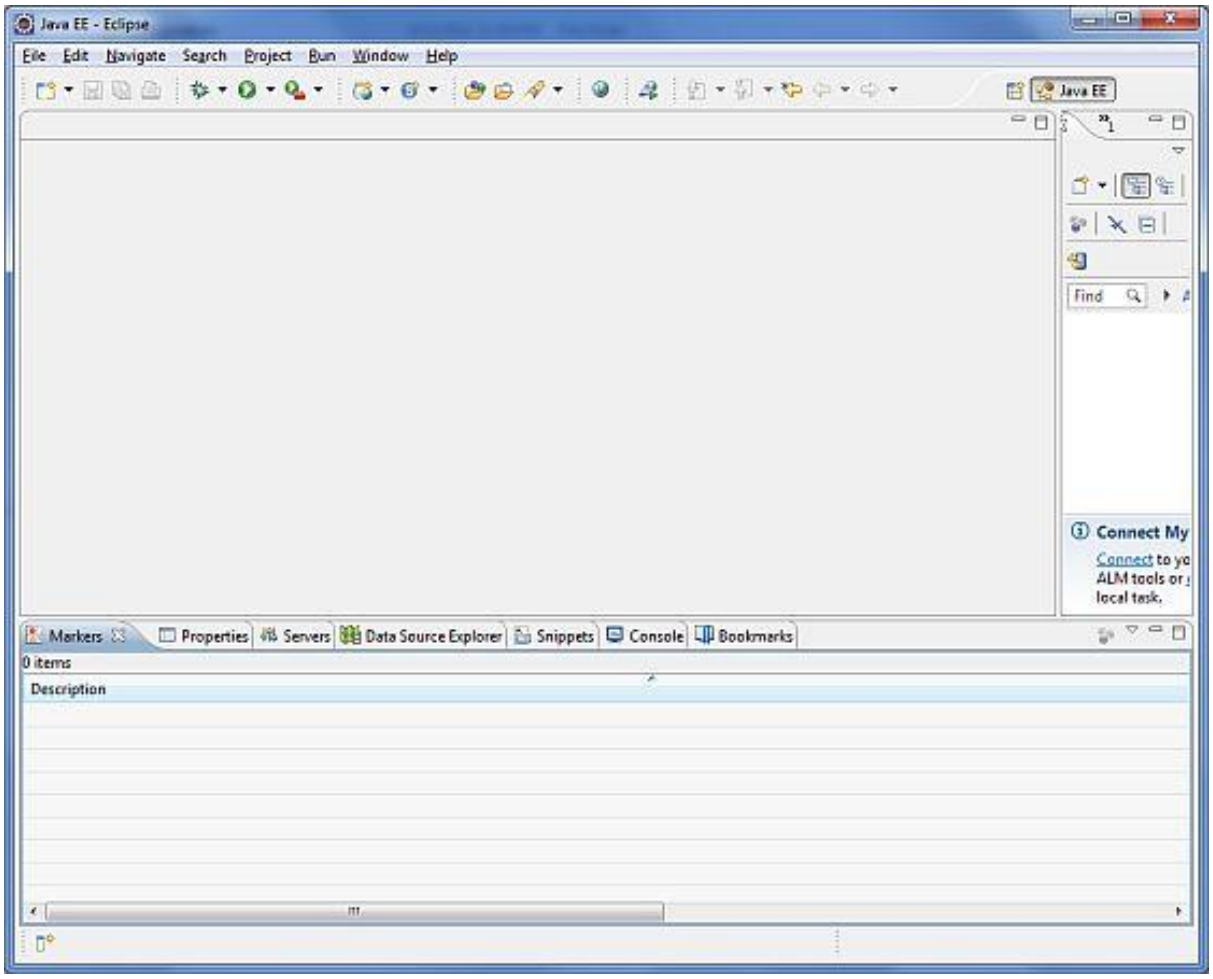
Eclipse can be started by executing the following commands on Windows machine, or you can double-click on eclipse.exe

```
%C:\eclipse\eclipse.exe
```

Eclipse can be started by executing the following commands on Unix (Solaris, Linux, etc.) machine:

```
$/usr/local/eclipse/eclipse
```

After a successful startup, if everything is fine then it should display the following result.



Once you are done with this last step, you are ready to proceed for your first AOP example, which you will see in the next chapter.

3. Spring AOP – Core Concepts

Before we start working with AOP, let us become familiar with the AOP concepts and terminologies. These terms are not specific to Spring, rather they are related to AOP.

Terms	Description
Aspect	A module which has a set of APIs providing cross-cutting requirements. For example, a logging module would be called AOP aspect for logging. An application can have any number of aspects depending on the requirement.
Join point	This represents a point in your application where you can plug-in AOP aspect. You can also say, it is the actual place in the application where an action will be taken using Spring AOP framework.
Advice	This is the actual action to be taken either before or after the method execution. This is the actual piece of code that is invoked during program execution by Spring AOP framework.
Pointcut	This is a set of one or more join points where an advice should be executed. You can specify pointcuts using expressions or patterns as we will see in our AOP examples.
Introduction	An introduction allows you to add new methods or attributes to existing classes.
Target object	The object being advised by one or more aspects. This object will always be a proxied object. Also referred to as the advised object.
Weaving	Weaving is the process of linking aspects with other application types or objects to create an advised object. This can be done at compile time, load time, or at runtime.

End of ebook preview

If you liked what you saw…

Buy it from our store @ **https://store.tutorialspoint.com**