



- [Java Core »](#)
- [Hibernate »](#)
- [Spring »](#)
- [Struts »](#)
- [Maven »](#)
- [Unit Test »](#)

Quick start Maven + Spring Example

Written on March 2, 2010 at 6:05 am by [mkyong](#)

Download this Maven + [Spring](#) example – [SpringExample.zip](#)

This quick guide example will use Maven to generate a simple [Java project](#) structure, and use Spring to retrieve the bean and print the hello world.

## Prerequisite requirement

- Basic knowledge of Maven
- Installed and configure Maven
- Installed and configured [Eclipse](#) IDE

## Tutorial ...

Maven.....

### 1. Generate project structure with Maven

In [command prompt](#), type “**mvn archetype:generate**“, choose project #15 to create a quick start Maven’s java project.

A large, empty rectangular box with a thin black border, likely intended for a screenshot or diagram.

```
E:\workspace>mvn archetype:generate
[INFO] Scanning for projects...
...
Choose a number:
```

```
(1/2/3....) 15: : 15
```

```
...
Define value for groupId: : com.mkyong.common
Define value for artifactId: : SpringExample
Define value for version: 1.0-SNAPSHOT: :
Define value for package: com.mkyong.common: : com.mkyong.common
[INFO] BUILD SUCCESSFUL
[INFO] -----
```

P.S Maven will generate all the Java's standard folders structure for you (beside resources folder, quick start archetype #15 does not contains the resources folder)

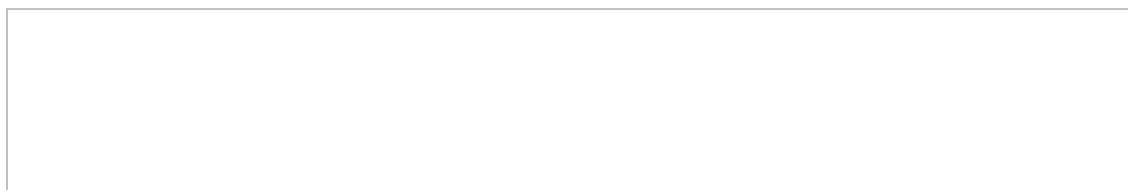
## 2. Convert to Eclipse project

Type “**mvn eclipse:eclipse**” to [convert](#) the newly generated project to Eclipse's style project

## 3. Import converted project into Eclipse IDE

In Eclipse IDE, Choose File → Import → General folder, Existing Projects into Workspace → Choose your project folder location. Done

## 4. Create a resources folder



Create a resources “**/src/main/resources**” folder, the Spring's bean xml [configuration file](#) will put here later. Maven will treat all files in this folder as resources files, and copy it to output classes automatically.

## 5. Add Spring dependency

Modify the Maven's **pom.xml** file, add support for Spring.

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
    http://maven.apache.org/maven-v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.mkyong.common</groupId>
  <artifactId>SpringExample</artifactId>
  <packaging>jar</packaging>
  <version>1.0-SNAPSHOT</version>
  <name>SpringExample</name>
  <url>http://maven.apache.org</url>

  <dependencies>
    <!-- Spring framework -->
    <dependency>
      <groupId>org.springframework</groupId>
      <artifactId>spring</artifactId>
      <version>2.5.6</version>
    </dependency>
  </dependencies>
</project>
```

Type “**mvn eclipse:eclipse**“, Maven will download the Spring dependency libraries automatically and put it into your Maven's local repository. At the same time, Maven will add the downloaded libraries into Eclipse “classpath” for dependency purpose.

Spring.....

## 6. Create a Spring bean (Java class)

Create a normal Java class (HelloWorld.java) at “src/main/java/com/mkyong/common/HelloWorld.java”. Spring’s bean is just a normal Java class, which will declare in Spring bean configuration file later.

```
package com.mkyong.common;

/**
 * Hello world!
 */
public class HelloWorld
{
    private String name;

    public void setName(String name) {
        this.name = name;
    }

    public void printHello() {
        System.out.println("Hello ! " + name);
    }
}
```

## 7. Create Bean Configuration File

Create a xml file (Spring-Module.xml) at “src/main/resources/Spring-Module.xml”. This is a Spring’s bean configuration files, which declared all the available beans in Spring IoC container.

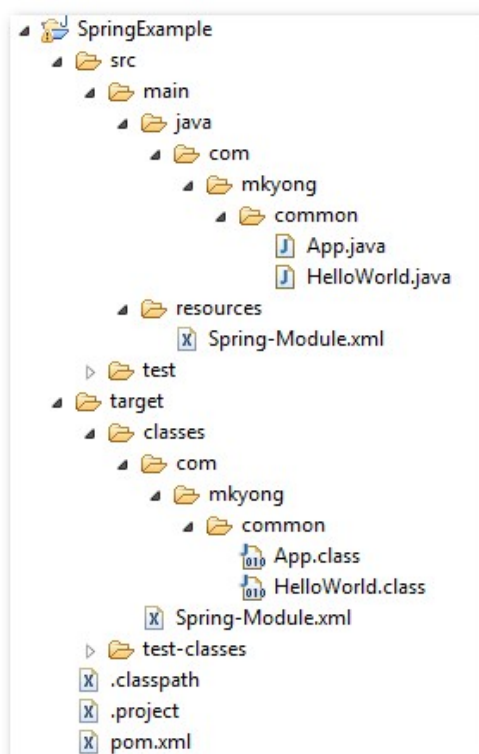
```
<beans xmlns="http://www.springframework.org/schema/beans"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-2.5.xsd">

    <bean id="helloBean" class="com.mkyong.common.HelloWorld">
        <property name="name" value="Mkyong" />
    </bean>

</beans>
```

## 8. Review project structure

Review it and make sure the folder structure as following



## Run It

Load the Spring bean configuration file (Spring-Module.xml) and retrieve it with `getBean()` method.

```
package com.mkyong.common;

import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;

/**
 * Hello world!
 *
 */
public class App
{
    public static void main( String[] args )
    {
        ApplicationContext context =
            new ClassPathXmlApplicationContext( "Spring-Module.xml" );

        HelloWorld obj = (HelloWorld) context.getBean( "helloBean" );
        obj.printHello();
    }
}
```

## Output

Hello ! Mkyong

Done ~ enjoy Spring 😊

Like

Be the first of your friends to like this.

## Popular Tutorials



- [Struts Tutorials](#)



- [Spring Tutorials](#)



- [Maven Tutorials](#)



- [Hibernate Tutorials](#)

### 3 Comments

1. [Spring Tutorials / Tutorials](#) says:  
[April 1, 2010 at 11:09 am](#)

[...] Quick start Maven + Spring Example A quick start to create a project with Maven and Spring.  
[...]

[Reply](#)

2. [Maven Tutorials / Tutorials](#) says:  
[April 1, 2010 at 1:59 am](#)

[...] Quick start Maven + Spring Example A quick start to create a project with Maven and Spring.  
[...]

[Reply](#)

3. [Maven + Spring + JDBC Example / Spring](#) says:  
[March 9, 2010 at 8:29 am](#)

[...] Maven [Tutorials](#) [\(function](#) () { var bsa = document.createElement('script'); bsa.type = 'text/javascript'; bsa.async = true; bsa.src = '//s3.buysellads.com/ac/bsa.js'; (document.getElementsByTagName('head')[0] || document.getElementsByTagName('body')[0]).appendChild(bsa); })(); Maven + Spring + JDBC Example Written on March 9, 2010 at 8:27 am by mkyong In this tutorial, you will reuse the last simple Maven + Spring hello world example. [...]

[Reply](#)

### Leave a Reply

Name (required)

Mail (will not be published) (required)

Website

☐ Notify me of followup comments via e-mail

## Popular Tutorials



- [Hibernate Tutorials](#)

Hibernate is a object/relational persistence tool for Java developers. Hibernate lets you to use object-oriented way to develop your persistent classes, It's also provide many data manipulation API like ...



- [Maven Tutorials](#)

Apache Maven is a [software](#) [project management tool](#) , not just a build tool like ant. It's provides a new concept of a project object model (POM) file, to manage project's ...



- [Apache Archiva Tutorials](#)

Apache Archiva is a Build Artifact Repository Manager, a repository management that helps to create a Maven repository in our end.

To simple, it helps to create a Maven repository ...



- [Spring Tutorials](#)

The Spring framework is a powerful and extensible Inversion of control(IoC) container, provides developers a good habit to program to interface, rather than classes to decouple your components, helps developers ...



- [JUnit Tutorials](#)

JUnit is an open source testing framework, and an instance of the xUnit architecture for unit testing frameworks. JUnit is simple and suitable used for pure unit testing, for ...



- [TestNG Tutorials](#)

TestNG (Next Generation) is a testing framework inspired from JUnit and NUnit, but introducing some new functionalities like dependency testing, grouping concept to make testing more powerful and easier to ...



- [Java XML Tutorials](#)

Java comes with two XML parsers – DOM and SAX to process XML file. The others popular third party XML parser is JDOM. Here's few examples to show how to ...



- [Java Regular Expression Tutorials](#)

Java has comprehensive support for Regular [Expression](#) functionality through the java.util.regex package. The regular expression language is easy to learn but hard to master, the better way to learn it ...



- [Struts Tutorials](#)

The Struts 1.x framework is a the most famous, classic and proven success MVC framework. Often times, you will listen something like, meaningless to learn Struts 1.x, it's a dead ...

## Recent Comments







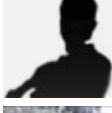

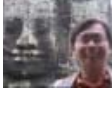
- Stephane : For a MySQL schema with 100 tables, each having some NOT NULL DEFAULT... [More](#)



- Neeraj : No need to set window.onload function at the end of page... [More](#)



- anonymous : Just go to the following site: warez-bb.org and register. Than... [More](#)

-  sahil : thanks , but There is no `getServletContext();` in `WebApplication.get()` [More](#)
-  Courtney : Hi my name is courtney and Um srry this must be a really really... [More](#)
-  jagadish : I gone through the notes and it is very good for begeers. ~Thanks,jaga [More](#)
-  mkyong : Sorry, are you tested in IE6? caused it worked fine in IE 7 & 8 [More](#)
-  dattai : that's very cool Thanks alot [More](#)
-  mkyong : use `jdom` to read the `xml` file and constructs any string format you... [More](#)
-  mkyong : Java cache? What to do with XML? [More](#)

## Authors

Hi, my name is Yong Mook Kim, person behind Mkyong.com.

This website is providing daily Java / J2EE web development tutorials, which involve Spring, Hibernate, Struts, Maven, Java Core, Unit Test...

## Friends & Links

- [China Wholesale](#)
- [Dropship](#)
- [Investment Life](#)
- [Webmaster Forum](#)
- [MySQL Tutorials](#)
- [PHP Tutorials](#)
- [ChanKelwin](#)
- [Internet Blogger](#)
- [TanWanMei](#)
- [TenthOfMarch](#)
- [Find Free Icons](#)
- [Excel Consultant](#)
- [IT Support Liverpool](#)

Copyright © 2008-2010 [Mkyong.com](#) | [RSS Feed](#) | [Privacy Policy](#) | [Write for Cash](#) | [Advertise with Us](#) | [Contact Us](#)