[Technicalkeeda.com](http://www.technicalkeeda.com/)

* [Tutorials](http://www.technicalkeeda.com/tutorials)
* [Jquery](http://www.technicalkeeda.com/spring/generate-pdf-using-spring-framework)

Top of Form



 Search

Bottom of Form

* [Login](http://www.technicalkeeda.com/spring/generate-pdf-using-spring-framework#login)

**Generate PDF Using Spring Framework**

 Posted On 2014-08-20 | [Yashwant Chavan](http://www.technicalkeeda.com/spring/generate-pdf-using-spring-framework)

In this tutorial we are going to learn how to generate PDF File Using Spring Framework + itext library. Article covers Spring application configuration, custom view definition and configuration which is required for pdf generation.

Useful to whom

Generate Pdf Using Spring Mvc tutorial is useful for beginners and experience developers. It will helps you to learn step by step with the help of below code.

There is another tutorial which will help you to generate the [Excel File Using Spring Mvc](http://www.technicalkeeda.com/spring/generate-excel-using-spring-mvc" \o "Excel File Using Spring Mvc)

Tools and Technologies

Basically we are using below technologies

1. Maven 3.0.4
2. JDK 1.6
3. Spring 3.0.5.RELEASE
4. itext 4.2.1

Maven Dependencies

We need to define required dependencies and their versions in pom.xml

<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/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>SpringExamples</groupId>

<artifactId>SpringExamples</artifactId>

<packaging>war</packaging>

<version>1.0</version>

<name>SpringExamples</name>

<description></description>

<build>

<plugins>

<plugin>

<artifactId>maven-compiler-plugin</artifactId>

<configuration>

<source>1.5</source>

<target>1.5</target>

</configuration>

</plugin>

<plugin>

<artifactId>maven-war-plugin</artifactId>

<version>2.0</version>

<configuration>

<webResources>

<resource>

<directory>WebContent</directory>

</resource>

</webResources>

</configuration>

</plugin>

</plugins>

</build>

<properties>

<spring.version>3.0.5.RELEASE</spring.version>

</properties>

<dependencies>

<dependency>

<groupId>javax.servlet</groupId>

<artifactId>servlet-api</artifactId>

<version>2.5</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-core</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-web</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-webmvc</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>javax.servlet</groupId>

<artifactId>jstl</artifactId>

<version>1.1.2</version>

</dependency>

<dependency>

<groupId>taglibs</groupId>

<artifactId>standard</artifactId>

<version>1.1.2</version>

</dependency>

<dependency>

<groupId>com.lowagie</groupId>

<artifactId>itext</artifactId>

<version>4.2.1</version>

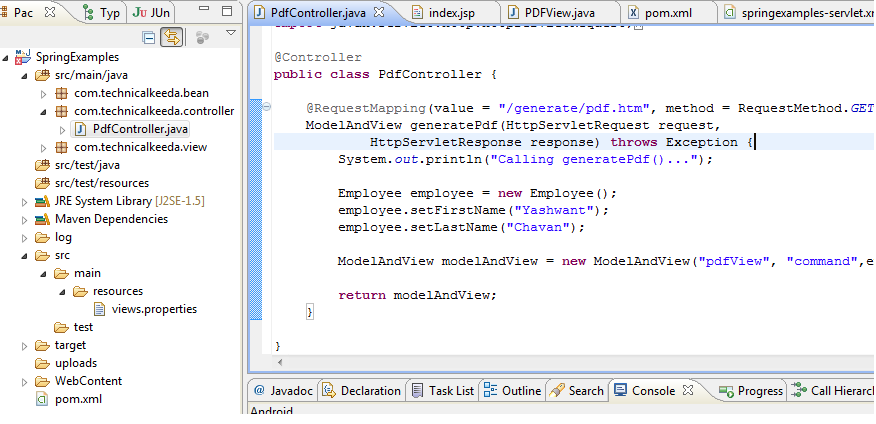
</dependency>

</dependencies>

</project>

*org.springframework* I am using spring 3.0.5.RELEASE version. Spring Framework 3 artifact (spring-core, spring-web and spring-webmvc) as it will be indexed in Maven central is listed as above.

Once you define your maven dependencies in pom.xml, please execute the maven command mvn clean install -e so that it will start downloading the respected jars files



PdfController.java

Define the PdfController is used to handle the the request to generate the PDF file using itext library

package com.technicalkeeda.controller;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.springframework.stereotype.Controller;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestMethod;

import org.springframework.web.servlet.ModelAndView;

import com.technicalkeeda.bean.Employee;

@Controller

public class PdfController {

@RequestMapping(value = "/generate/pdf.htm", method = RequestMethod.GET)

ModelAndView generatePdf(HttpServletRequest request,

HttpServletResponse response) throws Exception {

System.out.println("Calling generatePdf()...");

Employee employee = new Employee();

employee.setFirstName("Yashwant");

employee.setLastName("Chavan");

ModelAndView modelAndView = new ModelAndView("pdfView", "command",employee);

return modelAndView;

}

}

Create PdfView

Create a PdfView.java which is subclass of *org.springframework.web.servlet.view.document.AbstractPdfView* and provide the implementation of buildPdfDocument() method.

package com.technicalkeeda.view;

import java.util.Map;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.springframework.web.servlet.view.document.AbstractPdfView;

import com.lowagie.text.Chunk;

import com.lowagie.text.Document;

import com.lowagie.text.FontFactory;

import com.lowagie.text.Paragraph;

import com.lowagie.text.pdf.PdfWriter;

import com.technicalkeeda.bean.Employee;

public class PDFView extends AbstractPdfView {

protected void buildPdfDocument(Map model,

Document document, PdfWriter writer, HttpServletRequest req,

HttpServletResponse resp) throws Exception {

Employee employee = (Employee) model.get("command");

Paragraph header = new Paragraph(new Chunk("Generate Pdf USing Spring Mvc",FontFactory.getFont(FontFactory.HELVETICA, 30)));

Paragraph by = new Paragraph(new Chunk("Author " + employee.getFirstName() + " " + employee.getLastName(),FontFactory.getFont(FontFactory.HELVETICA, 20)));

document.add(header);

document.add(by);

}

}

Spring comes with AbstractPdfView abstract super class for PDF views, It helps to generate the PDF file dynamically using Bruno Lowagie's iText library. Application-specific view classes will extend this class. The view will be held in the subclass itself

Employee.java

Simple bean class used to store employee information.

package com.technicalkeeda.bean;

public class Employee {

private String firstName;

private String lastName;

public String getFirstName() {

return firstName;

}

public void setFirstName(String firstName) {

this.firstName = firstName;

}

public String getLastName() {

return lastName;

}

public void setLastName(String lastName) {

this.lastName = lastName;

}

}

Resource Bundles

Create views.properties file under the applicationÃ¢??s class-path. In eclipse it is located under /src/main/resources/views.properties.**"pdfView"** is the view name used in pdf controller new ModelAndView("pdfView", "command",employee);

views.properties

pdfView.(class)=com.technicalkeeda.view.PDFView

Spring Configuration File

Lets define the required spring configurations in springexamples-servlet.xml which is located under folder /WEB-INF/springexamples-servlet.xml

<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:context="http://www.springframework.org/schema/context"

xmlns:mvc="http://www.springframework.org/schema/mvc" 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-3.0.xsd http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-3.0.xsd http://www.springframework.org/schema/mvc http://www.springframework.org/schema/mvc/spring-mvc-3.0.xsd">

<mvc:annotation-driven />

<context:component-scan base-package="com.technicalkeeda.controller" />

<!--

<mvc:resources mapping="/css/\*\*" location="/css/" />

<mvc:resources mapping="/resources/\*\*" location="/resources/" />

-->

<bean

class="org.springframework.web.servlet.view.InternalResourceViewResolver">

<property name="order" value="2"></property>

<property name="prefix" value="/WEB-INF/jsp/" />

<property name="suffix" value=".jsp" />

</bean>

<bean id="viewResolver"

class="org.springframework.web.servlet.view.ResourceBundleViewResolver">

<property name="order" value="1"></property>

<property name="basename" value="views"></property>

</bean>

</beans>

<mvc:annotation-driven /> declares explicit support for annotation-driven MVC controllers (i.e. @RequestMapping, @Controller)

Here I am using the <context:component-scan.... tag to which have to scan my files and searching the annotations. e.g. when the controller class finds that a method is annotated by @RequestMapping("/employee/add.htm") annotation knows that this method handles the HTTP Request toward the URL ending with "/employee/add.htm".

<bean id="viewResolver"

class="org.springframework.web.servlet.view.ResourceBundleViewResolver">

<property name="order" value="1"></property>

<property name="basename" value="views"></property>

</bean>

ViewResolver is an implementation that uses bean definitions in a ResourceBundle, specified by the bundle basename. The bundle is typically defined in a properties file, located in the class path. The default bundle basename is "views". ViewResolver implements the Ordered interface, If your application contains multiple view resolvers , please provide the order value.

web.xml

Web application deployment descriptor helps to configure spring DispatcherServlet controller.

<?xml version="1.0" encoding="UTF-8"?>

<web-app version="2.4" xmlns="http://java.sun.com/xml/ns/j2ee"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee http://java.sun.com/xml/ns/j2ee/web-app\_2\_4.xsd">

<servlet>

<servlet-name>springexamples</servlet-name>

<servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>

<init-param>

<param-name>contextConfigLocation</param-name>

<param-value>/WEB-INF/springexamples-servlet.xml</param-value>

</init-param>

<load-on-startup>1</load-on-startup>

</servlet>

<servlet-mapping>

<servlet-name>springexamples</servlet-name>

<url-pattern>\*.htm</url-pattern>

</servlet-mapping>

<welcome-file-list>

<welcome-file> index.jsp </welcome-file>

</welcome-file-list>

</web-app>

index.jsp

This is default view page of the application, Simple contains the *Generate Pdf* link, which will forward your request to PdfController.

<%@ page language="java" contentType="text/html; charset=ISO-8859-1" pageEncoding="ISO-8859-1"%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

</head>

<body>

<h1>Generate / Render Pdf Using Spring Mvc</h1>

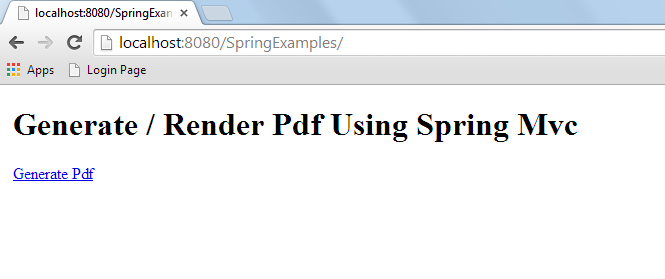
<a href="generate/pdf.htm">Generate Pdf</a>

</body>

</html>

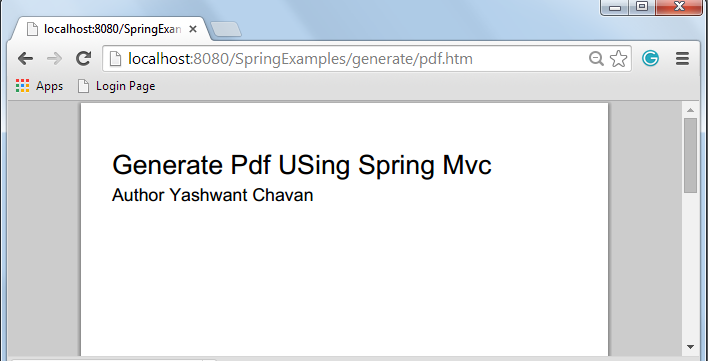
Run

Open a browser and type the url http://localhost:8080/SpringExamples/, It will display the index.jsp page, just navigate to next page.



Output

It will display the generate pdf file using itext liberabry.



Finish

Hope this Tutorial is useful to you, Please fill free to provide your comments if there is any problem. Thanks

Tags [Spring](http://www.technicalkeeda.com/tags/spring" \o "spring Tutorial)  [Spring Framework](http://www.technicalkeeda.com/tags/spring-framework" \o "spring-framework Tutorial)  [Abstractpdfview](http://www.technicalkeeda.com/tags/abstractpdfview" \o "abstractpdfview Tutorial)

Similar Articles

[Spring Framework + Jquery Ajax Request and Json Response Example](http://www.technicalkeeda.com/jquery/spring-framework-jquery-ajax-request-and-json-response-example)

[Spring Boot MVC Example](http://www.technicalkeeda.com/spring/spring-boot-mvc-example)

[Twitter Bootstrap With Spring MVC](http://www.technicalkeeda.com/bootstrap/twitter-bootstrap-with-spring-mvc)

[Hello World Spring Jmstemplate Example](http://www.technicalkeeda.com/spring/hello-world-spring-jmstemplate-example)

[Spring Email Velocity Template Example](http://www.technicalkeeda.com/spring/spring-email-velocity-template-example)

Comments

Top of Form









  by [Html Comment Box](http://www.htmlcommentbox.com/" \t "_blank)   email me replies

[[login]](https://accounts.google.com/o/oauth2/auth?scope=email&state=http%3A//www.technicalkeeda.com/spring/generate-pdf-using-spring-framework&redirect_uri=http%3A//www.htmlcommentbox.com%2Foauth2callback&response_type=code&client_id=509841468006-i7n9nufur9vg81ukmm33k0lfo3bmu0fr.apps.googleusercontent.com&approval_prompt=force)

Bottom of Form

(Feb 3, 2015) **mangita**said:

http://www.gravatar.com/avatar/64a5dd6879ece9ea40b6aac4fb22cc53?s=40&d=https%3A%2F%2Fwww.htmlcommentbox.com%2Fstatic%2Fimages%2Fgravatar.pngCould not resolve view with name 'pdfView' in servlet with name 'dispatcherServlet' i get this error massage

[http://www.htmlcommentbox.com/static/images/flag.png](javascript:hcb.flag('6245727')) [http://www.htmlcommentbox.com/static/images/like.png](javascript:hcb.like('6245727')) [http://www.htmlcommentbox.com/static/images/reply.png](javascript:hcb.reply('6245727'))

[rss](http://www.htmlcommentbox.com/rss_clean?page=http%3A%2F%2Fwww.technicalkeeda.com%2Fspring%2Fgenerate-pdf-using-spring-framework&opts=22494&mod=$1$wq1rdBcg$z3htaHpMsih/hgfONpSRw0)

Hi I am [Yashwant Chavan](https://plus.google.com/116241548760694569961?rel=author" \o "Google Plus" \t "_blank) founder of www.technicalkeeda.com, Purpose of this website to share the programming knowledge in the form post , blogs and articles.

Latest Updates

Stay updated with latest post, articles and technical discussions.

[http://feeds.feedburner.com/~fc/technicalkeeda?bg=99CCFF&fg=444444&anim=0&label=listeners](http://feeds.feedburner.com/technicalkeeda)

Top of Form



Bottom of Form

Most Popular Articles

[Jquery Form Validation Example](http://www.technicalkeeda.com/details/jquery-form-validation-example)

[Learn How to Validate Form Using Jquery, Custom Validtor Example...](http://www.technicalkeeda.com/details/jquery-form-validation-example)

[Sort collection in Java using comparable and comparator](http://www.technicalkeeda.com/details/how-to-sort-collection-using-comparable-and-comparator-in-java)

[Let have quick look on sorting technique in java, Using inbuilt sorting API and own customization sorting...](http://www.technicalkeeda.com/details/how-to-sort-collection-using-comparable-and-comparator-in-java)

[Spring Framework + Jquery Ajax Request and Json Response Example](http://www.technicalkeeda.com/jquery/spring-framework-jquery-ajax-request-and-json-response-example)

[Spring MVC 3 and JQuery is one of the great combination to perform the ajax request and response. In this tutorial we are going to create simple Employee form and submit it to spring controller using Jquery ajax request...](http://www.technicalkeeda.com/jquery/spring-framework-jquery-ajax-request-and-json-response-example)

© technicalkeeda.com 2016

[About us](http://www.technicalkeeda.com/aboutus" \o "About us) |  [Find us on Google+](https://plus.google.com/101630438296976305163) |  [Rss Feed](http://www.technicalkeeda.com/feed" \o "Subscribe to our rss feed" \t "_blank)

Loaded in 0.0092 seconds.