Struts 2 dynamic image example

By **mkyong** | February 13, 2012 | Last Updated : August 29, 2012

In this tutorial, we show you how to generate a dynamic image in Struts 2, via custom result type. For example,

```
<img src="<s:url action='ImageAction?imageId=darksouls.jpg' />" />
```

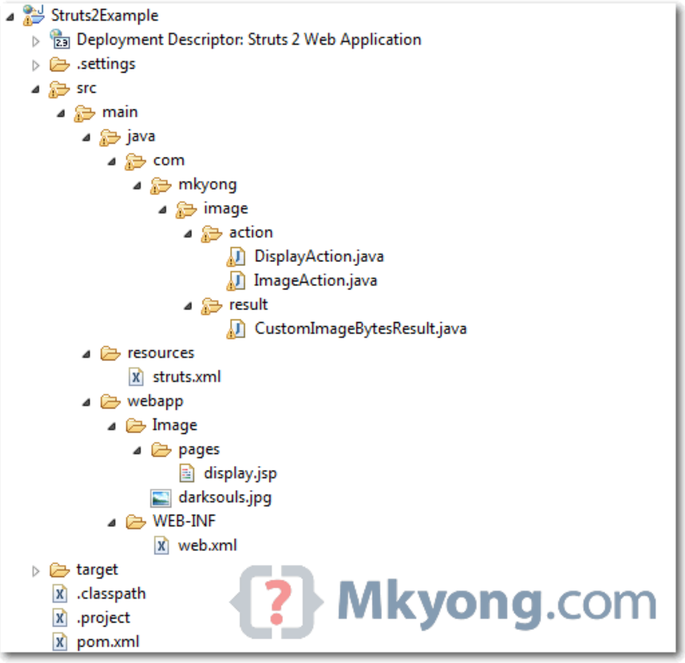
The value of `imageId` is the image name in your server path. In above case, the `ImageAction` class will output the image file based on the imageId paramater.

Tools and Struts used in this tutorial :

1. Struts 2.3.1.2
2. Eclipse 3.7
3. Maven 3

1. Directory Structure

Directory structure of this tutorial. Review the image file path.



```
Struts2Example
  Deployment Descriptor: Struts 2 Web Application
  .settings
  src
    main
      java
        com
          mkyong
            image
              action
                DisplayAction.java
                ImageAction.java
              result
                CustomImageBytesResult.java
      resources
        struts.xml
      webapp
        Image
          pages
            display.jsp
          darksouls.jpg
        WEB-INF
          web.xml
  target
  .classpath
  .project
  pom.xml
```

2. Action class

File : DisplayAction.java – A normal action, do nothing.

```
package com.mkyong.image.action;
```

```
import com.opensymphony.xwork2.ActionSupport;

public class DisplayAction extends ActionSupport  {

}
```

File : ImageAction.java – core action class, to get the image based on the provided `imageId` parameter and convert it into bytes array.

```
package com.mkyong.image.action;

import java.awt.image.BufferedImage;
import java.io.ByteArrayOutputStream;
import java.io.File;
import java.io.IOException;
import javax.imageio.ImageIO;
import javax.servlet.http.HttpServletRequest;
import org.apache.struts2.interceptor.ServletRequestAware;
import com.opensymphony.xwork2.ActionSupport;

public class ImageAction extends ActionSupport implements ServletRequestAware {

        byte[] imageInByte = null;
        String imageId;

        private HttpServletRequest servletRequest;

        public String getImageId() {
                return imageId;
        }

        public void setImageId(String imageId) {
                this.imageId = imageId;
        }

        public ImageAction() {
                System.out.println("ImageAction");
        }

        public String execute() {
                return SUCCESS;
        }

        public byte[] getCustomImageInBytes() {

                System.out.println("imageId" + imageId);

                BufferedImage originalImage;
                try {
                        originalImage = ImageIO.read(getImageFile(this.imageId));
                        // convert BufferedImage to byte array
                        ByteArrayOutputStream baos = new ByteArrayOutputStream();
                        ImageIO.write(originalImage, "jpg", baos);
                        baos.flush();
                        imageInByte = baos.toByteArray();
                        baos.close();
                } catch (IOException e) {
                        // TODO Auto-generated catch block
                        e.printStackTrace();
                }

                return imageInByte;
        }

        private File getImageFile(String imageId) {
                String filePath = servletRequest.getSession().getServletContext().getRealPath("/");
                File file = new File(filePath + "/Image/", imageId);
                System.out.println(file.toString());
                return file;
        }

        public String getCustomContentType() {
                return "image/jpeg";
        }

        public String getCustomContentDisposition() {
                return "anyname.jpg";
        }

        @Override
```

```
	public void setServletRequest(HttpServletRequest request) {
		this.servletRequest = request;

	}

}
```

3. Custom Result Type

A custom result type, and output the image to browser.

File : CustomImageBytesResult.java

```
package com.mkyong.image.result;

import javax.servlet.http.HttpServletResponse;
import org.apache.struts2.ServletActionContext;
import com.opensymphony.xwork2.ActionInvocation;
import com.opensymphony.xwork2.Result;
import com.mkyong.image.action.ImageAction;

public class CustomImageBytesResult implements Result {

	public void execute(ActionInvocation invocation) throws Exception {

		ImageAction action = (ImageAction) invocation.getAction();
		HttpServletResponse response = ServletActionContext.getResponse();

		response.setContentType(action.getCustomContentType());
		response.getOutputStream().write(action.getCustomImageInBytes());
		response.getOutputStream().flush();

	}

}
```

4. Struts.xml

See how everything is linked together.

File : struts.xml

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE struts PUBLIC
"-//Apache Software Foundation//DTD Struts Configuration 2.0//EN"
"http://struts.apache.org/dtds/struts-2.0.dtd">

<struts>

	<package name="image" namespace="/Image" extends="struts-default">

		<result-types>
			<result-type name="imageResult"
				class="com.mkyong.image.result.CustomImageBytesResult" />
		</result-types>

                                                                       isplayAction">

                                                                       n.ImageAction">
			<result name="success" type="imageResult">
			</result>
		</action>

	</package>

</struts>
```

Android Java Core ∨ Frameworks ∨ Misc ∨ Contact Us

5. JSP Page

A JSP page to display the dynamic image via `ImageAction`.

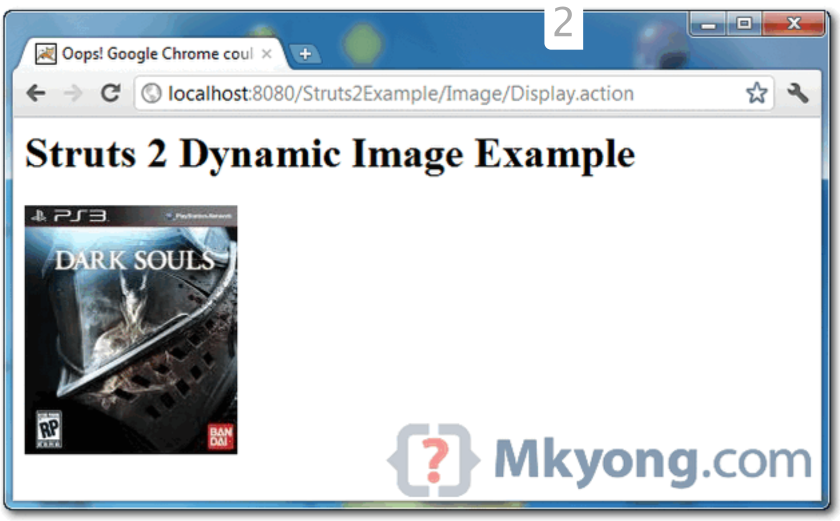
File : display.jsp

```
<%@ page contentType="text/html; charset=UTF-8"%>
<%@ taglib prefix="s" uri="/struts-tags"%>
<html>
<head></head>
<body>
        <h1>Struts 2 Dynamic Image Example</h1>

        <img src=" <s:url action='ImageAction?imageId=darksouls.jpg' />" />

</body>
</html>
```

6. Demo

See output.

http://localhost:8080/Struts2Example/Image/Display.action



Struts 2 Dynamic Image Example

Download Source Code

Download It – **Struts2-Dynamic-Image-Example.zip** (26kb)

References

1. **Struts2 display dynamic image as array of bytes**
2. **image servlet example**
3. **Struts2 URL tag example**
4. **Convert image to bytes of array in Java**
5. **Struts2 stream result example**

Tags : image struts2

About the Author

mkyong

Founder of **Mkyong.com** and **HostingCompass.com**, love Java and open source stuff. Follow him on **Twitter**, or befriend him on **Facebook** or **Google Plus**. If you like my tutorials, consider make a donation to **these charities**.

Comments

Developer Links

Android Getting Started
Google App Engine – Java
Spring 2.5.x Documentation
Spring 3.2.x Documentation
Spring 4.1.x Documentation
Java EE 5 Tutorial
Java EE 6 Tutorial
Java EE 7 Tutorial
Java 6 API
Java 7 API
Java 8 API
JSF Home Page
JSP Home Page
Maven Central Repository
Hibernate ORM
JAX-WS Home Page
JAX-RS Home Page (Jersey)

Friends & Partners

Java Code Geeks
TestNG Founder
DZone

Build Tools

Apache Ant
Apache Maven
Gradle

About Mkyong.com

Mkyong.com is for Java and J2EE developers, all examples are simple and easy to understand, of course it is well tested in my development environment.

Mkyong.com is created, written by, and maintained by Yong Mook Kim, aka Mkyong. It is built on **WordPress**, hosted by **Liquid Web**, and the caches are served by CloudFlare CDN.