

Velocity

Velocity is a templating language for Java.

For more information on Velocity itself, please visit the [Velocity website](#).



Velocity is similar to [FreeMarker](#), as both are template languages that can be used outside of a Servlet container. The framework uses FreeMarker internally since it has better error reporting. Developers may also like that FreeMarker supports JSP taglibs. However, both are good alternatives to JSP.

Getting Started

Getting started with Velocity is as simple as ensuring all the dependencies are included in your project's classpath. Other than that, [struts-default.xml](#) already configures the [Velocity Result](#).

struts.xml

```
<action name="test" class="com.acme.TestAction">
  <result name="success" type="velocity">test-success.vm</result>
</action>
```

test-success.vm

```
<html>
<head>
  <title>Hello</title>
</head>
<body>

Hello, ${name}

</body>
</html>
```

Where `name` is a property on the Action class. That's it!

There are few more details of interest, such as how templates are loaded and variables are resolved.

Template Loading

The framework looks for Velocity templates in two locations (in this order):

1. Web application
2. Class path

The ordering is designed so that a default set of templates can be placed in a JAR (perhaps shared between applications). If a template needs to be overridden, a different version can be placed in the web application.



Just the JARs, Ma'am

Unlike JSPs, templates can be loaded from a JAR. Templates are a great way to support "plugins", since the entire module can be delivered in a single JAR, and the views easily customized by the host application.

Variable Resolution

In Velocity, there are three sources for variables, searched in a specific order.

1. The value stack
2. The action context
3. Built-in variables

Since the action context is resolved after the value stack, you can reference the variable without the typical preceding marker (#) that has to be used with the JSP `s:property` tag. Omitting the marker can be convenient, but it can also trip you up, if used carelessly.

```
#surl "id=url" "value=http://www.yahoo.com"
Click <a href="${url}">here</a>!
```

The Stuts2-Velocity integration layer provides several implicit variables.

Variable	Description
stack	The value stack itself, useful for calls like <code>\${stack.findString('ognl expr')}</code>
action	The action most recently executed
response	The <code>HttpServletResponse</code>
res	Same as response
request	The <code>HttpServletRequest</code>
req	Same as request
session	The <code>HttpSession</code>
application	The <code>ServletContext</code>
base	The request's context path

Configuring Velocity

You can configure Velocity by placing configuration items in [velocity.properties](#).

Next: [Velocity Tags](#)