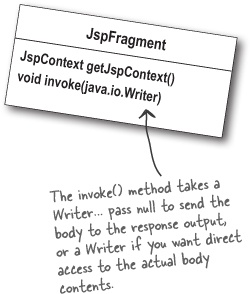
**What exactly IS a JspFragment?**

A JspFragment is an object that represents JSP code. Its sole purpose in life is to be invoked. In other words, it’s something that’s meant to *run* and generate *output*. The body of a tag that invokes a simple tag handler is encapsulated in the JspFragment object, then sent to the tag handler in the setJspBody() method.

The crucial thing you must remember about JspFragment is that it must NOT contain any scripting elements! It can contain template text, standard and custom actions, and EL expressions, but no scriptlets, declarations, or scripting expressions.



One cool thing is that since it’s an object, you can even pass the fragment around to other helper objects. And *those* objects, in turn, can get information from it by invoking the JspFragment’s *other* method—getJspContext(). And of course once you’ve got a context, you can ask for attributes. So the getJspContext() method is really a way for the tag body to get information to other objects.

Most of the time, though, you’ll use JspFragment simply to output the body of the tag to the response. You might, however, want to get access to the *contents* of the body. Notice that JspFragment doesn’t have an access method like getContents() or getBody(). You can *write* the body to something, but you can’t directly *get* the body. If you *do* want access to the body, you can use the argument to the invoke() method to ...