

Coding Actions

In the [Using Tags](#) lesson, we implemented a Logon form. In the *Coding Actions* lesson, we interpret the Logon form, and return a different result code depending on the circumstances.

If you have coded along, you can open the Logon action

```
http://localhost:8080/tutorial/Logon.action
```

and enter a likely username and password. Since we haven't given the Action any behavior, the mapping redisplay the default `Logon.jsp` page.

Let's add an Action class that will make the Logon form more interesting.

The Code

Just as an example, we can examine the username and password values. If either or both properties are empty, return INPUT, so that we can collect a valid Logon. Otherwise, return SUCCESS.

Logon.java

```
package tutorial;
import com.opensymphony.xwork2.ActionSupport;
public class Logon extends ActionSupport {

    public String execute() throws Exception {
        if (isInvalid(getUsername())) return INPUT;
        if (isInvalid(getPassword())) return INPUT;
        return SUCCESS;
    }

    private boolean isInvalid(String value) {
        return (value == null || value.length() == 0);
    }

    private String username;
    public String getUsername() {
        return username;
    }
    public void setUsername(String username) {
        this.username = username;
    }

    private String password;
    public String getPassword() {
        return password;
    }
    public void setPassword(String password) {
        this.password = password;
    }
}
```

How The Code Works

The framework automatically populates the username and password properties for us. All that's left to do is checking to see if either property is empty.

What to Remember

The Actions do the "heavy lifting" in a web application. Actions interact with data base systems and business rule engines, so that we can turn "billboard" HTML into a rich, dynamic web experience.

After doing its work, an Action returns a result code to indicate what the framework should do next. Often, the next step is to go onto the "success" result. Other times, we might need to go to an "error" result instead. In either case, the Action does not worry about generating the response, only deciding which logical result to present next.

💡 For more about Actions, see [Big Picture](#) in the Core Developers Guide.

Next	Onward to Selecting Results
Prev	Return to Using Tags