

# AcegiSpringJava5FormBased

NOTE: This information applies to an older version of Tapestry. For current information on Tapestry's Spring support, see <http://tapestry.apache.org/integrating-with-spring-framework.html>.

- [AcegiSpringJava5](#) - First part of the tutorial
- [AcegiSpringJava5Part2](#) - Second part

A modern web application uses form based logon instead of HTTP Basic authentication. Here is an attempt to add a login form to my solution presented in the first part of the tutorial.

Add these lines to `src/main/resources/META-INF/hivemodule.xml`

```
<contribution configuration-id="hivemind.ApplicationDefaults">
  <default symbol="tapestry.acegi.authenticationEntryPoint"
    value="de.zedlitz.tapestry.acegi.FormAuthenticationEntryPoint" />
  <!--      ^^^^
        you have to adjust this text according to your module id -->
</contribution>

<service-point id="FormAuthenticationEntryPoint"
  interface="org.acegisecurity.ui.AuthenticationEntryPoint">
  <invoke-factory>
    <construct class="org.acegisecurity.ui.webapp.AuthenticationProcessingFilterEntryPoint">
      <set property="loginFormUrl"
        value="/app?page=Login&service=page" />
      <set property="forceHttps" value="false" />
    </construct>
  </invoke-factory>
</service-point>
```

This tells Acegi to redirect all unauthenticated requests to `/app?page=Login&service=page`, our login page.

Create the login page `src/main/webapp/Login.html`:

```
<html>
  <head>
    <title>tapestry-acegi: login</title>
  </head>
  <body>
    <h1>tapestry-acegi: login</h1>
    <form jwcid="@Form" listener="listener:submit">
      <p>username: <input type="text" jwcid="@TextField" value="ognl:username" /></p>
      <p>password: <input type="text" jwcid="@TextField" value="ognl:password" /></p>
      <input type="submit" jwcid="@Submit" />
    </form>
  </body>
</html>
```

The logic is in the corresponding Java class `src/main/java/de/zedlitz/tapestry/acegi/Login.java`:

```

package de.zedlitz.tapestry.acegi;

import org.acegisecurity.Authentication;
import org.acegisecurity.AuthenticationException;
import org.acegisecurity.AuthenticationManager;

import org.acegisecurity.context.SecurityContextHolder;

import org.acegisecurity.providers.UsernamePasswordAuthenticationToken;

import org.acegisecurity.ui.AbstractProcessingFilter;
import org.acegisecurity.ui.savedrequest.SavedRequest;

import org.apache.tapestry.RedirectException;
import org.apache.tapestry.annotations.InjectObject;
import org.apache.tapestry.annotations.InjectPage;

import javax.servlet.http.HttpServletRequest;

public abstract class Login extends org.apache.tapestry.html.BasePage {

    public abstract String getUsername();

    public abstract String getPassword();

    @InjectObject("service:hivemind.acegi.AuthenticationManager")
    public abstract AuthenticationManager getAuthenticationManager();

    @InjectObject("service:tapestry.globals.HttpServletRequest")
    public abstract HttpServletRequest getHttpServletRequest();

    @InjectPage("Home")
    public abstract IPage getHomePage();

    public IPage submit() {
        UsernamePasswordAuthenticationToken authRequest =
            new UsernamePasswordAuthenticationToken(getUsername(), getPassword());
        Authentication authResult;

        try {
            authResult = this.getAuthenticationManager()
                .authenticate(authRequest);
        } catch (final AuthenticationException failed) {
            return null;
        }

        SecurityContextHolder.getContext().setAuthentication(authResult);

        SavedRequest savedRequest =
            (SavedRequest) this.getHttpServletRequest().getSession()
                .getAttribute(AbstractProcessingFilter.ACEGI_SAVED_REQUEST_KEY);

        if(savedRequest != null)
            throw new RedirectException(savedRequest.getFullRequestUrl());
        else
            return getHomePage();
    }
}

```

## Open Questions

- How do we retrieve the errors from Acegi and display them in the Login page?
1. Get the error message from the Exception thrown
  2. Set the message in a variable (or better yet use a message resource bundle)
  3. Retrieve the variable on the login page and display it (if there is one)

