

Tapestry5Acegi

This page describes how to use the Tapestry5 library `tapestry5-acegi` found at <http://www.localhost.nu/java/tapestry5-acegi>. It aids to help initialize `Acegi Security` and all its dependencies. There is also an example application available at <http://www.localhost.nu/svn/public/tapestry5-acegi-example>.

This page assumes the use is using maven to control it's project.

First, you need to add the correct dependencies to your `pom.xml` file. This example assumes version 1.0, however, check the website, as there might be new versions available.

```
<dependency>
  <groupId>nu.localhost.tapestry</groupId>
  <artifactId>tapestry5-acegi</artifactId>
  <version>1.0</version>
</dependency>
```

We also need to add the remote repository.

```
<repository>
  <id>localhost.nu</id>
  <url>http://www.localhost.nu/java/mvn</url>
</repository>
```

All configuration symbols provided by `tapestry5-acegi` can be overridden using the usual Tapestry methods. In this example we change the default password encoder and the url called when we failed to login.

```
public static void contributeApplicationDefaults(MappedConfiguration<String, String> configuration)
{
    configuration.add("acegi.failure.url", "/loginpage/failed");
    configuration.add("acegi.password.encoder", "org.acegisecurity.providers.encoding.Md5PasswordEncoder");
}
```

We also need some sort of authentication provider, most likely you will be using `daoAuthenticationManager` which can be used like this.

```
public static void bind(ServiceBinder binder) {
    binder.bind(UserDetailsService.class, UserDetailsServiceImpl.class);
}

public static UserDetailsService buildUserDetailsService(Session session) {
    return new UserDetailsServiceImpl(session);
}

public static void contributeProviderManager(
    OrderedConfiguration<AuthenticationProvider> configuration,
    @InjectService("DaoAuthenticationProvider") AuthenticationProvider daoAuthProvider) {
    configuration.add("daoAuthenticationProvider", daoAuthProvider);
}
```