

InjectEJB3

NOTE: This is outdated information that applies only to Tapestry 4.

Inject EJB 3 into Tapestry 4 page

this text is a copy of the mail :

"RE: Inject EJB 3 into Tapestry 4 page" by Denis Souza-3 Jan 20, 2007; 08:48pm <http://www.nabble.com/Inject-EJB-3-into-Tapestry-4-page-tf2841494.html#a8468544>

I sent an e-mail a few weeks ago asking about injecting EJBs in tapestry pages. I found a way to do it, so I'm sharing it with the list since there were other people interested. It's really very easy.

All you have to do is create a hivemind object provider that will supply you with the information you want. Something like this:

```
public class EJBOjectProvider implements ObjectProvider {  
  
    public Object provideObject(Module module, Class propertyType, String locator, Location location) {  
        return jndiLookup(locator);  
    }  
}
```

This is a bit simplified. In my case I use a caching mechanism to save on jndi lookups.

Then you have to tell hivemind about your new object provider by adding to you hivemodule.xml:

```
<service-point id="EJBOjectProvider" interface="org.apache.hivemind.service.ObjectProvider">  
    <invoke-factory>  
        <construct class="EJBOjectProvider">  
            <set-object property="applicationStateManager" value="infrastructure:applicationStateManager"/>  
        </construct>  
    </invoke-factory>  
</service-point>
```

I'm also injecting an application state manager for my own logic, but you can use anything here. Also add the following:

```
<contribution configuration-id="hivemind.ObjectProviders">  
    <provider prefix="ejb" service-id="EJBOjectProvider"/>  
</contribution>
```

That's it. Now you can inject any EJBs by simply adding something like this in your page or component:

```
@InjectObject("ejb:myBeanJndiName") public abstract MyBean getMyBean();
```

I hope it was helpful. I really don't know that much about hivemind use so it took me some tapestry source code exploring to figure this one out. I welcome any suggestions on how to improve this.

Cheers, Denis