

Adding a Normal WOComponent Page to an ERModernLook-based application

Suppose you have a component that you would like to use in your ERModernLook application. You also want it to be available as its own Tab in the Navigation bar. There are several changes that you need to make. We will walk you through the process of integrating a standard HelloWorld.wo component into your application.

Starting with the following HelloWorld WOComponent:

HelloWorld.wo

```
<wo:WForm>
  What is your name? <wo:WTextField value = "$username"/>
  <wo:WSubmitButton action="$fixUsername"/>
</wo:WForm>
<wo:WConditional condition = "$usernameExtended">
  Hello <wo:WString value = "$userNameExtended"/>
</wo:WConditional>
```

HelloWorld.java

```
import com.webobjects.appserver.WOContext;
import er.extensions.components.ERXComponent;

public class HelloWorld extends ERXComponent {

    private String username;
    private String userNameExtended;

    public HelloWorld(WOContext context) {
        super(context);
    }

    public ERXComponent fixUserName() {
        //create a new component, which will be sent to the user with it's name appended
        HelloWorld aPage = pageWithName(HelloWorld.class);

        // we do not like WObers without the name David
        String extendedUserName = "";
        if (username() != null) {
            extendedUserName = "David-" + username() + " (There is no WOBER that doesn't have
David in his firstname)";
        } else {
            extendedUserName = "Please fill in some name";
        }

        //set the User Name on the new page
        aPage.setUserNamesExtended(extendedUserName);
        return aPage;
    }

    public String username() {
        return username;
    }

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

    public String userNameExtended() {
        return userNameExtended;
    }

    public void setUserNameExtended(String userNameExtended) {
        this.userNameExtended = userNameExtended;
    }
}
```

This component takes a username and then "corrects" it to be a more socially-acceptable and, who are we kidding, much cooler one.

First we will add it to the **NavigationMenu.plist** so that it will show up in the navigation bar as its own tab:

NavigationMenu.plist entry

```
(
    {
        name = Root;
        children = ("Home", "HelloWorld");
    },
    {
        name = "Home";
        action = "session.navController.homeAction";
    }
    {
        name = "HelloWorld";
        action = "HelloWorld";
    },
)
```

Notice that we added it as an additional element of the "children" array binding of the "Root" entry, and as its own entry.

However, if you just do that clicking on the HelloWorld tab of your application will result in an error something like this:

```
Error: [<er.extensions.appserver.navigation.ERXNavigationMenuItem name: er.extensions.appserver.navigation.ERXNavigationMenuItem subcomponents: null > valueForKey()]:
lookup of unknown key: 'helloWorld'. The WComponent er.extensions.appserver.navigation.ERXNavigationMenuItem does not have
an instance variable of the name HelloWorld or _HelloWorld, nor a method of the name HelloWorld, _HelloWorld, getHelloWorld, or _getHelloWorld
```

The reason is that your navigation controller must have a method that tells it what to do when you click on the tab in the navigation bar. The "action" binding in the **NavigationMenu.plist** is the name of the method that it will try to execute.

MainNavigationController#helloWorld() method referenced by the "action" key in the plist

```
public ERXComponent helloWorld() {
    ERXComponent nextPage = (HelloWorld) myApp().pageWithName(HelloWorld.class.getName(), session().context());
    return nextPage;
}
```

However, you are still going to get the following error unless you make sure that the action binding is actually correct.

```
Error: [<er.extensions.appserver.navigation.ERXNavigationMenuItem name: er.extensions.appserver.navigation.ERXNavigationMenuItem subcomponents: null > valueForKey()]:
lookup of unknown key: 'helloWorld'. The WComponent er.extensions.appserver.navigation.ERXNavigationMenuItem does not have
an instance variable of the name HelloWorld or _HelloWorld, nor a method of the name HelloWorld, _HelloWorld, getHelloWorld, or _getHelloWorld
```

We need to tell it explicitly to call `session().navigationController().helloWorld()`

```
{
    name = "HelloWorld";
    action = "session.navController.helloWorld";
},
```

And indeed, we do get a response, but not what we expected:

What is your name?

Hello David-Chuck(There is no WOBBer that doesn't have David in his firstname)

Where is the header and the navigation bar? Those are provided by the **PageWrapper** component so we will need to add that to our component:

Add the PageWrapper component

```
<wo:PageWrapper>
  <wo:WOForm>
    What is your name? <wo:WOTextField value = "$username"/>
    <wo:WOSubmitButton action="$fixUsername"/>
  </wo:WOForm>
  <wo:WOConditional condition = "$usernameExtended">
    Hello <wo:WOString value = "$userNameExtended"/>
  </wo:WOConditional>
</wo:PageWrapper>
```

This should create all the menu's etc. Unfortunately, running this will result in the error:

Error: java.lang.NullPointerException at PageWrapper.bodyClass(PageWrapper.java:25)

The reason is that **PageWrapper** needs the component that it is wrapping to have a **D2WContext**. **ERXComponent** does not have a **D2WContext**. So we change the **HelloWorld** class to **ERD2WPage**:

```
public class HelloWorld extends ERD2WPage {
```

That, unfortunately results in this error:

Error: java.lang.NullPointerException at er.directtoweb.pages.ERD2WPage.appendToResponse(ERD2WPage.java:773)

The reason is that a page should be constructed via **D2W.factory()** instead of **myApp().pageWithName(String, WOContext)**. Looking at the code, you will find that line 644 tries to get something out of d2wcontext, which only gets created via the D2WFactory:

```
String info = "(" + d2wContext().dynamicPage() + ")";
```

But the D2wFactory creates pages via rules, if you do not add a couple of rules you will get this error:

Error: java.lang.IllegalStateException: Couldn't find the dynamic page named HelloWorld in your DirectToWeb model.task and entity is null, it seems that one model, maybe ERDirectToWeb d2w.d2wmodel is not loaded!

The reason is that every DirectToWebpage should have an "entity", and a "task". And you have to make sure that the templateNameFor[WO:task]Page for that task points to your component:

user.d2wmodel additions

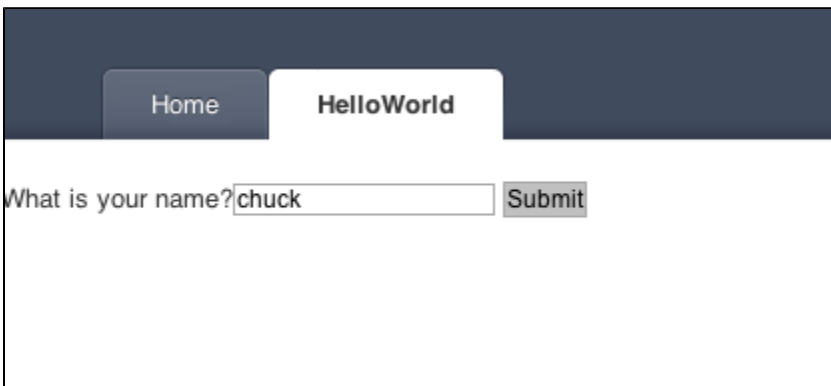
```
pageConfiguration = 'HelloWorld' => entity = "Contacten" [WO:com.webobjects.directtoweb.EntityAssignment]
pageConfiguration = 'HelloWorld' => task = "inspect" [WO:com.webobjects.directtoweb.Assignment]
pageConfiguration = 'HelloWorld' => templateNameForInspectPage = "HelloWorld" [WO:com.webobjects.directtoweb.Assignment]
```



We are nearly there. We just have to fix the navigation so the correct tab appears as selected, which we can do with a "navigationState" rule:

"navigationState" D2W Rule

```
pageConfiguration = 'HelloWorld' => navigationState = "HelloWorld" [WO:com.webobjects.directtoweb.Assignment]
```



And a submit will lead to:



The reason is that all the responses in the component should also be created as a D2WPage. Our action fixUsername() created a WOComponent but not in the D2WFactory.

We fix that by changing how the page is created:

Use the D2W#factory() to create the page

```
HelloWorld aPage = (HelloWorld) D2W.factory().pageForConfigurationNamed("HelloWorld", session());
```

And there we go:

Home HelloWorld

What is your name?

Hello David-chuck(There is no WOBBer that doesn't have David in his firstname)

Finished Code:

HelloWorld.java

```
import com.webobjects.appserver.WOContext;
import com.webobjects.directtoweb.D2W;
import er.directtoweb.pages.ERD2WPage;

public class HelloWorld extends ERD2WPage {

    private String username;
    private String userNameExtended;

    public HelloWorld(WOContext context) {
        super(context);
    }

    public ERD2WPage fixUserName() {
        //create a new component, which will be sent to the user with it's name appended
        HelloWorld aPage = (HelloWorld) D2W.factory().pageForConfigurationNamed("HelloWorld",
session());
        // we do not like WOBBers without the name David
        String extendedUserName = "";
        if (username() != null) {
            extendedUserName = "David-" + username() + "(There is no WOBBer that doesn't have David
in his firstname)";
        } else {
            extendedUserName = "Please fill in some name";
        }
        //set the User Name on the new page
        aPage.setUserNameExtended(extendedUserName);
        return aPage;
    }

    public String username() {
        return username;
    }

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

    public String userNameExtended() {
        return userNameExtended;
    }

    public void setUserNameExtended(String userNameExtended) {
        this.userNameExtended = userNameExtended;
    }
}
```

HelloWorld.wo

```
<wo:PageWrapper>
  <wo:WForm>
    What is your name? <wo:WTextField value = "$username"/>
    <wo:WSubmitButton action="$fixUsername"/>
  </wo:WForm>
  <wo:WConditional condition = "$usernameExtended">
    Hello <wo:WString value = "$userNameExtended"/>
  </wo:WConditional>
</wo:PageWrapper>
```

Additional user.d2wmodel entries

```
pageConfiguration = 'HelloWorld' => entity = "Contacten" [WO:com.webobjects.directtoweb.EntityAssignment]
pageConfiguration = 'HelloWorld' => task = "inspect" [WO:com.webobjects.directtoweb.Assignment]
pageConfiguration = 'HelloWorld' => templateNameForInspectPage = "HelloWorld" [WO:com.webobjects.directtoweb.
Assignment]
pageConfiguration = 'HelloWorld' => navigationState = "HelloWorld" [WO:com.webobjects.directtoweb.Assignment]
```

Additional entry in NavigationMenu.plist

```
{
    name = "HelloWorld";
    action = "session.navController.helloWorld";
}
```

Additional method in MainNavigationController.java

```
public WOComponent helloWorld() {
    WOComponent nextPage = D2W.factory().pageForConfigurationNamed("HelloWorld", session());
    return nextPage;
}
```