Table of Contents

It's possible to customize the standard login dialog a little bit as described in this article.

If you want full control of the login dialog, it's better to replace the dialog with a custom one. In our example, we'll create a login dialog with additional buttons to do custom authentication steps, e.g.

If you use the **standard application layout**, do following:

In your web.xml (WebContent/WEB-INF directory), search:

```
<init-param>
  <param-name>main</param-name>
  <param-value>com.sibvisions.apps.projx.ProjX</param-value>
</init-param>
```

and replace it with:

```
<init-param>
  <param-name>main</param-name>
  <param-value>com.sibvisions.apps.customlogin.MyCustomApplication</param-
value>
</init-param>
```

If you use the **corporation application layout**, do following:

In your web.xml (WebContent/WEB-INF directory), search:

```
<servlet-class>com.sibvisions.rad.ui.vaadin.server.VaadinServlet</servlet-
class>
```

and add the following:

```
<init-param>
  <param-name>Application.Login.corporation.classname</param-name>
  <param-value>com.sibvisions.apps.customlogin.MyCustomLogin</param-value>
</init-param>
```

Finally, let us create the source files (in *src.server* folder):

- com.sibvisions.apps.customlogin.**MyCustomApplication** (only for standard layout mode, but also works in corporation mode)
- com.sibvisions.apps.customlogin.**MyCustomLogin**

The Application simply creates the login, e.g.:

MyCustomApplication.java

```
package com.sibvisions.apps.customlogin;

import jvx.rad.application.genui.UILauncher;
```

```
import com.sibvisions.apps.projx.ILogin;
import com.sibvisions.apps.projx.ProjX;

public class MyCustomApplication extends ProjX
{
    public MyCustomApplication(UILauncher pLauncher) throws Throwable
    {
        super(pLauncher);
    }

    @Override
    protected ILogin createLogin() throws Throwable
    {
        if (getLauncher().isWebEnvironment())
  {
     return new MyCustomLogin();
  }

  return super.createLogin();
    }
}
```

We'll use our custom login, only in web environment. This is all.

The login class contains a little bit more code:

MyCustomLogin.java

```
package com.sibvisions.apps.customlogin;

import jvx.rad.genui.component.UIButton;
import jvx.rad.genui.container.UIPanel;
import jvx.rad.genui.layout.UIFormLayout;
import jvx.rad.ui.Style;

import com.sibvisions.apps.vaadin.web.WebLogin;
import com.vaadin.ui.Component;
import com.vaadin.ui.CssLayout;

public class MyCustomLogin extends WebLogin
{
    private UIPanel panCustomButtons;

    @Override
    protected void createLogin()
    {
        super.createLogin();

        updateUI();
    }
```

```
    @Override
    protected void handleLoginError(Throwable pException)
    {
        super.handleLoginError(pException);

        //important, because the error area will be added dynamically
        updateUI();
    }

    private void updateUI()
    {
        if (!isChangePassword())
        {
            CssLayout cssl =
(CssLayout)getInfoLabel().getParent().getParent();

            //cleanup
            if (panCustomButtons != null)
            {
cssl.removeComponent((Component)panCustomButtons.getResource());
            }
            else
            {
                UIFormLayout flCustomButtons = new UIFormLayout();
//flCustomButtons.setHorizontalAlignment(IAlignmentConstants.ALIGN_CENT
ER);
                flCustomButtons.setMargins(10, 18, 5, 18);
                flCustomButtons.setHorizontalGap(10);

                panCustomButtons = new UIPanel(flCustomButtons);

                UIButton butISA = new UIButton("ISA");
                butSAM.eventAction().addListener(this, "doISA");
                Style.addStyleNames(butISA, "default", "isa");

                UIButton butSAM = new UIButton("SAM");
                butSAM.eventAction().addListener(this, "doSAM");
                Style.addStyleNames(butSAM, "default", "sam");

                panCustomButtons.add(butISA,
flCustomButtons.getConstraints(0, 0));
                panCustomButtons.add(butSAM,
flCustomButtons.getConstraints(1, 0));
            }

            //always above the "error area"
cssl.addComponent((Component)panCustomButtons.getResource(), 2);
        }
    }
```

```
    public void doISA() throws Throwable
    {
      ((RemoteApplication)getApplication()).showInformation(null, "ISA
pressed!");
    }

    public void doSAM() throws Throwable
    {
      ((RemoteApplication)getApplication()).showInformation(null, "SAM
pressed!");
    }
}
```

Our code adds two new buttons and adds shows an information if one of the two buttons were pressed.

The stylesheet definition for the new buttons are:

```
.loginwindow .default.v-button.isa
{
  background: red;
  border-color: darkred;
}

.loginwindow .default.v-button.isa.v-pressed,
.loginwindow .default.v-button.isa:active
{
  background: red;
  border-color: darkred;
}

.loginwindow .default.v-button.isa:focus
{
  border-color: darkred;
  box-shadow: 0 0 4px darkred;
}

.v-button.isa:focus:after
{
    border-color: darkred;
    box-shadow: none;
}

.loginwindow .default.v-button.sam
{
  background: green;
  border-color: darkgreen;
}

.loginwindow .default.v-button.sam.v-pressed,
```

```
.loginwindow .default.v-button.sam:active
{
  background: green;
  border-color: dargreen;
}

.loginwindow .default.v-button.sam:focus
{
  border-color: darkgreen;
  box-shadow: 0 0 4px darkgreen;
}

.v-button.sam:focus:after
{
    border-color: darkgren;
    box-shadow: none;
}
```

If you want to create a complete new layout for the login dialog, simply write your own code in **createLogin** method and don't call **super.createLogin()**.

The **createLogin** method of WebLogin:

WebLogin.java

```
protected void createLogin()
{
    sNewPassword = null;
    sConfirmPassword = null;

    boolean bChangePwd = isChangePassword();

    cslayLoginPanel = new CssLayout();
    cslayLoginPanel.addStyleName("login-panel");
    cslayLoginPanel.setSizeFull();

    layoutMode = getLayoutMode();

    boolean bMini = layoutMode == LayoutMode.Mini;

    if (bMini)
    {
        winLogin.addStyleName("small");

        cslayLoginPanel.addStyleName("small");
    }
    else
    {
        winLogin.removeStyleName("small");
    }
```

```
    //-------------------------------------------------------------------
-----------
    // Header
    //-------------------------------------------------------------------
-----------

        lblTitle = new Label();
        lblTitle.setSizeUndefined();
        lblTitle.addStyleName("welcome");

        lblInfo = new Label();
        lblInfo.setSizeUndefined();
        lblInfo.addStyleName("info");

        lblError = new Label();
        lblError.setSizeUndefined();
        lblError.addStyleName("welcomerror");

    VerticalLayout vlayLabels = new VerticalLayout();
    vlayLabels.setSpacing(false);
    vlayLabels.setWidth("100%");
    vlayLabels.setSizeUndefined();
    vlayLabels.setMargin(true);
    vlayLabels.addStyleName("labels");

    vlayLabels.addComponent(lblTitle);
    vlayLabels.addComponent(lblError);

    if (!bMini)
    {
        vlayLabels.addComponent(lblInfo);
    }

    //-------------------------------------------------------------------
-----------
    // Username and Password
    //-------------------------------------------------------------------
-----------

    AbstractOrderedLayout aolFields;

    if (bMini)
    {
        aolFields = new VerticalLayout();
    }
    else
    {
        aolFields = new HorizontalLayout();
    }

    aolFields.setSpacing(true);
```

```
    aolFields.setMargin(true);
    aolFields.addStyleName("fields");

        tfUserName = new TextField(projx.translate("Username"));
        tfUserName.setIcon(VaadinIcons.USER);
        tfUserName.setId("UserName");
        tfUserName.addStyleName(ValoTheme.TEXTFIELD_INLINE_ICON);

        if (sUserName != null)
        {
            tfUserName.setValue(sUserName);
        }

        pfPassword = new PasswordField();

        if (bChangePwd)
        {
            pfPassword.setIcon(VaadinIcons.UNLOCK);
        }
        else
        {
            pfPassword.setIcon(VaadinIcons.LOCK);
        }

        pfPassword.setId("Password");
        pfPassword.addStyleName(ValoTheme.TEXTFIELD_INLINE_ICON);

        if (sPassword != null)
        {
            pfPassword.setValue(sPassword);
        }

        butLogin = new Button();
        butLogin.addStyleName("default");
        butLogin.addStyleName("ok");
        butLogin.setId("OK");

        if (bChangePwd)
        {
            butLogin.setCaption(projx.translate("Change"));
        }
        else
        {
            butLogin.setCaption(projx.translate("Login"));
        }

    aolFields.addComponent(tfUserName);
    aolFields.addComponent(pfPassword);

    if (sUserName == null)
    {
```

```
        tfUserName.focus();
    }
    else
    {
        pfPassword.focus();
    }

    //-----------------------------------------------------------------
-----------
    // New Password and Confirm Password
    //-----------------------------------------------------------------
-----------

    AbstractOrderedLayout aolPasswordFields;

    if (bChangePwd)
    {
        lblTitle.setValue(projx.translate(CHANGE_PASSWORD));

        pfPassword.setCaption(projx.translate("Old Password"));

        if (bMini)
        {
            aolPasswordFields = new VerticalLayout();
        }
        else
        {
            aolPasswordFields = new HorizontalLayout();
        }

        aolPasswordFields.setSpacing(true);
        aolPasswordFields.setMargin(true);
        aolPasswordFields.addStyleName("fields");
        aolPasswordFields.addStyleName("passwordfields");

        pfNewPassword = new PasswordField(projx.translate("New
Password"));
        pfNewPassword.setId("PasswordNew");
        pfNewPassword.setIcon(VaadinIcons.LOCK);
        pfNewPassword.addStyleName(ValoTheme.TEXTFIELD_INLINE_ICON);


        if (sNewPassword != null)
        {
            pfNewPassword.setValue(sNewPassword);
        }

        pfPassword.focus();

        pfConfirmPassword = new PasswordField(projx.translate("Confirm
Password"));
```

```
        pfConfirmPassword.setId("PasswordConfirm");
        pfConfirmPassword.setIcon(VaadinIcons.LOCK);
pfConfirmPassword.addStyleName(ValoTheme.TEXTFIELD_INLINE_ICON);

        AbstractOrderedLayout aolButtons;

        if (bMini)
        {
            aolButtons = new HorizontalLayout();
            aolButtons.addStyleName("nopadding");
            aolButtons.setSpacing(true);
            aolButtons.setMargin(false);
            aolButtons.setSizeFull();
        }
        else
        {
            aolButtons = aolPasswordFields;
        }

        butCancel = new Button(projx.translate("Cancel"));
        butCancel.addStyleName("default");
        butCancel.addStyleName("cancel");
        butCancel.setId("Cancel");

        aolPasswordFields.addComponent(pfNewPassword);
        aolPasswordFields.addComponent(pfConfirmPassword);

        aolButtons.addComponent(butLogin);
        aolButtons.setComponentAlignment(butLogin,
Alignment.BOTTOM_LEFT);

        Label lblOr = new Label(projx.translate("or"),
ContentMode.HTML);
        lblOr.addStyleName("or");

        if (!bMini)
        {
            aolButtons.addComponent(lblOr);
            aolButtons.setComponentAlignment(lblOr,
Alignment.BOTTOM_LEFT);
        }

        aolButtons.addComponent(butCancel);

        if (bMini)
        {
            aolButtons.setComponentAlignment(butCancel,
Alignment.BOTTOM_RIGHT);
        }
        else
        {
```

```
            aolButtons.setComponentAlignment(butCancel,
Alignment.BOTTOM_LEFT);
        }

        if (aolButtons != aolPasswordFields)
        {
            aolPasswordFields.addComponent(aolButtons);
        }

        if (projx.isConnected())
        {
            lblInfo.setValue(projx.translate("Please enter and confirm
the new password."));

            tfUserName.setEnabled(false);
        }
        else
        {
            if (projx.isForceChangePassword())
            {
                lblInfo.setValue(projx.translate("Please enter and
confirm the new password."));

                tfUserName.setEnabled(false);
                pfPassword.setEnabled(false);
            }
            else
            {
                lblInfo.setValue(projx.translate("Please enter your
username and passwords."));
            }
        }

        ShortcutListener sclEsc = new
ShortcutListener(projx.translate("Cancel"), KeyCode.ESCAPE, null)
        {
            public void handleAction(Object pSender, Object pTarget)
            {
                doCancel();
            }
        };

        butCancel.addClickListener(new ClickListener()
        {
            public void buttonClick(ClickEvent event)
            {
                doCancel();
            }
        });

        butLogin.addShortcutListener(sclEsc);
```

```
    }
    else
    {
        pfPassword.setCaption(projx.translate("Password"));

        lblTitle.setValue(projx.translate(WELCOME));

        lblInfo.setValue(projx.translate("Please enter your username
and password."));

        pfNewPassword = null;
        pfConfirmPassword = null;

        aolFields.addComponent(butLogin);
        aolFields.setComponentAlignment(butLogin,
Alignment.BOTTOM_LEFT);

        aolPasswordFields = null;
    }

    //------------------------------------------------------------------
-----------
    // ALL together
    //------------------------------------------------------------------
-----------

    cslayLoginPanel.addComponent(vlayLabels);
    cslayLoginPanel.addComponent(aolFields);

    if (aolPasswordFields != null)
    {
        cslayLoginPanel.addComponent(aolPasswordFields);
    }

    ((CssLayout)winLogin.getContent()).addComponent(cslayLoginPanel);

    //------------------------------------------------------------------
-----------
    // Event handling
    //------------------------------------------------------------------
-----------

    ShortcutListener sclEnter = new
ShortcutListener(projx.translate("Login"), KeyCode.ENTER, null)
    {
        public void handleAction(Object pSender, Object pTarget)
        {
            doOk();
        }
    };
```

```
    butLogin.addClickListener(new ClickListener()
    {
        public void buttonClick(ClickEvent event)
        {
            doOk();
        }
    });

    butLogin.addShortcutListener(sclEnter);

    updateError();
}
```

From:
https://doc.sibvisions.com/ - **Documentation**

Permanent link:
https://doc.sibvisions.com/vaadin/replace_login

Last update: **2024/11/18 10:29**