

# Another DropDownChoice Example by Adam

I had a terrible time understanding DropDownChoices. So I'm hoping by posting this someone might understand it a little bit better.

```
package com.*.webapp.wicket;

import java.io.Serializable;
import java.util.ArrayList;

import org.apache.wicket.markup.html.basic.Label;
import org.apache.wicket.markup.html.form.ChoiceRenderer;
import org.apache.wicket.markup.html.form.DropDownChoice;
import org.apache.wicket.markup.html.form.Form;
import org.apache.wicket.markup.html.form.IChoiceRenderer;
import org.apache.wicket.model.IModel;
import org.apache.wicket.model.Model;
import org.apache.wicket.model.PropertyModel;

public class TestPage extends BasePage {

    public TestPage() {
        add(new TestPageForm("form", new TestFormModel()));
    }
}

class TestPageForm extends Form {

    // one way to get the model is to set an instance variable
    private final TestFormModel m_modelObject;

    public TestPageForm(String id, TestFormModel modelObject) {
        super(id); // form with no validation

        m_modelObject = modelObject;

        // !!! Note the period in the expression
        add(new Label("label", new PropertyModel(modelObject, "comboChoice.label")));
        add(new Label("value", new PropertyModel(modelObject, "comboChoice.value")));
        // ----OR----
        // if you use this method, you'll have to create two more get methods
        // add(new Label("label", new PropertyModel(modelObject, "comboChoiceLabel")));
        // add(new Label("value", new PropertyModel(modelObject, "comboChoiceValue")));

        DropDownChoice comboChoiceDropDown = new DropDownChoice(
            "comboChoice",
            new PropertyModel(modelObject, "comboChoice"),
            new Model(getChoices()),
            new ChoiceRenderer("label", "value")
        );
        // ----OR----
        // DropDownChoice comboChoiceDropDown = new DropDownChoice(
        //     "comboChoice",
        //     new PropertyModel(modelObject, "comboChoice"),
        //     new Model(getChoices()),
        //     new TestChoiceRenderer()
        // );

        comboChoiceDropDown.setRequired(true);
        add(comboChoiceDropDown);
    }

    // This list can be created anywhere. Note that I'm adding SelectOptions to the list.
    private static ArrayList getChoices() {
        ArrayList list = new ArrayList();
        for (int i = 0; i < 10; i++) {
            list.add(new SelectOption("Show to user " + i, "for backend " + i));
        }
        return list;
    }
}
```

```

// one way to get the model
protected void onSubmit() {
    // I don't recommend System.out, but to make this example
    // as simple as possible, I'm doing that here.
    System.out.println("m_modelObject = " + m_modelObject);
}
}

class TestFormModel implements Serializable {
    // !!! Note the use of a SelectOption in my model object
    SelectOption comboChoice;

    public SelectOption getcomboChoice() {
        return comboChoice;
    }

    public void setComboChoice(SelectOption comboChoice) {
        this.comboChoice = comboChoice;
    }

    // if using this:
    // add(new Label("label", new PropertyModel(modelObject, "comboChoiceLabel")));
    // add(new Label("value", new PropertyModel(modelObject, "comboChoiceValue")));
    // you'll have to use these get methods:
    // public String getComboChoiceValue() {
    //     if (comboChoice == null)
    //         return null;
    //     return comboChoice.getValue();
    // }
    // public String getComboChoiceLabel() {
    //     if (comboChoice == null)
    //         return null;
    //     return comboChoice.getLabel();
    // }
}

// Here's my own ChoiceRenderer, but a simpler method is to use new ChoiceRenderer("label", "value")
class TestChoiceRenderer implements IChoiceRenderer {

    public Object getDisplayValue(Object object) {
        // I don't know if it can be anything other...??
        if (object instanceof SelectOption) {
            SelectOption so = (SelectOption) object;
            return so.getLabel();
        }
        return null;
    }

    public String getIdValue(Object key, int index) {
        // I don't know if it can be anything other...??
        if (key instanceof SelectOption) {
            SelectOption selectOption = (SelectOption) key;
            return selectOption.getValue();
        }
        return null;
    }
}

class SelectOption implements Serializable {

    private String m_label;
    private String m_value;

    public SelectOption(String label, String value) {
        this.m_label = label;

```

```
        this.m_value = value;
    }

    public String getLabel() {
        return m_label;
    }

    public void setLabel(String label) {
        this.m_label = label;
    }

    public String getValue() {
        return m_value;
    }

    public void setValue(String value) {
        this.m_value = value;
    }
}
```

Disclaimer: I'm just a user of Wicket and not a developer of Wicket.