

Plantilla Canigó amb Primefaces

A qui va dirigit

Aquest how-to va dirigit a tots aquells usuaris que vulguin utilitzar Primefaces en comptes de Richfaces al front-end de la seva aplicació.

Versió de Canigó

Els passos descrits en aquest document apliquen a la darrera versió del Framework Canigó 3.

Introducció

La plantilla Canigó 3 per defecte està configurada per defecte amb Richfaces. Com es pot veure a la gràfica de google trends, Primefaces ha guanyat quota de mercat respecte a Richfaces durant els últims anys. Pot ser interessant per a noves aplicacions, o inclús per a aplicacions ja realitzades, canviar a primefaces.



Seguidament expliquem quins canvis s'han de fer a l'aplicació per passar d'utilitzar Richfaces a utilitzar Primefaces.

Plantilla Canigó amb Primefaces

Exemple d'ús Spring ViewScope

Configuració de Dependències

Cal treure la dependències de Richfaces i afegir la dependència a Primefaces:

```
<!-- <dependencyManagement>-->
<!-- <dependencies>-->
<!-- <dependency>-->
<!-- <groupId>org.richfaces</groupId>-->
<!-- <artifactId>richfaces-bom</artifactId>-->
<!-- <version>${org.richfaces.bom.version}</version>-->
<!-- <type>pom</type>-->
<!-- <scope>import</scope>-->
<!-- </dependency>-->
<!-- </dependencies>-->
<!-- </dependencyManagement>-->

<!-- <dependency>-->
<!-- <groupId>org.richfaces.ui</groupId>-->
<!-- <artifactId>richfaces-components-ui</artifactId>-->
<!-- </dependency>-->
<!-- <dependency>-->
<!-- <groupId>org.richfaces.core</groupId>-->
<!-- <artifactId>richfaces-core-impl</artifactId>-->
<!-- </dependency>-->

<dependency>
  <groupId>org.primefaces</groupId>
  <artifactId>primefaces</artifactId>
  <version>5.0</version>
</dependency>
```

Al canviar les llibreries s'ha de refer la classe LogBean.java amb les llibreries de primefaces:

```
package cat.gencat.howtooctubre.bean;

import java.util.ArrayList;
import java.util.Iterator;
import java.util.List;
import java.util.StringTokenizer;
import javax.faces.application.FacesMessage;
import javax.faces.context.FacesContext;
import javax.faces.model.SelectItem;
import org.apache.commons.logging.Log;
import org.apache.commons.logging.LogFactory;
import org.primefaces.event.NodeSelectEvent;
import org.primefaces.model.DefaultTreeNode;
import org.primefaces.model.TreeNode;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Component;
import cat.gencat.ctti.canigo.arch.core.logging.jmx.Log4jManagerImpl;

@Component("LogBean")
public class LogBean {

    private static final Log log = LogFactory.getLog(LogBean.class);

    //LOG LEVELS
    private static final String LOG_DEBUG = "debug";
    private static final String LOG_INFO = "info";
    private static final String LOG_WARN = "warn";
    private static final String LOG_ERROR = "error";
    private static final String LOG_FATAL = "fatal";

    @Autowired
    private Log4jManagerImpl log4jManager;

    private String logLevel = "";
    private String addAppender = "";
```

Plantilla Canigó amb Primefaces

```
//TreeNode
private DefaultTreeNode treeNode = null;
private static final String treeNodeName = "Appenders";
public String getLogLevel() {
    return logLevel;
}
public void setLogLevel(String logLevel) {this.logLevel = logLevel;}
public String getAddAppender() {return addAppender;}
public void setAddAppender(String addAppender) {this.addAppender = addAppender;}
public DefaultTreeNode getTreeNode() { if (treeNode == null) {
    loadTree();
}return treeNode; }
public void addLogLevel() {
    FacesMessage facesMessage = null;
    String resultat = "";
    if (getAuthorizedLevels().contains(logLevel) && addAppender != null && !addAppender.equals("")) {
        try {
            log4jManager.setLogLevel(addAppender, logLevel);
            addNode(addAppender, treeNode);
            resultat = "S'han canviat l'appender '" + addAppender + "' al nivell '" + logLevel + "'.";
        } catch (Exception e) {
            log.info("No s'ha pogut canviar l'appender del servei '" + logLevel + "'.");
        }
        log.debug(resultat);
        facesMessage = new FacesMessage(FacesMessage.SEVERITY_INFO, resultat, null);
        FacesContext.getCurrentInstance().addMessage("logLevelForm", facesMessage);
    } else {
        facesMessage = new FacesMessage(FacesMessage.SEVERITY_ERROR, "El appender i;ks obligatori", null);
        FacesContext.getCurrentInstance().addMessage("logLevelForm", facesMessage);
    }
}
public void changeAllLogLevels() {
    FacesMessage facesMessage = null;
    String resultat = "";
    if (getAuthorizedLevels().contains(logLevel)) {
        String[] activeLoggerNames = log4jManager.getActiveLoggerNames();
        int logsOK = 0;
        int logsTotals = 0;
        for (int i=0; i< activeLoggerNames.length; i++) {
            try {
                log4jManager.setLogLevel(activeLoggerNames[i], logLevel);
            } catch (Exception e) {
                log.info("No s'ha pogut canviar l'appender del servei '" + logLevel + "'.");
            } finally {
                logsTotals++;
            }
            logsOK++;
        }
        resultat = "S'han canviat " + logsOK + " de " + logsTotals + " appenders.";
        log.debug(resultat);
        facesMessage = new FacesMessage(FacesMessage.SEVERITY_INFO, resultat, null);
        FacesContext.getCurrentInstance().addMessage("logLevelForm", facesMessage);
    } else {
        facesMessage = new FacesMessage(FacesMessage.SEVERITY_ERROR, "El nivell de log es obligatori", null);
        FacesContext.getCurrentInstance().addMessage("logLevelForm", facesMessage);
    }
}
public List<SelectItem> getLevels() {
    List<SelectItem> levels = new ArrayList<SelectItem>();
    List<String> authorizedLevels = getAuthorizedLevels();
    for (int i=0; i<authorizedLevels.size(); i++) {
        levels.add(new SelectItem(authorizedLevels.get(i), authorizedLevels.get(i)));
    }
    return levels;
}
public void processSelection(NodeSelectEvent event) {
    DefaultTreeNode currentNode = (DefaultTreeNode) event.getTreeNode();
    String appender = null;
    while (!(String)currentNode.getData().equals(treeNodeName)) {
        if (appender == null) {
            appender = (String)currentNode.getData();
        } else {
            appender = (String)currentNode.getData() + "." + appender;
        }currentNode = (DefaultTreeNode)currentNode.getParent();
    }
    addAppender = appender;
    String level = log4jManager.getLogLevel(appender).toLowerCase();
    logLevel = level;
}
```

Plantilla Canigó amb Primefaces

```
private void loadTree() {
    treeNode = new DefaultTreeNode(treeNodeName, null);

    String[] activeLoggerNames = log4jManager.getActiveLoggerNames();
    for (int i=0; i< activeLoggerNames.length; i++) {
        System.out.println("activeLoggerNames["+i+"]: " + activeLoggerNames[i]);
        addNode(activeLoggerNames[i], treeNode);
    }
}

private void addNode(String completeChildValue, DefaultTreeNode node) {

    DefaultTreeNode actualTreeNode = node;

    StringTokenizer st = new StringTokenizer(completeChildValue, ".");

    while (st.hasMoreTokens()) {
        String aux = st.nextToken();

        DefaultTreeNode childNode = getChildNode(actualTreeNode, aux);
        if (childNode == null) {
            //Si no contiene ese nodo lo creamos y se lo añadimos
            childNode = new DefaultTreeNode(aux, actualTreeNode);
            actualTreeNode.getChildren().add(contador++, childNode);
        }
        actualTreeNode = childNode;
    }

    private DefaultTreeNode getChildNode(DefaultTreeNode parent, String childValue) {
        Iterator<TreeNode> it = parent.getChildren().iterator();
        while (it.hasNext()) {
            DefaultTreeNode entry = (DefaultTreeNode)it.next();
            if (((String)entry.getData()).equals(childValue)){
                return entry;
            }
        }
        return null;
    }

    private List<String> getAuthorizedLevels() {
        List<String> authLevels = new ArrayList<String>();
        authLevels.add(LOG_DEBUG);
        authLevels.add(LOG_INFO);
        authLevels.add(LOG_WARN);
        authLevels.add(LOG_ERROR);
        authLevels.add(LOG_FATAL);
        return authLevels;
    }
}
```

I es pot eliminar la classe DefaultCanigóTreeNodeImpl que ja no s'utilitza.

Web.xml

Treure la configuració de Richfaces

```
<!-- <context-param>-->
<!-- <param-name>org.richfaces.skin</param-name>-->
<!-- <param-value>plain</param-value>-->
<!-- </context-param>-->
```

Canvi de tags

Primer de tot hem de treure de tots els llocs (jsf i xhtml) on apareguin els tags de richfaces:

```
xmlns:a4j="http://richfaces.org/a4j"
xmlns:rich="http://richfaces.org/rich"
```

Plantilla Canigó amb Primefaces

I afegir el tag de primefaces als fitxers jsf i xhtml

```
xmlns:p="http://primefaces.org/ui"
```

Després s'han de substituir els tags `<a4j>` i `<richfaces>` utilitzats a l'aplicació plantilla pels seus equivalents que proporciona Primefaces `<p>`:

Exemples:

```
<a4j:commandLink rendered="{menu.link == null}" value="{msg[menu.value]}"></a4j:commandLink>  
s'ha de substituir per:  
<p:commandLink rendered="{menu.link == null}" value="{msg[menu.value]}"></p:commandLink>  
  
<a4j:commandButton value="{msg.canigoSearch}" action="{logReaderBean.execute}"  
reRender="logList" />  
s'ha de substituir per:  
<p:commandButton value="{msg.canigoSearch}" actionListener="{logReaderBean.execute}"  
update="logList" />  
  
<rich:list value="{menuBean.menu.default}" id="filAriadna" var="menu" type="unordered">  
s'ha de substituir per:  
<p:dataList value="{menuBean.menu.prova}" id="filAriadna" var="menu" type="unordered">  
  
<rich:togglePanel switchType="client" label="{msg.errorStackTraceLabel}" opened="false">  
s'ha de substituir per:  
<p:panel toggleable="true" header="{msg.errorStackTraceLabel}" collapsed="true">  
  
<a4j:outputPanel ajaxRendered="true" layout="block">  
s'ha de substituir per:  
<p:outputPanel autoUpdate="true" >  
  
<rich:calendar value="{logReaderBean.date}" locale="ca_ES"  
popup="true" datePattern="d/MM/yyyy cellWidth="24px" cellHeight="22px" style="width:200px"/>  
s'ha de substituir per:  
<p:calendar value="{logReaderBean.date}" locale="ca_ES" mode="popup" pattern="d/MM/yyyy"  
style="width:200px"/>  
  
<rich:tree id="tree" nodeType="log" var="item"  
value="{LogBean.treeNode}" toggleType="client"  
selectionType="ajax"  
rowKeyConverter="org.richfaces.IntegerSequenceRowKeyConverter"  
selectionChangeListener="{LogBean.processSelection}" >  
<rich:treeNode type="log">  
#{item.data}  
</rich:treeNode></rich:tree>  
s'ha de substituir per  
<p:tree id="tree" var="item" value="{LogBean.treeNode}" selectionMode="single">  
<p:ajax event="select" listener="{LogBean.processSelection}"/>  
<p:treeNode expandedIcon="ui-icon ui-icon-folder-open" collapsedIcon="ui-icon ui-icon-  
folder-collapsed">  
<h:outputText value="{item}"/>  
</p:treeNode>  
</p:tree>  
  
<rich:dataTable> s'ha de substituir per <p:dataTable>  
<rich:column> s'ha de substituir per <p:column>
```

Plantilla Canigó amb Primefaces

Estils

Es crea un nou fitxer primefaces-custom.css per adaptar els estils de Primefaces a Canigó:

```
/** override primefaces style */  
  
.ui-datalist-empty-message {  
    display: none !important;  
}  
  
.ui-widget-content {  
    border: none !important;  
    background: none !important;  
}  
  
.ui-tree {  
    width: 100%!important;  
}  
  
.ui-datatable-odd {  
    background-color: #EBEBEB !important;  
}
```

I afegim el fitxer a import.css

```
/* CSS Document */  
/*  
NOTA CSS:  
inclou els css que seran utilitzats a l'aplicació  
*/  
@import url(dgac/layout.css);  
@import url(dgac/basic.css);  
@import url(dgac/navegacio.css);  
@import url(dgac/forms.css);  
@import url(richfaces.css);  
@import url(primefaces-custom.css);
```

Per a que agafi els css de l'aplicació quan hi hagi conflictes amb primefaces es pot utilitzar la etiqueta !important;

Per exemple al fitxer basic.css hem de remarcar com a important el hover de les nostres taules:

```
#contentMain .dgac tr:hover {  
    background-color: #D9D9D9 !important;  
}
```

JS

Degut a les diferències de codi final entre Richfaces i Primefaces al menú s'ha de modificar el fitxer **innerNavigation.class.js**

Plantilla Canigó amb Primefaces

```
function InnerNavigation(){};
InnerNavigation.prototype = {
  klass: {
    currentItem: 'selected',
    currentTarget: 'opened',
    closeTarget: 'closed'
  },
  targetedItems: [],
  init: function(selector){
    this.navigation = $(selector);
    this.targetedItems = this.navigation.find("LI:has(UL) > A");

    extend(arguments[1], this.klass);

    this.fire();
    return this;
  },
  fire: function(){
    this.targetedItems.click(function(event){ this.toggleNav(event) }.bind(this));
    this.navigation.find("LI:not(.selected)").find("UL").addClass("closed");
    this.navigation.find("LI.selected > UL").addClass("opened");
  },
  toggleNav: function(event){
    var target = event.target.parentNode || event.srcElement.parentNode,
        that = this;
    $(target).toggleClass("selected");
    if(!$ (target).children("div").children().children("ul").is(":visible"))
      $(target).children("div").children().children("ul").slideDown(function(){
    $(this).addClass(that.klass.currentTarget).removeClass(that.klass.closeTarget);
      $(this).removeAttr("style");
    });
    else
      $(target).children("div").children().children("ul").slideUp(function(){
    $(this).addClass(that.klass.closeTarget).removeClass(that.klass.currentTarget);
      $(this).removeAttr("style");
    });
    return false;
  }
}
```

Plantilla Canigó amb Primefaces

Resultat

CSCanigó

Usuari: admin Perfil: [ROLE_ADMIN, ROLE_USER]

Plantilla Canigó

[Català](#) [Castellà](#) [Anglès](#)

Apartat 1

Apartat.1 > Administració

- Subapartat 1
- Subapartat 2
- Subapartat 3
- Subapartat 4

Administració

- Monitorització
- Canviar nivell logs
- Exposar propietats
- Mòduls carregats

Nom	Versió
canigo.web.jsf	[1.2.0]
canigo.core	[3.1.1]
canigo.web.core	[1.1.0]
canigo.security	[1.1.0]