

Fil d'Ariadna

A qui va dirigit

Aquest how-to va dirigit a tots aquells usuaris que vulguin fer servir a la seva aplicació Canigó 3 un fil d'Ariadna.

Versió de Canigó

Els passos descrits en aquest document apliquen a la darrera versió del Framework Canigó 3. La implementació de l'exemple proposat és compatible per JSF2 / Richfaces 4.0.0.

Introducció

Fins al moment, la plantilla de Canigó 3 està proporcionant un model d'Ariadna que només renderitza dinàmicament les opcions dels primers nivell del menú lateral. Donat que aquesta implementació no es considera que cobreixi les necessitats del usuaris, s'ha ampliat la seva funcionalitat, proporcionant les eines necessàries per tal que els usuaris puguin definir-la i gestionar-la des de l'aplicació.

Fil d'Ariadna

Exemple d'ús d'Ariadna

Configuració de Dependències

No cal afegir cap nova dependència però cal assegurar que el pom.xml carrega la darrera versió publicada del mòdul canigo.web.jsf. A data de la redacció d'aquest howTo, aquesta és la 1.1.1:

pom.xml

```
<canigo.web.jsf.version>[1.1.0,1.2.0]</canigo.web.jsf.version>
...
<dependency>
  <groupId>cat.gencat.ctti</groupId>
  <artifactId>canigo.web.jsf</artifactId>
  <version>${canigo.web.jsf.version}</version>
</dependency>
...
```

Definició i configuració del Bean de l'Ariadna

Caldrà definir el Bean que gestionarà el fil d'Ariadna. Aquest Bean el definirem a un nou fitxer que a l'exemple anomenarem app-custom-breadcrumb.xml i que caldrà incloure'l dins de:

src/main/resources/spring/app-custom-breadcrumb.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:context="http://www.springframework.org/schema/context"
  xmlns:aop="http://www.springframework.org/schema/aop"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:p="http://www.springframework.org/schema/p"
  xmlns:util="http://www.springframework.org/schema/util"
  xsi:schemaLocation="http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans/spring-beans-3.0.xsd
    http://www.springframework.org/schema/aop
    http://www.springframework.org/schema/aop/spring-aop-3.0.xsd
    http://www.springframework.org/schema/context
    http://www.springframework.org/schema/context/spring-context-3.0.xsd
    http://www.springframework.org/schema/util
    http://www.springframework.org/schema/util/spring-util-3.0.xsd">

  <bean lazy-init="true" id="breadCrumbBean"
    class="cat.gencat.ctti.canigo.arch.web.jsf.BreadCrumbBean">
    <!--Opcional, per defecte val 10 -->
    <property name="maxBreadCrumbOptions" value="5"/>
    <!--Opcional, per defecte no s'afegeix cap enllaç inicial -->
    <property name="breadcrumbList" ref="l1listat"/>
  </bean>

  <bean id="l1listat" class="java.util.ArrayList">
    <constructor-arg>
      <list>
        <ref bean="home" />
      </list>
    </constructor-arg>
  </bean>

  <bean id="home" class="cat.gencat.ctti.canigo.arch.web.jsf.BreadCrumbObject">
    <property name="label" value="Home" />
    <property name="key" value="" />
    <property name="url" value="/views/index.xhtml" />
  </bean>
</beans>
```

Fil d'Ariadna

- `breadCrumbBean` : Bean que conté els mètodes necessaris per gestionar els enllaços de l'Ariadna.
- `maxBreadCrumbOption`: Serveix per fixar el número màxim d'enllaços que pot arribar a contenir l'Ariadna. Es tracta d'un paràmetre opcional, si no s'indica cap valor, per defecte val 10. En arribar al límit no es podran afegir més enllaços.
- `breadcrumbList`: Serveix per definir els enllaços inicials amb els que apareix l'Ariadna al arrencar l'aplicació. Es tracta d'un paràmetre opcional. Si no s'indica per defecte al arrencar l'aplicació no apareixerà cap enllaç a l'Ariadna.
- `lListat`: N'indica una llista de d'enllaços que es mostraran d'inici a l'Ariadna. Aquest bean és referenciat per el `breadcrumbList`. A l'exemple s'ha inclòs l'enllaç de la Home.
- `home`: Aquest bean representa l'enllaç de la Home i es troba inclòs dins el bean `lListat`. És un objecte de la classe `BreadCrumbObject` i els atributs que cal d'informar són:
 - `label`: el Literal que es mostrarà al menú.
 - `key`: La clau que ha de cercar als fitxer d'internacionalització de missatges.
 - `url`: La url a la que conduirà l'enllaç.Nomes cal informar `label` o `key`. Si s'indiquen tots 2 valors, la plantilla d'Ariadna que es proporciona (`ariadna.jsf`) intentarà fer servir la `key` per localitzar el missatge.

Definir la vista

Cal definir el fitxer `jsf/xhtml` que contindrà l'Ariadna. Es pot fer sobreescrivre el mateix arxiu que per a la versió anterior de l'Ariadna (`ariadna.jsf`) o bé crear-ne un de nou i referenciar-ho al `template.jsf`. A l'exemple l'hem sobreescrit. Cal assegurar que el tenim referenciat correctament al `template.jsf`:

`src/main/webapp/views/layouts/template.jsf`

```
...  
<div id="contentWrapper" class="withNav">  
  ...  
  <div id="breadCrumb">  
    <ui:insert name="ariadna">  
      <ui:include src="/views/includes/ariadna.jsf" />  
    </ui:insert>  
  </div>  
  ...  
</div>  
...
```

`src/main/webapp/views/includes/ariadna.jsf`

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="yes" ?>  
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  
<html xmlns="http://www.w3.org/1999/xhtml" xmlns:ui="http://java.sun.com/jsf/facelets"  
  xmlns:h="http://java.sun.com/jsf/html" xmlns:a4j="http://richfaces.org/a4j"  
  xmlns:rich="http://richfaces.org/rich">  
  <ui:component>  
    <rich:list value="#{breadCrumbBean.breadcrumbList}" id="filAriadna" var="opcion"  
      type="unordered">  
      <h:link rendered="#{opcion.url != null}" outcome="#{opcion.url}"  
        value="#{(opcion.key != null)?msg[opcion.key]:opcion.label}" />  
      <a4j:commandLink rendered="#{opcion.url == null}"  
        value="#{(opcion.key != null)?msg[opcion.key]:opcion.label}" />  
    </rich:list>  
  </ui:component>  
</html>
```

Fil d'Ariadna

Es poden distingir els següents components:

<rich:list>

```
<rich:list value="#{breadCrumbBean.breadcrumbList}" id="filAriadna" var="opcio" type="unordered">
```

Aquest component itera el llistat d'enllaços del bean `breadCrumbBean`.

<h:link>

```
<h:link rendered="#{opcio.url != null}" outcome="#{opcio.url}" value="#{(opcio.key !=null)?msg[opcio.key]:opcio.label}" />
```

Amb aquest component es renderitzen els enllaços definits al llistat `breadcrumbList`. A destacar els atributs:

- `rendered`: Segons la configuració proposada, només renderitzarem l'enllaç si hi ha definida una url per navegar.
- `outcome`: Url a la que haurà de navegar.
- `value`: El literal que es mostrarà per l'enllaç. La configuració proposada agafa en primer lloc la clau definida per als missatges d'internacionalització per cercar el missatge. Si no hi ha clau definida, agafa el literal.

<h:commandLink>

```
<a4j:commandLink rendered="#{opcio.url == null}" value="#{(opcio.key !=null)?msg[opcio.key]:opcio.label}"/>
```

Aquest component es presenta com a alternativa per a renderitzar enllaços no-navegables. En aquests casos només mostrarà el literal o valor de la clau presentada.

Ús de l'Ariadna

El Bean que gestiona l'Ariadna admet les següents operacions:

- **setBreadcrumbLevel**(int level, String key, String label, String url): Estableix el literal (valor de la *key / label*) i la navegació (*url*) de l'enllaç que es troba a la posició *level*. Si *level* supera el número de enllaços totals, no fa cap acció tret de reflexar al log un error.
- **addBreadcrumbLevel**(int level, String key, String label, String url): Afegeix a la posició *level* un nou enllaç, fent servir el literal (valor de la *key / label*) i la navegació (*url*) indicades. Com a resultat d'aquesta acció s'incrementa el número d'enllaços totals en 1 donat que s'ha produït un desplaçament al llistat a partir de la posició *level*. Si *level* supera el número de enllaços actuals, no fa cap acció tret de reflexar al log un error.
- **addBreadcrumbLevel**(String key, String label, String url): Afegeix a la cua de l'Ariadna un nou enllaç, fent servir el literal (valor de la *key / label*) i la navegació (*url*) indicades. Com a resultat d'aquesta acció, s'incrementa el número d'enllaços totals en 1.

Fil d'Ariadna

- **removeBreadCrumbLevel(int level):**
Esborra l'enllaç que es troba a la posició *level*. Com a resultat d'aquesta acció es redueix el número d'enllaços totals en 1 donat que s'ha produït un desplaçament al llistat a partir de la posició *level*.
Si *level* supera el número de enllaços actuals, no fa cap acció tret de reflexar al log un error.
- **removeBreadCrumbLevel():**
Esborra el darrer enllaç de l'Ariadna. Com a resultat d'aquesta acció, es redueix el número d'enllaços totals en 1.
- **keepNFirstsBreadCrumbLevels(int level):**
Esborra tots els enllaços que es trobin a partir de la posició *level*, o el que és el mateix, manté els *level* primers enllaços. Com a resultat d'aquesta acció es redueix el número total d'enllaços en $([\text{total enllaços}] - \text{level})$.
Si *level* supera el número de enllaços actuals, no fa cap acció tret de reflexar al log un error.
- **getBreadCrumbLevel(int level):**
Obté un objecte de tipus *BreadCrumbObject* que conté la informació de l'enllaç a la posició *level*.
Si *level* supera el número de enllaços actuals, no fa cap acció tret de reflexar al log un error.

Ús de l'Ariadna des de les vistes

Es pot fer ús del bean configurat amb les accions esmentades directament des de les vistes. Per exemple:

Amb aquesta acció estaríem afegint un enllaç a la darrera posició de l'Ariadna:

```
<h:commandButton value="#{msg['addPage.btn']}"  
  action="#{breadCrumbBean.addBreadCrumbLevel(null,'Vista 1','/views/vista1.xhtml)"/>
```

El resultat:

[Home](#) > [Vista 1](#)

Amb aquesta acció estaríem afegint un enllaç a la posició 1 de l'Ariadna, entre 'Home' i 'Vista1':

```
<h:commandButton value="#{msg['addBetweenPage.btn']}"  
  action="#{breadCrumbBean.addBreadCrumbLevel(1,null,'Vista 0','/views/vista0.xhtml)"/>
```

El resultat:

[Home](#) > [Vista 0](#) > [Vista 1](#)

Amb aquesta acció estaríem sobreescrivint l'enllaç de la posició 1 de l'Ariadna ('Vista 0' -> 'Enllaç'):

```
<h:commandButton value="#{msg['setPage.btn']}"  
  action="#{breadCrumbBean.setBreadCrumbLevel(1,null,'Enllaç','/views/vista1.xhtml)"/>
```

El resultat:

[Home](#) > [Enllaç](#) > [Vista 1](#)

Fil d'Ariadna

NOTA IMPORTANT:

Per fer-ne ús de les notacions emprades a l'exemple cal tenir en compte que aquestes segueixen la nomenclatura per a l'**Expression Language 2.2** (EL 2.2) que incorporen les darreres versions dels Servidors d'aplicacions (Tomcat 7, Weblogic 12c, ...).

Ús de l'Ariadna des de la capa de negoci

És pot fer ús del Bean de l'Ariadna als propis components/Beans/Serveis definits per a l'aplicació si aquest s'injecta mitjançant les següents anotacions:

```
@Autowired
@ManagedProperty(value="#{breadCrumbBean}")
private BreadCrumbBean breadCrumbBean;
```

A continuació es proporciona un exemple complet de Bean + vistes que e fan servir on es troben incloses totes les operacions disponibles que es poden realitzar amb l'Ariadna:

Fil d'Ariadna

TestBean.java

```
import javax.faces.application.FacesMessage;
import javax.faces.bean.ManagedProperty;
import javax.faces.context.FacesContext;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.context.annotation.Scope;
import org.springframework.stereotype.Component;
import cat.gencat.ctti.canigo.arch.web.jsf.BreadCrumbBean;
import cat.gencat.ctti.canigo.arch.web.jsf.BreadCrumbObject;

@Component("testBean")
@Scope("request")
public class TestBean {

    @Autowired
    @ManagedProperty(value="#{breadCrumbBean}")
    private BreadCrumbBean breadCrumbBean;

    private Integer level;
    private String key;
    private String label;
    private String url;

    public Integer getLevel() {return level; }
    public void setLevel(Integer level) {this.level = level; }
    public String getKey() {return key; }
    public void setKey(String key) {this.key = key; }
    public String getLabel() {return label; }
    public void setLabel(String label) {this.label = label; }
    public String getUrl() {return url; }
    public void setUrl(String url) {this.url = url; }

    public void testAddSimpleNavigation(){
        breadCrumbBean.addBreadCrumbLevel(key, label, url);
    }

    public void testAddNavigationAtLevel(){
        breadCrumbBean.addBreadCrumbLevel(level, key, label, url);
    }

    public void testSetNavigationAtLevel(){
        breadCrumbBean.setBreadCrumbLevel(level, key, label, url);
    }

    public void testRemoveNavigation(){
        breadCrumbBean.removeBreadCrumbLevel();
    }

    public void testRemoveNavigationAtLevel(){
        breadCrumbBean.removeBreadCrumbLevel(level);
    }

    public void testSkewNavigationAtLevel(){
        breadCrumbBean.keepNFirstsBreadCrumbLevels(level);
    }

    public void testGetLevelInfo(){
        BreadCrumbObject info=breadCrumbBean.getBreadCrumbLevel(level);
        FacesContext.getCurrentInstance().addMessage("breadCrumbForm",
            new FacesMessage(FacesMessage.SEVERITY_INFO,"Label: "+info.getLabel()+", URL: "
            + info.getUrl(), null));
    }
}
```

Fil d'Ariadna

/views/formulari.xhtml

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="yes" ?>
<ui:composition xmlns="http://www.w3.org/1999/xhtml"
  xmlns:ui="http://java.sun.com/jsf/facelets"
  xmlns:h="http://java.sun.com/jsf/html" >
  <h:form id="breadCrumbForm">
    <h:panelGrid columns="2">
      <h:outputText value="Level:" />
      <h:inputText id="level" value="#{testBean.level}" />

      <h:outputText value="Key:" />
      <h:inputText id="key" value="#{testBean.key}" />

      <h:outputText value="Label:" />
      <h:inputText id="label" value="#{testBean.label}" />

      <h:outputText value="URL:" />
      <h:inputText id="url" value="#{testBean.url}" />
    </h:panelGrid>

    <h:panelGrid columns="2">
      <h:commandButton value="#{msg['testAddSimpleNavigation.btn']}"
        action="#{testBean.testAddSimpleNavigation}" />
      <h:outputText value="#{msg['testAddSimpleNavigation.description']}" />

      <h:commandButton value="#{msg['testAddNavigationAtLevel.btn']}"
        action="#{testBean.testAddNavigationAtLevel}" />
      <h:outputText value="#{msg['testAddNavigationAtLevel.description']}" />

      <h:commandButton value="#{msg['testSetNavigationAtLevel.btn']}"
        action="#{testBean.testSetNavigationAtLevel}" />
      <h:outputText value="#{msg['testSetNavigationAtLevel.description']}" />

      <h:commandButton value="#{msg['testRemoveNavigation.btn']}"
        action="#{testBean.testRemoveNavigation}" />
      <h:outputText value="#{msg['testRemoveNavigation.description']}" />

      <h:commandButton value="#{msg['testRemoveNavigationAtLevel.btn']}"
        action="#{testBean.testRemoveNavigationAtLevel}" />
      <h:outputText value="#{msg['testRemoveNavigationAtLevel.description']}" />

      <h:commandButton value="#{msg['testSkewNavigationAtLevel.btn']}"
        action="#{testBean.testSkewNavigationAtLevel}" />
      <h:outputText value="#{msg['testSkewNavigationAtLevel.description']}" />

      <h:commandButton value="#{msg['testGetLevelInfo.btn']}"
        action="#{testBean.testGetLevelInfo}" />
      <h:outputText value="#{msg['testGetLevelInfo.description']}" />
    </h:panelGrid>
    <h:message for="breadCrumbForm" infoStyle="color: green;"
      errorStyle="color: red;" />
  </h:form>
</ui:composition>
```


Fil d'Ariadna

/views/vista1.xhtml

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="yes" ?>
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:ui="http://java.sun.com/jsf/facelets"
      xmlns:h="http://java.sun.com/jsf/html"
      xmlns:rich="http://richfaces.org/rich" >

  <ui:composition template="/views/layouts/template.jsf">
    <ui:define name="body">
      <h2>Vista 1</h2>
      <br />
      <h:form>
        <h:commandButton value="#{msg['addPage.btn']}"
          action="#{breadCrumbBean.addBreadCrumbLevel(null, 'Vista 1', '/views/vista1.xhtml)}/>
      </h:form>
      <ui:include src="/views/formulari.xhtml" />
    </ui:define>
  </ui:composition>
</html>
```

src/main/resources/config/i18n/applicationResources_ca_ES.properties (+ la resta)

```
...
#Proves Ariadna
addPage.btn=Afegeix aquesta pàgina a l'ariadna
testAddSimpleNavigation.btn=Afegir navegació a cua
testAddSimpleNavigation.description=* Informar camps Key/Label i URL
testAddNavigationAtLevel.btn=Afegir nivell navegació a posició
testAddNavigationAtLevel.description=* Informar camps Level, Key/Label i URL
testSetNavigationAtLevel.btn=Sobreescriu nivell navegació a posició
testSetNavigationAtLevel.description=* Informar camps Level, Key/Label i URL
testRemoveNavigation.btn=Esborra la darrera navegació
testRemoveNavigation.description=* No cal informar cap camp
testRemoveNavigationAtLevel.btn=Esborra un nivell de navegació
testRemoveNavigationAtLevel.description=* Informar camp Level
testSkewNavigationAtLevel.btn=Manté els N primers nivells de navegació
testSkewNavigationAtLevel.description=* Informar camp Level
testGetLevelInfo.btn=Obté informació nivell navegació
testGetLevelInfo.description=* Informar camp Level
...
```

La vista renderitzada de l'exemple ha de quedar com la següent imatge:

[Home](#)

Vista 1

Afegeix aquesta pàgina a l'ariadna	
Level:	<input type="text"/>
Key:	<input type="text"/>
Label:	<input type="text"/>
URL:	<input type="text"/>

Afegir navegació a cua	* Informar camps Key/Label i URL
Afegir nivell navegació a posició	* Informar camps Level, Key/Label i URL
Sobreescriu nivell navegació a posició	* Informar camps Level, Key/Label i URL
Esborra la darrera navegació	* No cal informar cap camp
Esborra un nivell de navegació	* Informar camp Level
Manté els N primers nivells de navegació	* Informar camp Level
Obté informació nivell navegació	* Informar camp Level