

## FAM 4.0 - Navegacio

### Taula de continguts

- 1. Què és PrettyFaces?
- 2. Configuració
- 3. Navegació
  - 3.1. Rutes, verbs HTTP i accions
  - 3.2. Definició de les rutes
  - 3.3. Com funciona?
    - 3.3.1. Entrada
    - 3.3.2. Sortida
- 4. Flux de navegació
  - 4.1. Exemples d'enllaços
    - 4.1.1. Anar a l'índex de les entitats des del menú principal
    - 4.1.2. Vista de llista o índex: edició de l'entitat
    - 4.1.3. Vista de detall
      - 4.1.3.1. Actualitza o crea l'entitat
      - 4.1.3.2. Torna a l'índex
      - 4.1.3.3. Suprimeix l'entitat amb confirmació

### 1. Què és [PrettyFaces](#) [1]?

PrettyFaces és una llibreria open-source per re-escriure les URL (URL-Rewriting) i fer-les més amigables amb suport millorat per JSF 1.1, 1.2 i 2.0. PrettyFaces resol el problema "URL REST" de forma elegant, i a més incorpora característiques com: accions de càrrega de pàgina, integració perfecta amb la navegació JSF i compatibilitat sense configuració amb d'altres frameworks web.

### 2. Configuració

El primer pas és afegir les dependències al projecte Maven. L'arquetipo del FAM ja les incorpora al pom.xml

```
<!-- Pretty URL -->
<dependency>
  <groupId>org.ocpsoft.rewrite</groupId>
  <artifactId>rewrite-servlet</artifactId>
  <version>2.0.12.Final</version>
</dependency>
<dependency>
  <groupId>org.ocpsoft.rewrite</groupId>
  <artifactId>rewrite-config-prettyfaces</artifactId>
  <version>${prettyfaces.version}</version>
</dependency>
<!-- End PrettyFaces -->
```

A continuació, a l'arxiu web.xml, cal indicar els fitxers que defineixen les rutes de la nostra aplicació sota el paràmetre com.ocpsoft.pretty.CONFIG\_FILES. Aquí tenim l'exemple del FAM:

```
<!-- Configuració PrettyFaces -->
<context-param>
  <param-name>com.ocpsoft.pretty.CONFIG_FILES</param-name>
  <param-value>/WEB-INF/pretty-aplicacio.xml,/WEB-INF/pretty-index.xml,/WEB-
INF/pretty-demos.xml</param-value>
</context-param>
<context-param>
  <param-name>com.ocpsoft.pretty.DEVELOPMENT</param-name>
  <param-value>true</param-value>
</context-param>
```

### 3. Navegació

Per mostrar com es configura la navegació farem servir com a exemple els següents valors:

- El model/entitat és EntitatBase
- Les pàgines JSF es troben al directori WebRoot/views
- El domini és aplicacions.diba.cat
- El context és fambo

#### 3.1. Rutes, verbs HTTP i accions

A continuació la taula que defineix la relació entre una petició HTTP i l'acció que es desencadena en el cas d'operacions amb les entitats i la base de dades, bàsicament lo que s'anomena CRUD: Create (crea), Read (llegeix), Update (actualitza) i Destroy (suprimeix).

Verb HTTP	Ruta	BackingBean::mètode	Usat per
GET	/entitatsBase	entitatBaseBB::index	Retorna la vista on apareixen totes les entitats base
GET	/entitatsBase/crea	entitatBaseBB::create	Retorna la vista per crear una nova entitat base
POST	/entitatsBase/crea	entitatBaseBB::saveOrUpdate	Crea (desa) una nova entitat base
GET	/entitatsBase/:id/edita	entitatBaseBB::edit	Retorna la vista per editar una entitat base que ja existeix
POST	/entitatsBase/:id/edita	entitatBaseBB::saveOrUpdate	Actualitza (desa) una entitat base concreta
POST	/entitatsBase/:id/edita	entitatBaseBB::delete	Suprimeix una entitat base concreta

En JSF, és el BackingBean i el mètode qui determina l'acció a realitzar. Per aquest motiu, una mateixa ruta i verb poden tenir diferents resultats segons el mètode a executar.

#### 3.2. Definició de les rutes

Una de les avantatges de PrettyFaces és el fet de poder definir una URL que executa un mètode d'un Backing Bean. Per exemple, en determinades etiquetes JSF no està disponible l'atribut action o actionListener. O en d'altres casos, la URL és una etiqueta HTML que no permet afegir cap referència a Java, però amb la definició de la navegació amb PrettyFaces si s'executarà el mètode desitjat.

La següent taula mostra les accions de navegació definides per FAMEntitatBase, la adreça URL corresponent i la configuració de PrettyFaces:

Acció de navegació	Ruta (URL)	PrettyFaces
--------------------	------------	-------------

Índex de les entitats	/entitatsBase	<pre>&lt;url-mapping id="entitatsBase"&gt; &lt;pattern value="/entitatsBase" /&gt; &lt;view-id value="/views/entitatBase/entitatBaseLlista.xhtml" /&gt;&lt;/url-mapping&gt;</pre>
Inicialitza la nova entitat base i retorna la vista HTML	/entitatsBase/crea	<pre>&lt;url-mapping id="newEntitatBase"&gt; &lt;pattern value="/entitatsBase/crea" /&gt; &lt;view-id value="/views/entitatBase/entitatBaseDetall.xhtml" /&gt; &lt;!-- Evita executar l'acció amb submit o Ajax, només amb GET --&gt; &lt;action onPostback="false"&gt;#{entitatBaseBB.crea} &lt;/action&gt;&lt;/url-mapping&gt;</pre>
Inicialitza la edició d'una entitat base i retorna la vista HTML	/entitatsBase/{id}/edita	<pre>&lt;url-mapping id="viewEntitatBase"&gt; &lt;pattern value="/entitatsBase/#{id}/edita" /&gt; &lt;view-id value="/views/entitatBase/entitatBaseDetall.xhtml" /&gt; &lt;!-- Evita executar l'acció amb submit o Ajax, només amb GET --&gt; &lt;action onPostback="false"&gt;#{entitatBaseBB.edit} &lt;/action&gt;&lt;/url-mapping&gt;</pre>
Inicialitza la duplicació d'una entitat base i retorna la vista HTML	/entitatsBase/{id}/duplica	<pre>&lt;url-mapping id="duplicaEntitatBase"&gt; &lt;pattern value="/entitatsBase/#{id}/duplica" /&gt; &lt;view-id value="/views/entitatBase/entitatBaseDetall.xhtml" /&gt; &lt;!-- Evita executar l'acció amb submit o Ajax, només amb GET --&gt; &lt;action onPostback="false"&gt;#{entitatBaseBB.duplica} &lt;/action&gt;&lt;/url-mapping&gt;</pre>
Mostra un diàleg.	/entitatsBase/dialog	<pre>&lt;url-mapping id="dialogEntitatBase"&gt; &lt;pattern value="/entitatsBase/dialog" /&gt; &lt;view-id value="/views/entitatBase/entitatBaseDialog.xhtml" /&gt;&lt;/url-mapping&gt;</pre>
Útil per seleccionar una entitat base des d'una altra pantalla		

### 3.3. Com funciona?

#### 3.3.1. Entrada

Quan arriba una petició, el servlet de PrettyFaces verifica si la URL coincideix amb algun dels patrons configurats. Si és així, llavors primer executa l'acció, si en té d'alguna definida, i a continuació retorna la vista corresponent.

#### 3.3.2. Sortida

Per generar la URL correcta a les vistes (arxius xhtml) i els Backing Bean, només cal escriure "pretty:<view\_id>"

Exemples:

- Enllaç a la pàgina índex en HTML

```
<h:link outcome="pretty:entitatsBase" value="Índex" />
```

- Enllaç a la pàgina índex des d'un bean:

```
public String index() {  
    return "pretty:entitatsBase";  
}
```

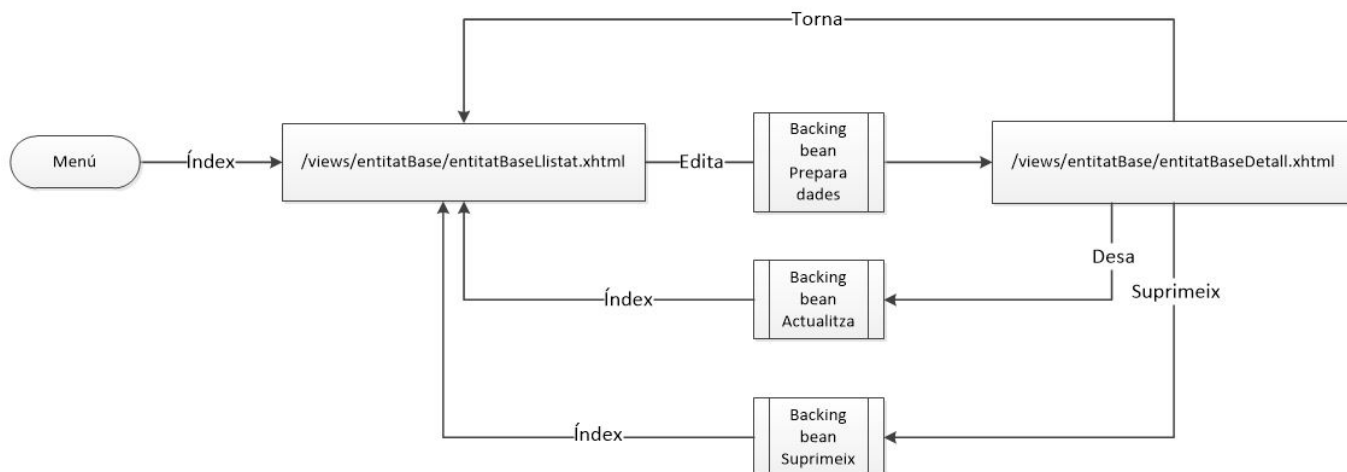
- Enllaç al detall des de l'index (fila). **El nom del paràmetre (id) coincideix amb el nom que s'ha definit a la navegació** `<pattern value="/entitatsBase/#{id}/edita" />`

```
<p:column headerText="Assumppte" sortBy="#{element.famEntAssumppte}">  
    <h:link outcome="pretty:viewEntitatBase" value="#{element.famEntAssumppte}">  
        <f:param name="id" value="#{element.id}"></f:param>  
    </h:link>  
</p:column>
```

Cal tenir en compte que si no troba cap regla de navegació que coincideixi amb la ruta, ja sigui definida amb PrettyFaces o amb JSF, llavors torna a mostrar la mateixa vista.

## 4. Flux de navegació

El flux normal de navegació és el que mostra la següent imatge:



### 4.1. Exemples d'enllaços

#### 4.1.1. Anar a l'Índex de les entitats des del menú principal

El menú principal està definit a l'arxiu menuAplicacio.xhtml i caldria afegir un enllaç com aquest:

```
<h:link outcome="pretty:entitatsBase">#{literalsAplicacio['Menu.EntitatBase']}</h:link>
```

#### 4.1.2. Vista de llista o índex: edició de l'entitat

```
<p:column headerText="Assumptes">
  <h:link outcome="pretty:viewEntitatBase" value="#{element.famEntAssumptes}">
    <f:param name="id" value="#{element.id}"></f:param>
  </h:link>
</p:column>
```

#### 4.1.3. Vista de detall

##### 4.1.3.1. Actualitza o crea l'entitat

```
<p:commandLink action="#{entitatBaseBB.saveOrUpdate}" id="save" ajax="true"
  styleClass="btn btn-default" type="submit" update="@form messages">
  <i class="fa fa-save" aria-hidden="true" />
  #{literalsCore['Helper.Save']}
</p:commandLink>
```

##### 4.1.3.2. Torna a l'índex

```
<h:link styleClass="btn btn-default" outcome="pretty:entitatsBase">
  <i class="fa fa-undo" aria-hidden="true" />
  #{literalsCore['Helper.Back']}
</h:link>
```

##### 4.1.3.3. Suprimeix l'entitat amb confirmació

```
<p:commandLink action="#{entitatBaseBB.delete}" styleClass="btn btn-
danger" immediate="true">
  <i class="fa fa-trash" aria-hidden="true" />
  #{literalsCore['Helper.Destroy']}
  <p:confirm header="Confirmació" message="#{literalsCore['Helper.Destroy.Confirm']}"
  />
</p:commandLink>
```

El diàleg de confirmació està definit com

```
<p:confirmDialog global="true" showEffect="fade" hideEffect="fade">
  <p:commandButton value="#{literalsCore['Helper.Yes']}" type="button"
    styleClass="ui-confirmdialog-yes btn btn-default" icon="fa fa-
check" />
  <p:commandButton value="#{literalsCore['Helper.No']}" type="button"
    styleClass="ui-confirmdialog-no btn btn-default" icon="fa fa-
close" />
</p:confirmDialog>
```

URL d'origen: <https://comunitatdstsc.diba.cat/wiki/fam-30-navegacio>

#### Enllaços:

[1] <http://www.ocpssoft.org/prettyfaces/>