FAM 4.0 - Creació d'un manteniment simple

Taula de continguts

1. Prerequisits

- Instal·lada la darrera versió de l'entorn de desenvolupament de la Diputació.
- Un projecte basat en l'arquetip del FAM Backoffice

2. Creació del manteniment

En aquest article s'explica com generar un CRUD (Create Update Read Delete) agafant com a base una entitat de les més simples del projecte FAM Backoffice: FamZona.

> Nota: Als noms de les entitats es respecta el prefix FAM de les taules, però a la resta de classes no per facilitar la lectura i la cerca.

2.1. Generació de l'entitat

Per generar l'entitat a partir de la taula cal seguir l'article Generació model de dades JPA [1]

2.2. Generació de les classes base: servei, BB, data model, cerca i vistes. Scaffolding

Aquí s'expliquen els passos bàsics, per veure en detalla com funciona llegir l'article scaffolding [2].

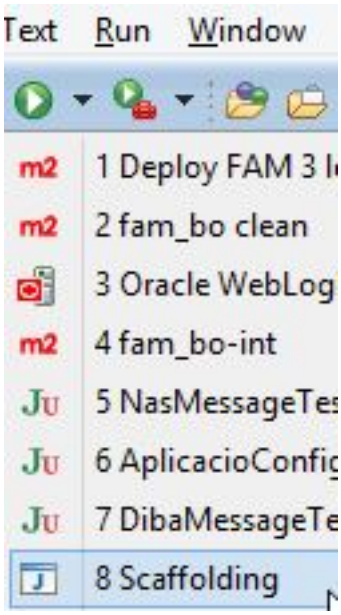
Els canvis es fan a l'arxiu /scaffolding.json i cal modificar la clau entities amb el nom de l'entitat ["FamZona"] i la clau base_package amb el nom del paquet del projecte. L'arixu quedaria així:

```
{
  "output_directory": ".",
  "template_location": "scaffolding",
```

```
    "base_package": "cat.diba.jee.fam",
    "read": false,
    "only_with_entity_directives": true,
    "entities": ["Zona"]
  }
```

> Nota: com es vol obtenir ZonaService, el nom de l'entitat s'informa sense el prefix Fam. A les plantilles, per solucionar possibles problems de si porta o no prefix, ja està contemplat en cadascun dels casos com es veurà als apartats de revisió del codi generat.

Executar la classe tools.Scaffolding que està definida al menú per obtenir les classes generades a partir de les plantilles.



En aquest punt ja estan creades les classes i les vistes mínimes per implementar el CRUD de l'entitat FamZona. Al següent apartat s'expliquen els canvis concrets de cadascuna de les classes.

3. Modificacions a realitzar

3.1. Servei: ZonaService

Les classes de servei són les que inclouen el negoci de l'aplicació i no tenen accés a la vista. Com que les aplicacions fan servir JPA i un contenidor de transaccions del servidor d'aplicacions, no cal una capa de DAO.

El servei generat en base a la plantilla és el següent:

```
package cat.diba.jee.fam.service;

import javax.ejb.TransactionManagement;
import javax.ejb.TransactionManagementType;

import com.querydsl.core.types.EntityPath;

import cat.diba.jee.core.service.DibaService;
import cat.diba.jee.fam.entity.FamZona;
import cat.diba.jee.fam.entity.QFamZona;
```

```
@TransactionManagement(TransactionManagementType.BEAN)
public class ZonaService extends DibaService<FamZona> {

    /**
     * Constructor per defecte
     *
     */
    public ZonaService() {
        super(FamZona.class);
    }

    @Override
    public EntityPath<FamZona> entityPath() {
        return QFamZona.famZona;
    }
}
```

Es pot observa l'ús de la classe generada per QueryDSL: QFamZona. En aquest cas no cal modificar ni afegir res al servei.

3.2. Cerca: ZonaSearchBean

Aquesta classse s'utilitza a la vista ja que proporciona els camps per filtrar o cercar resultats i alhora utilitza el servei per poder aplicar les condicions de cerca als resultats.

La classe generada en base a la plantilla és la següent:

```
package cat.diba.jee.fam.view.search;

import javax.faces.bean.ManagedBean;
import javax.faces.bean.SessionScoped;

import org.primefaces.model.SortOrder;

import com.querydsl.core.BooleanBuilder;
import com.querydsl.core.types.OrderSpecifier;
import com.querydsl.core.types.Predicate;

import cat.diba.jee.core.view.search.DibaSearchBean;
import cat.diba.jee.fam.entity.QFamZona;

@ManagedBean
@SessionScoped
public class ZonaSearchBean extends DibaSearchBean {

    /**
     *
     */
    private static final long serialVersionUID = 1L;

    /*
     * (non-Javadoc)
     *
     * @see cat.diba.jee.core.view.search.DibaSearchBean#orderBy()
     */
    @Override
```

```
    public OrderSpecifier<?> orderBy() {
        // FIXME
        // return QFamZona.zona..asc();
        return null;
    }

    /*
     * (non-Javadoc)
     *
     * @see cat.diba.jee.core.view.search.DibaSearchBean#order(java.lang.String,
     * org.primefaces.model.SortOrder)
     */
    @Override
    public OrderSpecifier<?> orderBy(String field, SortOrder sortOrder) {
        // FIXME
        return orderBy();
    }

    /*
     * (non-Javadoc)
     *
     * @see cat.diba.jee.core.view.search.DibaSearchBean#reset()
     */
    public void reset() {
        // FIXME
    }

    /*
     * (non-Javadoc)
     *
     * @see cat.diba.jee.core.view.search.DibaSearchBean#where()
     */
    public Predicate where() {
        QFamZona zona = QFamZona.famZona;
        BooleanBuilder where = new BooleanBuilder();

        // FIXME
        return where;
    }
}
```

A tenir en compte que l'abast del bean és @SessionScoped per poder mantenir els criteris de cerca i l'índex de la pàgina al navegar per l'aplicació. Si això no és necessari, llavors por canviar-se per @ViewScoped que és més eficient des del punt de vista de memòria al servidor.

La cerca de zona és només pel nom, llavors caldrà afegir a la classe un atribut que permeti introduir aquest valor a la vista:

```
    private String name;

    public String getName() {
        return name;
    }

    @Override
    public void reset() {
        this.name = null;
    }
```

```
    public void setName(String name) {
        this.name = name;
    }
```

I adaptar les condicions de cerca amb l'ajuda de la classse QFamZona generada per QueryDSL:

```
    public Predicate where() {
        QFamZona famZona = QFamZona.famZona;
        BooleanBuilder where = new BooleanBuilder();

        if (null != name) {
            where = where.and(famZona.famZonaNom.containsIgnoreCase(name));
        }
        return where;
    }
```

I per últim, es modifican les condicions d'ordenació. Més endavant s'explica com es lliga amb la vista i la taula:

```
    /**
     * Ordenació per defecte
     */
    @Override
    public OrderSpecifier<?> orderBy() {
        return QFamZona.famZona.famZonaNom.asc();
    }

    /**
     * Ordenació segons les columnes de la taula de la vista
     */
    @Override
    public OrderSpecifier<?> orderBy(String sortField, SortOrder sortOrder) {
        OrderSpecifier<?> orderBy = orderBy();
        if (sortField != null && sortOrder != null) {
            if (
sortField.endsWith(QFamZona.famZona.famZonaNom.getMetadata().getName())) {
                orderBy = SortOrder.DESCENDING.equals(sortOrder)
                        ? QFamZona.famZona.famZonaNom.desc()
                        : QFamZona.famZona.famZonaNom.asc();
            }
        }
        return orderBy;
    }
```

3.3. Model de dades: ZonaDataModel

Aquesta classe està relacionada amb el model de dades necessari per a les taules de PrimeFaces [3] amb càrrega lazy dels elements.

El model de dades generat en base a la plantilla és el següent:

```
package cat.diba.jee.fam.view.datamodel;

import javax.inject.Inject;
```

```
import cat.diba.jee.core.service.DibaService;
import cat.diba.jee.core.view.datamodel.DibaDataModel;
import cat.diba.jee.fam.entity.FamZona;
import cat.diba.jee.fam.service.ZonaService;

public class ZonaDataModel extends DibaDataModel<FamZona> {

    @Inject
    private ZonaService zonaService;


    @Override
    public DibaService<FamZona> service() {
        return zonaService;
    }
}
```

No cal fer cap modificació.

3.4. Backing Bean o Managed Bean: ZonaBB

Aquesta és la classe que gestiona totes les crides de la vista i retorna els valors a mostrar.

La classe generada en base a la plantilla és la següent:

```
package cat.diba.jee.fam.view.bb;

import javax.faces.bean.ManagedBean;
import javax.faces.bean.ManagedProperty;
import javax.faces.bean.ViewScoped;
import javax.inject.Inject;

import org.apache.commons.logging.Log;
import org.apache.commons.logging.LogFactory;

import cat.diba.jee.core.service.DibaService;
import cat.diba.jee.core.view.bb.DibaBB;
import cat.diba.jee.core.view.search.DibaSearchBean;
import cat.diba.jee.fam.entity.FamZona;
import cat.diba.jee.fam.service.ZonaService;
import cat.diba.jee.fam.view.datamodel.ZonaDataModel;
import cat.diba.jee.fam.view.search.ZonaSearchBean;

@ManagedBean
@ViewScoped
public class ZonaBB extends DibaBB<FamZona> {

    private static final long serialVersionUID = 1L;
    private static final String CLASS_ID = ZonaBB.class.getName();
    private static final Log LOG = LogFactory.getLog(CLASS_ID);

    @Inject
    private ZonaService zonaService;
    @Inject
    private ZonaDataModel dataModel;
    @ManagedProperty(value = "#{zonaSearchBean}")
```

```
    private ZonaSearchBean zonaSearchBean;

    @Override
    public ZonaDataModel getDataModel() {
        return dataModel;
    }

    @Override
    public String index() {
        LOG.trace(CLASS_ID + "::index()");
        return "pretty:zonas";
    }

    @Override
    public DibaService<FamZona> service() {
        return zonaService;
    }

    @Override
    protected FamZona newEntity() {
        LOG.trace(CLASS_ID + "::newEntity()");
        return new FamZona();
    }

    /**
     * Instància el bean de cerca, que és de sessió, per poder mantenir els
     * valors dels camps i la darrera pàgina de la taula al tornar d'editar una
     * entitat.
     *
     * @see DibaSearchBean#setLastFirst(int)
     * @see DibaSearchBean#getLastFirst()
     *
     */
    public void setZonaSearchBean(ZonaSearchBean searchBean) {
        this.zonaSearchBean = searchBean;
        this.dataModel.setSearchBean(searchBean);
    }
}
```

Aspectes a considerar:

- El bean de cerca s'injecta com a una propietat de forma que pot compartir-se entre diversos beans: @ManagedProperty(value = "#{zonaSearchBean}")
- L'abast és @ViewScoped per una millor gestió de la memòria al servidor
- Quan s'injecta el bean de cerca, automàticament s'instància al model de dades. Veure: public void setZonaSearchBean(ZonaSearchBean searchBean)
- La vista de 'índex no és el plural correcte de Zona i es modificarà per coherència a "pretty:zones"

Segons les necessitats del negoci, cal ampliar ZonaBB per poder tornar totes les zones ordenades pel nom i mostrar-les en un camp de selecció desplegable. A la vista veurem com es tracta la llista d'entitats sense necessitat de transformar-la.

```
    /**
     * Zones amb l'ordenació per defecte (el nom)
     *
     * @return
     */
```

```
    public List<FamZona> getZones() {
        List<FamZona> zones = new ArrayList<FamZona>();
        try {
            zones = zonaService.findAll(null, zonaSearchBean.orderBy());
        } catch (IllegalStateException | SecurityException | SystemException | Rollba
ckException | HeuristicRollbackException e) {
            MessagesUtils.errorMessage(new DibaException(e));
        }

        return zones;
    }
```

I aquestes són les modificacions a fer en les classes generades. A continuació es veure com definir la navegació amb PrettyFaces y les vistes.

4. Navegació: PrettyFaces

Les regles de navegació que cal agefir a l'arxiu pretty-aplicacio.xml són

```
    <!-- Índex o llista: des del menu o des de l'edició -->
    <url-mapping id="zones">
        <pattern value="/zones" />
        <view-id value="/views/zona/zonaLlista.xhtml" />
    </url-mapping>

    <!-- Edició de la zona des de la fila de la taula -->
    <url-mapping id="viewZona">
        <pattern value="/zones/#{id}/edita" />
        <view-id value="/views/zona/zonaDetall.xhtml" />
        <!-- Evita executar l'acció amb submit o Ajax, només amb GET -->
        <action onPostback="false">#{zonaBB.edit}</action>
    </url-mapping>

    <!-- Nova zona des de l'índex o llistat -->
    <url-mapping id="newZona">
        <pattern value="/zones/crea" />
        <view-id value="/views/zona/zonaDetall.xhtml" />
        <!-- Evita executar l'acció amb submit o Ajax, només amb GET -->
    <action onPostback="false">#{zonaBB.create}</action>
    </url-mapping>
```

A l'article navegació [4]es pot trobar una explicació detallada.

5. Vistes

Ara les vistes tenen una plantilla base, /WebRoot/WEB-INF/templates/layout.xhtml, i els arxius xhtml concrets de les entitats només descriuen la vista concreta.

Com a pas previ, cal afegir a l'arxiu on es defineix el menú, /WebRoot/WEB-INF/templates/menuAplicacio.xhtml, l'opció de la llista de zones:

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

El scaffolding també genera dos plantilles per les vistes més habituals de l'aplicació dins d'un directori amb el nom de l'entitat:

- /WebRoot/views/zona/zonaDetall.xhtml
- /WebRoot/views/zona/zonaLlista.xhtml

Cal tenir en compte que el layout està basat en Bootstrap 3 [5].

5.1. Llista

El codi generat per la vista és el següent:

```
<ui:composition
  xmlns="http://www.w3.org/1999/xhtml"
  xmlns:h="http://java.sun.com/jsf/html"
  xmlns:f="http://java.sun.com/jsf/core"
  xmlns:ui="http://java.sun.com/jsf/facelets"
  xmlns:p="http://primefaces.org/ui"
  xmlns:o="http://omnifaces.org/ui" xmlns:of="http://omnifaces.org/functions"
  xmlns:diba="http://java.sun.com/jsf/composite/diba/components"
  template="/WEB-INF/templates/layout.xhtml">


  <ui:define name="content">

    <div class="page-header c-diba">
      <h2>#{literalsAplicacio['Zona.IndexTitle']}
          <small>#{literalsCore['Helper.Index']}</small>
      </h1>
    </div>

    <h:form id="zona_llistat_form" styleClass="form-horizontal">
      <!-- http://stackoverflow.com/questions/4573190/jsf-primefaces-update-
attribu... [6]  -->
      <h:panelGroup id="zona_search_fields" layout="block" styleClass="well well-sm">

        <!-- FIXME: camps de cerca -->
        <div class="form-group">
          <!-- Exemple
          <o:outputLabel for="descripcio" styleClass="col-sm-1 control-
label" value="#{literalsAplicacio['Zona.Descripcio']}" />
          <div class="col-sm-3">
            <h:inputText styleClass="form-
control" id="descripcio" value="#{zonaSearchBean.descripcio}" />
          </div>
          -->
        </div>

        <diba:llistaCercaButtons
          search="#{zonaBB.dataModel.search}"
          reset="#{zonaBB.dataModel.reset}" />

      </h:panelGroup>

      <div class="table-responsive well well-sm">

        <p:dataTable id="zona_taula" paginatorPosition="top"
          var="element"
```

```
          tableStyleClass="table table-striped table-hover"
          tableStyle="width: 100%;"
          value="#{zonaBB.dataModel.lazyDataModel}"
          lazy="true"
          paginator="true"
          rows="20"
          paginatorTemplate="{RowsPerPageDropdown} {FirstPageLink} {PreviousPageLink}
 {PageLinks} {NextPageLink} {LastPageLink} {CurrentPageReport} {New}"
          currentPageReportTemplate="#{literalsCore['Helper.Table.PageTemplate']}"
          emptyMessage="#{literalsCore['Helper.Table.Empty']}"
          first="#{zonaBB.dataModel.first}">

          <f:facet name="{New}">
            <span class="pull-right">
              <h:link styleClass="btn btn-default" style="text-
align: right;" role="button" outcome="pretty:newZona" >
                <i class="fa fa-plus" aria-hidden="true"></i>
                #{literalsCore['Helper.New']}
              </h:link>
            </span>
          </f:facet>

          <!-- FIXME: columnes de la taula -->
          <!-- Exemple
          <p:column headerText="#{literalsAplicacio['Zona.Descripcio']}">
            <h:link outcome="pretty:viewZona" value="#{element.id}">
              <f:param name="id" value="#{element.id}"></f:param>
            </h:link>
          </p:column>
          -->

        </p:dataTable>

      </div>

    </h:form>

  </ui:define>

</ui:composition>
```

Aspectes a destacar:

- Els botons de la cerca fan servir un component propi de la Diputació que rep com a paràmetres els dos mètodes a cridar: cerca i neteja.

```
<diba:llistaCercaButtons
      search="#{zonaBB.dataModel.search}"
      reset="#{zonaBB.dataModel.reset}" />
```

- El model de dades de la taula es declara així:

```
value="#{zonaBB.dataModel.lazyDataModel}"
```

- El botó de nou element es declara com a un facet de la taula:

```
<f:facet name="{New}">
  <span class="pull-right">
```

```
    <h:link styleClass="btn btn-default" style="text-
align: right;" role="button" outcome="pretty:newZona" >
      <i class="fa fa-plus" aria-hidden="true"></i>
      #{literalsCore['Helper.New']}
    </h:link>
  </span>
</f:facet>
```

Les modificacions a fer són:

- Afegir un camp de cerca al primer FIXME:

```
<div class="form-group">
  <o:outputLabel for="nom" styleClass="col-sm-1 control-
label" value="#{literalsAplicacio['Zona.Nom']}" />
  <div class="col-sm-3">
    <h:inputText styleClass="form-control" id="nom" value="#{zonaSearchBean.name}" />
  </div>
</div>
```

- Afegir una columna a la taula amb ordenació i enllaç al detall de l'entitat:

```
Indicar a la taula la columna per defecte afegint la propietat següent a p:dataTable:
 sortBy="#{element.famZonaNom}"

<p:column headerText="#{liter
alsAplicacio['Zona.Nom']}" sortBy="#{element.famZonaNom}">
  <h:link outcome="pretty:viewZona" value="#{element.famZonaNom}">
    <f:param name="id" value="#{element.id}"></f:param>
  </h:link>
</p:column>
```

I aquest entitat no necessita més canvis a la llista.

5.2. Detall

El codi generat per la vista és el següent:

```
<ui:composition xmlns="http://www.w3.org/1999/xhtml"
  xmlns:h="http://java.sun.com/jsf/html"
  xmlns:f="http://java.sun.com/jsf/core"
  xmlns:ui="http://java.sun.com/jsf/facelets"
  xmlns:p="http://primefaces.org/ui"
  xmlns:o="http://omnifaces.org/ui"
  xmlns:of="http://omnifaces.org/functions"
  xmlns:diba="http://java.sun.com/jsf/composite/diba/components"
  template="/WEB-INF/templates/layout.xhtml">

  <ui:define name="content">
    <o:importConstants type="cat.diba.jee.fam.view.bb.ZonaBB" />

    <div class="page-header c-diba">
      <h2>#{literalsAplicacio['Zona.EditTitle']}
          <small>#{literalsCore['Helper.Edit']}</small>
      </h2>
    </div>
```

```
    <h:form styleClass="form-horizontal" id="zona_detail_form">
      <div class="form-edit">
        <!-- FIXME: camps del formulari -->
        <div class="form-group">
        </div>
      </div>

      <diba:detailButtons
        deleteAction="#{zonaBB.delete}"
        saveOrUpdateAction="#{zonaBB.saveOrUpdate}"
        cancelLink="pretty:zones"
        isNew="#{zonaBB.detailEntity.id eq null}" />

    </h:form>

    <o:highlight styleClass="has-error" focus="true" />

  </ui:define>

</ui:composition>
```

Aspectes a destacar:

- Els botons del detall estan definits com a un component de la Diputació. Només cal proporcionar els paràmetres corresponents:

```
<diba:detailButtons
   deleteAction="#{zonaBB.delete}"
   saveOrUpdateAction="#{zonaBB.saveOrUpdate}"
   cancelLink="pretty:zones"
   isNew="#{zonaBB.detailEntity.id eq null}" />
```

- Per facilitar la validació (veure l'article de validació [7] per més informació), cal afegir fora del form:

```
<o:highlight styleClass="has-error" focus="true" />
```

La plantilla generada no incorpora el contingut del formulari amb els camps, per tant si afegim els camps al formulari el resultat és:

```
<div class="form-edit">
  <div class="form-group">
    <o:outputLabel for="zonaNom" value="#{literalsAplicacio['Zona.Nom']}" styleClass=
"control-label col-sm-1" />
    <div class="col-sm-3">
      <h:inputText value="#{zonaBB.detailEntity.famZonaNom}" id="zonaNom" styleClass=
"form-control">
        <f:validateRequired />
      </h:inputText>
    </div>
  </div>
</div>
```

El punt més important és que l'atribut **for** de l'**outputLabel** coincideix amb l'**id** de l'**InputText** i llavors queda lligat el missatge d'error que és obligatori. La documentació original d'OmniFaces [8] és molt clara i amb exemples: <o:outputLabel> [9] <o:highlight> [10]

URL d'origen: https://comunitatdstsc.diba.cat/wiki/fam-30-creacio-dun-manteniment-simple

Enllaços:
[1] https://comunitatdstsc.diba.cat/wiki/fam-generacio-del-model-de-dades
[2] https://comunitatdstsc.diba.cat/wiki/fam-30-scaffolding
[3] http://www.primefaces.org/showcase/ui/data/datatable/basic.xhtml
[4] https://comunitatdstsc.diba.cat/wiki/fam-30-navegacio
[5] http://getbootstrap.com/
[6] http://stackoverflow.com/questions/4573190/jsf-primefaces-update-attribute-does-not-update-component#4574266
[7] https://comunitatdstsc.diba.cat/wiki/fam-30-validacio
[8] http://showcase.omnifaces.org/
[9] http://showcase.omnifaces.org/components/outputLabel
[10] http://showcase.omnifaces.org/components/highlight