

Développement d'une base de données et de son interface graphique structurée par Struts2



LOUISE OTT
LIFL – UNIVERSITÉ LILLE 1
2010/2011

Qui suis-je ?



- **2005 - 2010 :**

Master en bioinformatique à l'Université de Strasbourg

- **11/2010 – 11/2011 :**

Ingénieure d'étude pour la plateforme bioinformatique de Lille

- **11/2011 à aujourd'hui :**

Ingénieure d'étude CNRS dans l'équipe diabète et obésité à l'IBL

Base de données - Création

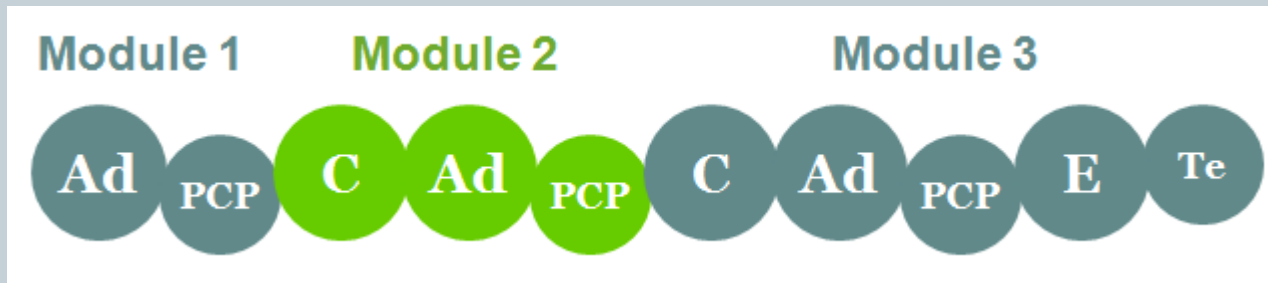


- **Besoin :**
Base de données biologique de Non Ribosomal Peptid Synthase (NRPS) pour le laboratoire PROBIOGEM.
- **Utilisation de POSTGRESQL**
- **NRPS** : Synthetase d'un peptide non ribosomal

Base de données - NRPS



- Organisée en modules.
- Chaque module est organisé en domaines :
 - **Ad** : Sélectionne l'AA
 - **PCP** : Fixe AA
 - **C** : Forme le lien peptidique entre AA
 - **Te** : Catalyse la libération du NRP



Base de données - Création



- **Besoin :**
Base de données biologique de Non Ribosomal Peptid Synthase (NRPS) pour le laboratoire PROBIOGEM.
- Utilisation de POSTGRES SQL
- **NRPS** : Synthetase d'un peptide non ribosomal
- Chaque objet biologique aura **son équivalence** dans une table de la base de données

program	
id_program	INTEGER
name_program	CHARACTER VARYING(500)
link	CHARACTER VARYING(1,000)

prediction	
id_prediction	INTEGER
id_program	INTEGER
id_protein	INTEGER
score	INTEGER

domain	
id_domain	INTEGER
id_prediction	INTEGER
code_descr	CHARACTER VARYING(50)
id_module	INTEGER
start_domain	INTEGER
end_domain	INTEGER
length_domain	INTEGER
order_on_prot	INTEGER

descr_domain	
code_descr	CHARACTER VARYING(50)
name_descr	CHARACTER VARYING(100)
function	CHARACTER VARYING(500)

module	
id_module	INTEGER
name_module	CHARACTER VARYING(100)
order_on_prot	INTEGER
incorporate_aa	CHARACTER VARYING(20)
monomer_norine	CHARACTER VARYING(50)

ref_syn	
id_ref	INTEGER
id_synthetase	INTEGER

protein	
id_protein	INTEGER
id_synthetase	INTEGER
name_seq	CHARACTER VARYING(300)
length	INTEGER
weight	INTEGER
seq	CHARACTER VARYING(30,000)
order_on_syn	INTEGER
seq_nuc	CHARACTER VARYING(90,000)
first_bound	INTEGER
last_bound	INTEGER
ref_id	CHARACTER VARYING(30)
complement	BOOLEAN

organism	
id_orga	INTEGER
taxid	INTEGER
name_orga	CHARACTER VARYING(100)
taxo	CHARACTER VARYING(1,000)
gram	CHARACTER VARYING(100)
syno	CHARACTER VARYING(1,000)

orga_syn	
id_orga	INTEGER
id_synthetase	INTEGER

synthetase	
id_synthetase	INTEGER
name	CHARACTER VARYING(500)
iscomplete	CHARACTER VARYING(20)

link_orga	
id_orga	INTEGER
acc	CHARACTER VARYING(32)

link_protein	
id_protein	INTEGER
acc	CHARACTER VARYING(32)

link	
acc	CHARACTER VARYING(32)
name_db	CHARACTER VARYING(100)

link_syn	
id_synthetase	INTEGER
acc	CHARACTER VARYING(32)

Base de données – L'interface graphique



- Choix du couple :
jsp (pages web) et JAVA (langage de traitement)
- Mise en place du pattern MVC (Modèle-Vue-Contrôleur):
 - STRUTS = Framework MVC pour le couple jsp/java
 - Il existe Struts 1 et Struts 2 : non compatibles
 - ⇒ Il faut en choisir un.

Strust 2 – Pourquoi ?



- Pas de couplage de code avec l'api servlet
- Capture d'un formulaire **automatique**
- **Conversion** automatique des types via OGNL
- Intercepteurs pour pré/post traitements sur la requête = **traitement des erreurs**
- **PAS de code java** dans les jsp

Strust 2 - Composition

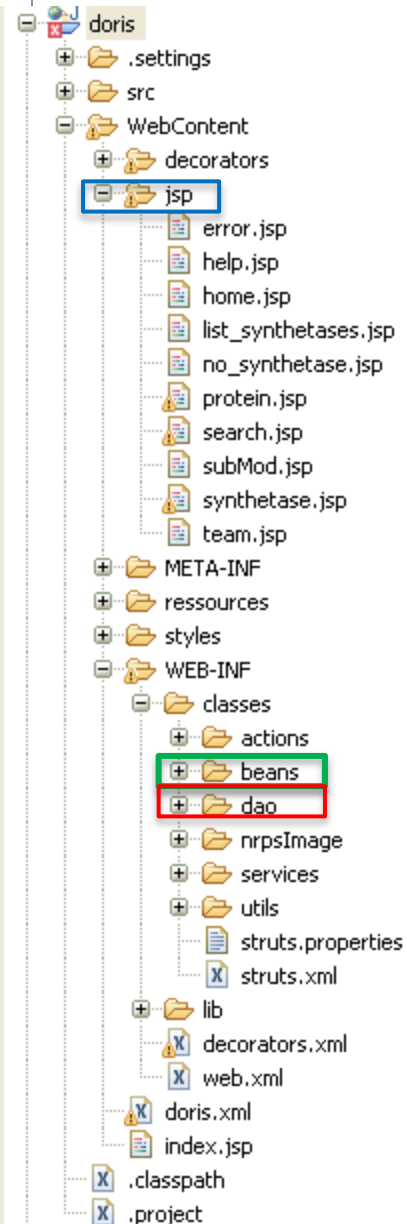


- **STRUTS 1:**
 - Projet Open Source développé par la communauté **Jakarta** d'Apache
 - Squelette d'application type MVC proposant des options pour un **développement rapide**
- **WebWork :**
 - Projet à l'initiative d'**OpenSymphony**, en mars 2002
 - Créé à partir du noyau Struts
 - Introduit de **nouveaux concepts** et fonctionnalités

Strust 2 - Structure de mon projet



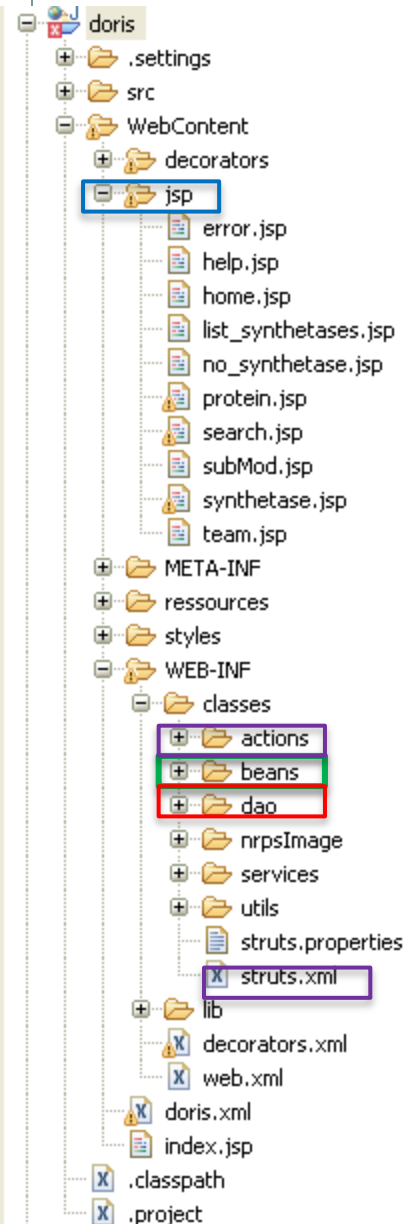
- **Modèle** : BEANS
- Accès au modèle : **DAO** = Data Access Object
- **Vue** : JSP



Strust 2 - Structure de mon projet



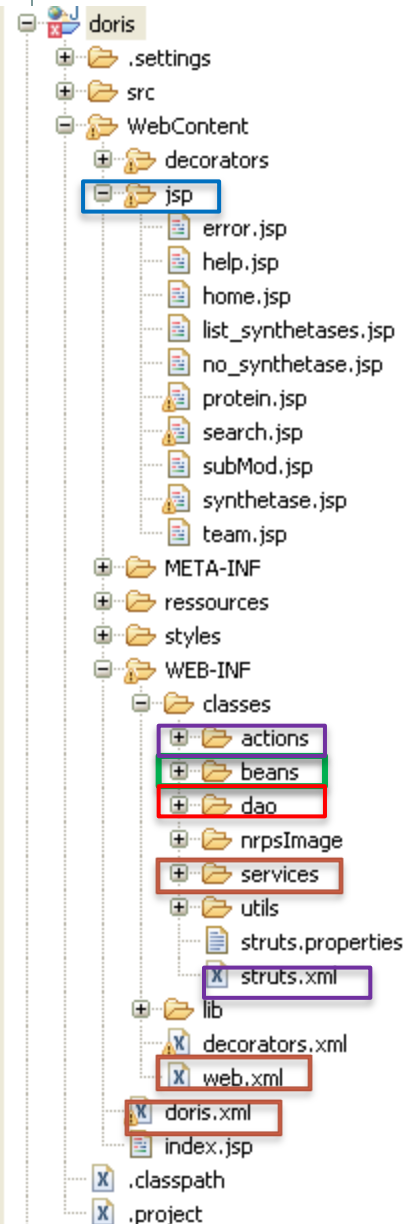
- **Modèle** : BEANS
- Accès au modèle : **DAO** = Data Access Object
- **Vue** : JSP
- **Contrôleur** :
 - `struts.xml` : fait le lien entre les actions et les jsp
 - Les actions : initialisent les beans via les DAO



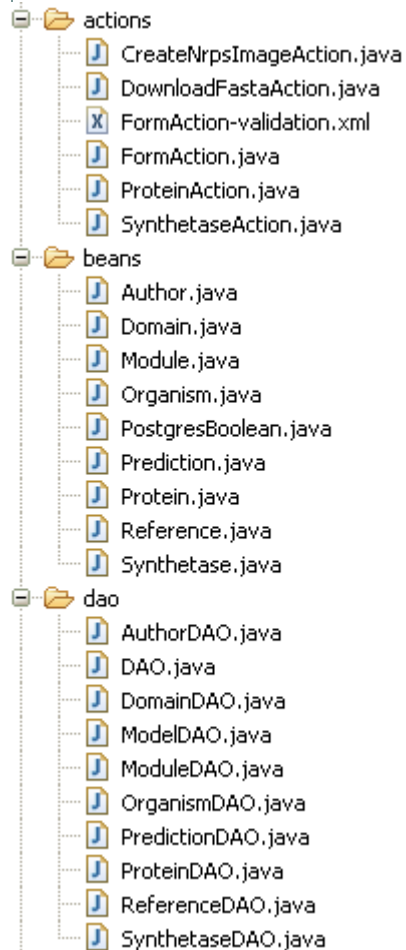
Strust 2 - Structure de mon projet



- **Modèle** : BEANS
- Accès au modèle : **DAO** = Data Access Object
- **Vue** : JSP
- **Contrôleur** :
 - struts.xml : fait le lien entre les actions et les jsp
 - Les actions : initialisent les beans via les DAO
- **services** : Datasource permettant accès à la base de données
- **web.xml** : déclaration de la DataSource
- **doris.xml** : Context, où trouver le projet et la base de données



Strust 2 - Structure de mon projet



- Dans mon cas :
 - Un bean pour chaque table de la base de données
 - Un DAO pour chaque bean :
 - Permet accès à la **base de données** et **remplissage des beans**, pour réutilisation de leur valeur dans l'**action**
 - Une action à chaque fois qu'il est nécessaire d'accéder à la base de données
 - Une validation pour les formulaires

La Vue - Affichage de search.jsp



Doris

[home](#) [general search](#) [help](#) [submit/modify](#)

Synthetase search

When several fields are selected, the results must match each of them.

Basic search

by Doris id (ex: DOR00015):

by name :

by status:

Protein

by name :

by RefSeq id :

by GI id :

by Uniprot id :

by length of the protein, ranging between : and :

protein containing special domain :

Bibliography reference search

by author :

by title :

by journal :

by year :

by pmid :

Organism Search

by taxon id :

by name :

by taxonomy (use ';' to separate the words) :

Lors du « submit » :

- Validation des champs
- Fonction de l'action cherche dans la BD les valeurs grâce aux DAO et aux objets des beans.
- Renvoie à la page [list_synthetase.jsp](#) si pas d'erreur

La Vue - search.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ taglib prefix="s" uri="/struts-tags"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/1d
<html><head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Doris::Search</title>
</head>

<body>

    <h1>Synthetase search</h1>
    <br />
    When several fields are selected, the results must match each of them.
    <br />

    <s:form name="searchForm" method="post" action="formAction"
        cssClass="formulaire" validate="true">

        <s:label value="Basic search" labelposition="top"
            cssClass="littleTitle"></s:label>

        <s:textfield name="idSyn" label="by Doris id (ex: DOR00015)" labelposition="left" />

        <s:textfield name="syn.name" id="synName" label="by name "
            labelposition="left" />

        <s:select name="syn.isComplete" headerKey="1"
            headerValue="-- Please Select --" list="statusList" label="by status"
            list="#{'complete':'complete','uncomplete':'uncomplete','unverified':'unverified'

        <s:label value="Protein" labelposition="top" cssClass="littleTitle"></s:label>

        <s:textfield name="prot.name" id="protName" label="by name "
            labelposition="left" />

        <s:textfield name="prot.refseq" id="refseqId" label="by RefSeq id "
            labelposition="left" />
        <s:textfield name="prot.gi" id="giId" label="by GI id "
            labelposition="left" />
        <s:textfield name="prot.uniprot" id="uniprotId" label="by Uniprot id "
            labelposition="left" />
```

- Les « name » sont liés aux variables de l'action et initialisent ces variables

- Exemple :

`syn.name` initialise la valeur de la variable `name` de l'objet du bean `Synthetase` déclaré dans l'action `FormAction`

La Vue – la validation

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE validators PUBLIC "-//OpenSymphony Group//XWork Validator 1.0.2//EN" "htt:

<validators>
  <field name="syn.id">
    <field-validator type="regex">
      <param name="trim">true</param>
      <param name="expression"><![CDATA[^[ 0-9DOR] *$]]></param>
      <message>The field doris id must be DOR followed by a number.</message>
    </field-validator>
  </field>

  <field name="syn.name">
    <field-validator type="regex">
      <param name="trim">true</param>
      <param name="expression"><![CDATA[^[ A-Za-z] *$]]></param>
      <message>The field synthetase name can't be a number.</message>
    </field-validator>
  </field>

  <field name="prot.name">
    <field-validator type="regex">
      <param name="trim">true</param>
      <param name="expression"><![CDATA[^[ A-Za-z] *$]]></param>
      <message>The field protein name can't be a number.</message>
    </field-validator>
  </field>
  <field name="prot.lengthLimit1">
    <field-validator type="regex">
      <param name="trim">true</param>
      <param name="expression"><![CDATA[^[ 0-9] *$]]></param>
      <message>The field length of the protein must be a number.</message>
    </field-validator>
  </field>
  <field name="prot.lengthLimit2">
    <field-validator type="regex">
      <param name="trim">true</param>
      <param name="expression"><![CDATA[^[ 0-9] *$]]></param>
      <message>The second field of the length of the protein must be a
        number.</message>
    </field-validator>
  </field>
```

- Le champ **syn.name** du formulaire est vérifié par un **regex**.
- Affichage de l'erreur sur le formulaire **sans le modifier**

Le Contrôleur - struts.xml

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE struts PUBLIC
    "-//Apache Software Foundation//DTD Struts Configuration 2.0//EN"
    "http://struts.apache.org/dtds/struts-2.0.dtd">

<struts>
    <constant name="struts.enable.DynamicMethodInvocation" value="true" />
    <constant name="struts.devMode" value="false" />

    <package name="actions" extends="struts-default">

        <action name="home">
            <result>/jsp/home.jsp</result>
        </action>

        <action name="help">
            <result>/jsp/help.jsp</result>
        </action>

        <action name="search">
            <result>/jsp/search.jsp</result>
        </action>

        <action name="synthetase" class="actions.SynthetaseAction"
            method="searchSyn">
            <result name="success">/jsp/synthetase.jsp</result>
            <result name="error">jsp/error.jsp</result>
        </action>

        <action name="protein" class="actions.ProteinAction" method="searchProt">
            <result name="success">/jsp/protein.jsp</result>
            <result name="error">jsp/error.jsp</result>
        </action>

        <action name="formAction" class="actions.FormAction" method="searchListOfSyn">
            <result name="success">/jsp/list_synthetases.jsp</result>
            <result name="error">jsp/no_synthetase.jsp</result>
        </action>
    </package>
</struts>
```

L'action de nom
formAction lance
la méthode
searchListOfSyn de
la classe
FormAction

Le Contrôleur - FormAction

```
public class FormAction extends ActionSupport {  
  
    private static final long serialVersionUID = -1711847256080102361L;  
  
    private ArrayList<Synthetase> syns;  
  
    private Synthetase syn;  
  
    private Protein prot;  
  
    private Domain domain;  
  
    private Reference ref;  
  
    private Author auth;  
  
    private Organism orga;  
  
    private String idSyn;  
  
    public String getIdSyn() {  
        return idSyn;  
    }  
  
    public void setIdSyn(String idSyn) {  
        this.idSyn = idSyn;  
    }  
  
    public Author getAuth() {  
        return auth;  
    }  
  
    public void setAuth(Author auth) {  
        this.auth = auth;  
    }  
  
    public Protein getProt() {  
        return prot;  
    }  
}
```

syn.name est initialisé
grâce au setter
setSynthetase dans
l'action

Le Contrôleur - FormAction

```
public String searchListOfSyn() {
    boolean filled = false;
    String action = "error";
    syns = new ArrayList<Synthetase>();

    SynthetaseDAO dao;

    if (prot != null) {
        dao = new SynthetaseDAO();
        ArrayList<Synthetase> synthetasesFromProt = dao
            .identifySynthetases(prot);
        // Si l'objet n'était pas rempli (donc que l'utilisateur n'a rien
        // mis dans ces champs) dans ce cas filled reste a false.
        if (!dao.isNullObject()) {
            syns.addAll(synthetasesFromProt);
            filled = true;
        }
    }
    // Si un objet a bien été rempli mais que la liste est tjs vide : on
    if (filled && syns.size() == 0) {
        return "error";
    } else {
        if (syn != null) {
            dao = new SynthetaseDAO();
            if (syn.getIsComplete().equals("1"))
                syn.setIsComplete("");
            if (idSyn != null && !idSyn.equals("")) {
                try {
                    syn.setId(CreateDORISid.getDBId(idSyn));
                    syn.setDorisId(idSyn);
                } catch (Exception e) {
                    return "error";
                }
            }
        }
        ArrayList<Synthetase> synthetasesFromSyn = dao
            .identifySynthetases(syn);
    }
}
```

- La méthode `searchListOfSyn` est lancée par l'action
- Initialise le DAO qui permet de récupérer la liste des synthetases

Le Modèle - SynthetaseDAO

```
// Find the list of syns that corresponds to the informations inserted
public ArrayList<Synthetase> identifySynthetases(Protein prot) {

    ArrayList<Synthetase> syns = null;
    // initialisation de la requete
    String requeteString = null;
    String sql = "";

    // Verification de ce qui est vide ou non
    if (prot.getName() != null && !prot.getName().equals("")) {
        sql += "LOWER(name_seq) LIKE '%" + prot.getName().toLowerCase()
            + "%'";
    }

    if (prot.getGi() != 0) {
        if (sql != "")
            sql += " AND ";
        sql += "link.acc = '" + prot.getGi()
            + "' AND link.name_db = 'protein'";
    }

    if (prot.getUniprot() != null && !prot.getUniprot().equals("")) {
        if (sql != "")
            sql += " AND ";
        // Deja passe par la validation, n'utilise que majuscules
        // chiffres + lettres
        sql += "link.acc LIKE '" + prot.getUniprot()
            + "%' AND link.name_db = 'uniprot'";
    }

    if (prot.getRefseq() != null && !prot.getRefseq().equals("")) {
        if (sql != "")
            sql += " AND ";
        sql += "link.acc LIKE '" + prot.getRefseq()
            + "%' AND link.name_db = 'refseq'";
    }

    if (prot.getLengthLimit1() != 0) {
        if (prot.getLengthLimit2() != 0) {
            if (sql != "")
                sql += " AND ";
            sql += "length BETWEEN " + prot.getLengthLimit1() + " AND "
                + prot.getLengthLimit2();
        }
    }
}
```

Création de la requête sql à partir du bean **Protein** initialisé grâce à la page **jsp**

Le Modèle - SynthetaseDAO

```
if (!sql.equals("")) {
    nullObject = false;
}

if (!nullObject) {
    try {

        // ouverture connexion
        connexion = super.getConnection();
        // creation requête

        if (sql.contains("link.acc")) {
            requeteString = "SELECT DISTINCT synthetase.id_synthetase, name FROM synthetase, protein, " +
                "link, link_protein WHERE synthetase.id_synthetase = protein.id_synthetase "
                + "AND link_protein.id_protein = protein.id_protein AND " +
                "link.acc = link_protein.acc AND "
                + sql;
        } else {
            requeteString = "SELECT DISTINCT synthetase.id_synthetase, name " +
                "FROM synthetase, protein WHERE synthetase.id_synthetase = " +
                "protein.id_synthetase "
                + "AND " + sql;
        }

        // preparation requête
        request = connexion.prepareStatement(requeteString);
        // requete.setString(1, "" + id);

        // Execution requete
        result = request.executeQuery();

        System.out.println(request.toString());
        // On stocke le resultat dans l'objet utilisateur
        if (result != null) {
            // if (result.next()) {
            // mapping des attributs/champs Sql
            syns = mappingSynthetases(result, true);
            // }
        }
    } catch (Exception e) {
        syns = null;
    }
}
```

- Interrogation de la base
- Puis **mapping du ResultSet** sur les objets des beans
- Retourne la liste des synthetases

Le Modèle - Synthetase.java

```
package beans;

import java.util.ArrayList;

public class Synthetase implements Comparable {

    private int id;
    private String dorisId;
    private String name;
    private String isComplete;
    private Organism orga;
    private ArrayList<Reference> refs;
    private ArrayList<Protein> prots;
    private String idNorine;
    private String linkNorine;

    public String getDorisId() {
        return dorisId;
    }

    public void setDorisId(String dorisId) {
        this.dorisId = dorisId;
    }

    public String getLinkNorine() {
        return linkNorine;
    }

    public void setLinkNorine(String linkNorine) {
        this.linkNorine = linkNorine;
    }

    public String getIdNorine() {
        return idNorine;
    }

    public void setIdNorine(String idNorine) {
        this.idNorine = idNorine;
    }
}
```

Les beans sont formés de :

- leurs attributs
- getters
- setters

Le Contrôleur - struts.xml

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE struts PUBLIC
    "-//Apache Software Foundation//DTD Struts Configuration 2.0//EN"
    "http://struts.apache.org/dtds/struts-2.0.dtd">

<struts>
  <constant name="struts.enable.DynamicMethodInvocation" value="true" />
  <constant name="struts.devMode" value="false" />

  <package name="actions" extends="struts-default">

    <action name="home">
      <result>/jsp/home.jsp</result>
    </action>

    <action name="help">
      <result>/jsp/help.jsp</result>
    </action>

    <action name="search">
      <result>/jsp/search.jsp</result>
    </action>

    <action name="synthetase" class="actions.SynthetaseAction"
      method="searchSyn">
      <result name="success">/jsp/synthetase.jsp</result>
      <result name="error">jsp/error.jsp</result>
    </action>

    <action name="protein" class="actions.ProteinAction" method="searchProt">
      <result name="success">/jsp/protein.jsp</result>
      <result name="error">jsp/error.jsp</result>
    </action>

    <action name="formAction" class="actions.FormAction" method="searchListOfSyn">
      <result name="success">/jsp/list_synthetases.jsp</result>
      <result name="error">jsp/no_synthetase.jsp</result>
    </action>
  </package>
</struts>
```

Le succès de l'action
affiche la page
list_synthetases.jsp

bioinfo.lifl.fr

Bonsai

mreps

YASS

Magnolia

Protea

RNA

Carnac

RNAfamily

Gardenia

REGLISS

CG-seq

TFM

TFM-Explorer

TFM-Scan

TFM-Pvalue

TFM-CUDA

NRPS

Norine

Doris

1 synthetases found

- **Synthetase DOR00015** : hypothetical nonribosomal synthetase
Yersinia pseudotuberculosis IP 32953
 - **Protein DOR00015.1**: yersiniabactin biosynthetic protein
 - GI : 51595932
 - RefSeq : YP_070123.1
 - **Protein DOR00015.2**: yersiniabactin biosynthetic protein
 - GI : 51595933
 - RefSeq : YP_070124.1
 - Uniprot : Q7CI41_YERPE

La Vue - list_synthetases.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ taglib prefix="s" uri="/struts-tags"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Doris::Synthetases</title>
</head>
<body>
  <h3><s:property value="#{syns.size()}" /> synthetases found</h3>
  <ul>
    <s:iterator value="syns" status="synStatus">
      <s:set value="id" name="id_syn" />
      <s:set value="dorisId" name="doris_id_syn" />
      <s:url id="url" action="synthetase">
        <s:param name="idSyn" value="#id_syn" />
      </s:url>
      <li><s:a href="#{url}">
        <b>Synthetase <s:property value="dorisId" /> : </b>
      </s:a> <s:property value="name" /><br>
      <s:push value="orga">
        <i><s:property value="name" /></i>
      </s:push>
    </li>
  </ul>
  <s:iterator value="prots" status="protStatus">
    <s:url id="url2" action="proteine">
      <s:param name="prot.id" value="id" />
      <s:param name="idSyn" value="#id_syn" />
    </s:url>
    <li><s:a href="#{url2}">
      <b>Proteine <s:property value="#doris_id_syn" />. <s:property value="name" />
    </s:a> <s:property value="name" />
    <ul>
      <s:if test="gi != 0">
        <li><s:a href="#{giLink}" target="_blank">GI : <s:property value="gi" />
        </li>
      </s:if>
    </ul>
  </li>
</ul>
```

- Accès aux données de l'action grâce à `<s:property>`
- En cliquant sur l'url associée à l'id de la synthetase, l'action `synthetase` est lancée

Le Contrôleur - Struts.xml

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE struts PUBLIC
    "-//Apache Software Foundation//DTD Struts Configuration 2.0//EN"
    "http://struts.apache.org/dtds/struts-2.0.dtd">

<struts>
    <constant name="struts.enable.DynamicMethodInvocation" value="true" />
    <constant name="struts.devMode" value="false" />

    <package name="actions" extends="struts-default">

        <action name="home">
            <result>/jsp/home.jsp</result>
        </action>

        <action name="help">
            <result>/jsp/help.jsp</result>
        </action>

        <action name="search">
            <result>/jsp/search.jsp</result>
        </action>

        <action name="synthetase" class="actions.SynthetaseAction"
            method="searchSyn">
            <result name="success">/jsp/synthetase.jsp</result>
            <result name="error">jsp/error.jsp</result>
        </action>

        <action name="protein" class="actions.ProteinAction" method="searchProt">
            <result name="success">/jsp/protein.jsp</result>
            <result name="error">jsp/error.jsp</result>
        </action>

        <action name="formAction" class="actions.FormAction" method="searchListOfSyn">
            <result name="success">/jsp/list_synthetases.jsp</result>
            <result name="error">jsp/no_synthetase.jsp</result>
        </action>
    </package>
</struts>
```

L'action de nom **synthetase** lance la méthode **searchSyn** et affiche le résultat dans **synthetase.jsp**

Le Contrôleur - SynthetaseAction

```
public class SynthetaseAction extends ActionSupport {  
  
    private Synthetase syn;  
    private int idSyn;  
  
    public int getIdSyn() {  
        return idSyn;  
    }  
  
    public void setIdSyn(int idSyn) {  
        this.idSyn = idSyn;  
    }  
  
    public Synthetase getSyn() {  
        return syn;  
    }  
  
    public void setSyn(Synthetase syn) {  
        this.syn = syn;  
    }  
  
    // Instanciation du model  
    private SynthetaseDAO modelSyn = new SynthetaseDAO();  
  
    public String searchSyn() {  
        String action = "error";  
  
        if (this.idSyn != 0) {  
            syn = modelSyn.identify(this.idSyn);  
        }  
  
        if (syn != null) {  
            //stockage des proteines pour pour downloadFasta et CreateNrpsImage Action  
            for (int i = 0; i < syn.getProts().size(); i++) {  
                ActionContext.getContext().getSession()  
                    .put(""+i, syn.getProts().get(i));  
            }  
            action = SUCCESS;  
        }  
        return action;  
    }  
}
```

L'objet du bean
Synthetase est
initialisé grâce à la
méthode **identify** du
DAO **SynthetaseDAO**

La Vue - synthetase.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ taglib prefix="s" uri="/struts-tags"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Doris::Synthetase</title>
</head>
<body>

<s:push value="syn">
  <s:set value="id" name="id_syn" />
  <s:set value="dorisId" name="doris_id_syn" />
  <h1>
    <s:property value="name" default="unknown" />
    <s:property value="dorisId" />
  </h1>

  <h2>Synthetase information :</h2>

  <ul>
    <li><b>Doris id :</b> <s:property value="dorisId" />
    </li>
    <li><b>Status of the synthetase :</b> <s:property
      value="isComplete" />
    </li>
    <s:if test="idNorine != null">
      <li>NRP synthetised : <s:a href="{linkNorine}" target="_blank">
        <s:property value="idNorine" />
      </s:a>
    </li>
    </s:if>
  </ul>

  <h2>Organism :</h2>

  <ul>
    <s:if test="orga != null">
      <s:push value="orga">
        <s:set name="nameOrga" value="name" />
        <li><b>Name :</b> <s:property value="name" default="unknown" />
        </li>
      </s:push>
    </s:if>
  </ul>
</body>
</html>
```

- Résultat de l'action synthetase
- Affichage des valeurs des beans stockées dans l'action grâce à `<s:property>`
- Exemple :
`syn.name` est stocké dans la valeur name de l'objet du bean Synthetase, attribut de l'action.

hypothetical nonribosomal synthetase DOR00015

bioinfo.lifl.fr

Bonsai

mreps

YASS

Magnolia

Protea

RNA

Carnac

RNAfamily

Gardenia

REGLISS

CG-seq

TFM

TFM-Explorer

TFM-Scan

TFM-Pvalue

TFM-CUDA

NRPS

Norine

Doris

. Synthetase information :

- **Doris id** : DOR00015
- **Status of the synthetase** : unverified

. Organism :

- **Name** : Yersinia pseudotuberculosis IP 32953
- **Taxonomy** : Bacteria; Proteobacteria; Gammaproteobacteria; Enterobacteriales; Enterobacteriaceae; Yersinia
- **Taxid** : 273123
- **Genome link** : [NC_006155.1](#)

. References :

- No references

. Proteins :

- **Protein DOR00015.1** :



[Proteic sequence](#)

[Nucleic sequence](#)

- **Name** : yersiniabactin biosynthetic protein
- **Molecular weight** : 348653 Da
- **Length** : 3163
- **GI** : 51595932
- **RefSeq** : YP_070123.1
 - [link to the protein](#)
 - [link to the gene](#)
- **Uniprot** : Unknown
- **Positions on the genome (NC_006155.1)** : [1924950 to 1934441](#)
- **Strand** : minus
- **Predictions** :
 - **Prediction 1** :

- **Predictions :**

- **Prediction 1 :**

- **Score :** 0

- **Program name :** Automatic annotation by NCBI with CD-Search

- **Structure :**

<i>Domains</i>	KS	AT	M	CP	C	M	CP	Te
<i>On the protein</i>	29 to 447	554 to 843	1093 to 1183	1822 to 1889	1915 to 2210	2593 to 2671	2831 to 2894	2914 to 3160
<i>On the nucleic sequence</i>	87 to 1341	1662 to 2529	3279 to 3549	5466 to 5667	5745 to 6630	7779 to 8013	8493 to 8682	8742 to 9480



- **Protein DOR00015.2 :**



Proteic sequence

Nucleic sequence

- **Name :** yersiniabactin biosynthetic protein

- **Molecular weight :** 228521 Da

- **Length :** 2035

- **GI :** 51595933

- **RefSeq :** YP_070124.1

- [link to the protein](#)

- [link to the gene](#)

- **Uniprot :** Q7CI41_YERPE

- **Positions on the genome (NC_006155.1) :** 1934529 to 1940636

- **Strand :** minus

- **Predictions :**

Conclusion



- Struts 2 est une **grosse machinerie**.
- Un framework très pratique lorsque l'on a besoin de gérer **beaucoup de données** sur le web.
- Fournit une structure claire qui permet d'empaqueter nos données au format objet et d'y **accéder facilement**.
- Une fois pris en main, permet de comprendre le code très rapidement.
- Très utile dans le cas de bases de données !

Questions

