

☒ Adam Bernouy

☐ Bryan Deloison

☐ Florent Dewilde

☐ Martin Malet

☐ Jonathan Selle

Générateur de formulaire GUI

☒ Documentation utilisateur

Equipe : Infotik
Date : 15/01/2018
Annee : 2017-2018
Projet : Projet tutoré S3
Enseignants : M. Legrix, M. Le Pivert,
Mme. Boukachour, M. Duflo

Valider

Effacer



Introduction

Dans le cadre du projet tutoré de deuxième année de DUT informatique, nous avons développé un générateur de formulaire GUI destiné aux premières années de DUT informatique. En effet, ils pourront créer de manière simple des formulaires graphiques et récupérer les données qu'ils rentreront.

Ce document utilisateur va vous permettre de découvrir le fonctionnement du générateur de formulaire graphique. Nous verrons donc comment installer tous les fichiers nécessaires pour le bon fonctionnement du programme. Ensuite, nous vous expliquerons comment créer un fichier XML qui sera ensuite traité par le programme pour afficher le formulaire. Enfin, nous vous expliquerons comment utiliser les commandes du programme pour pouvoir afficher le formulaire, ou bien récupérer les valeurs que vous avez rentré dans celui-ci.

Vous trouverez dans une dernière partie des fonctionnalités supplémentaires pour aller un peu plus loin, comme notamment afficher le type et les id de tous les éléments présents sur le formulaire ou afficher une prévisualisation du formulaire dans un navigateur. Quelques exemples avec leurs rendues sont disponibles à la fin du document utilisateur.

Table des matières

I. Installation	1
II. Création du document XML	2
1. Patron de fichier XML	2
2. Déclaration des éléments par type	3
III. Prévisualisation du formulaire dans un navigateur	5
IV. Récupération des données du formulaire dans un code JAVA.....	6
1. Création du formulaire	6
2. Modification des valeurs	6
3. Affichage de la fenêtre	7
4. Récupération des valeurs.....	8
V. Pour aller plus loin.....	10
1. Choix de la langue du formulaire	10
2. Alternative à la création du formulaire.....	11
3. Affichage des identifiants et des types.....	11
VI. Exemples	12

I. Installation

Tout d'abord afin de vous servir de nos outils, vous devez avoir déjà téléchargé le package iut.jar ainsi que le fichier form.xml, form.dtd et form.css disponibles à [cette](#) adresse.

Si vous n'avez pas procédé à l'installation de JAVA sur votre machine, veuillez la suivre à l'adresse suivante : [installation de java](#) . Vous êtes désormais prêt à vous servir de nos outils.

II. Création du document XML

La première étape est de créer votre formulaire, pour ce faire vous devez tout d'abord ouvrir votre éditeur de texte préféré et créer un fichier « nomdefichier.xml ».

1. Patron de fichier XML

Copier-Coller cette entête dans votre fichier (vous n'avez nul besoin de savoir à quoi celui-ci sert, vos professeurs vous l'expliqueront très bien en 2^{ème} année en cours de XML) :

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE form SYSTEM "form.dtd">

<?xml-stylesheet type="text/xsl" href="form.xsl"?>
```

Désormais vous pouvez commencer à déclarer votre formulaire. Le XML devra se présenter ainsi :

```
<form>
    <fenetre longueur="650" titre="test" largeur="1340"
x="100" y="100">
    <type option1="500" option2="a01" />
        <menu option1="option">
            <choix option1= "2"/>
            <choix option1="1"/>
        </menu>
        [...]
    </fenetre>
</form>
```

Les paramètres de la balise « fenetre » permettent de la positionner (x, y) et de lui donner une taille (longueur, largeur). Le « titre » définit le titre du formulaire. L'ensemble de ces paramètres sont obligatoires.

2. Déclaration des éléments par type

- **case (checkbox) :**

```
<case x="36" y="30" label="texte" id="a01"/>
```

La balise « case » permet de créer une case à cocher.

- **texte (text) :**

```
<texte x="36" y="60" label="groupeBoutton" id="a02"  
type="entier"/>
```

La balise « texte » permet de créer une zone de texte renvoyant son contenu du type défini dans sa déclaration.

- **menu (dropdown) :**

```
<menu x="36" y="120" label="deroulant" id="a03">  
  <choix label="ah"/>  
  <choix label="oh"/>  
  <choix label="uh"/>  
</menu>
```

La balise « menu » permet de déclarer un menu déroulant avec différent choix déclarés avec les balises « choix ».

- **tableau (array) :**

```
<tableau x="500" y="120" label="box" id="a04"  
type="booleen" nb_lig="7" nb_col="8"/>
```

nb_lig : nombre de lignes dans le tableau
nb_col : nombre de colonne dans le tableau

Attention : ces deux paramètres sont obligatoires.

- **boutons (buttons) :**

```
<boutons x="36" y="150" id="a05" label="bouton">  
  <bouton>25</bouton>  
</boutons>
```

La balise « boutons » crée une liste de « bouton » : attention sur cette liste, un seul bouton pourra être coché.

- **label (label) :**

```
<label x="36" y="0" label="Mon label seul" id="a06"/>
```

La balise « label » permet de créer un texte affiché à l'écran que l'utilisateur ne pourra modifier.

- **calendrier (calendar) :**

```
<calendrier x="300" y="300" id="a07" label="date"/>
```

La balise « calendrier » permet de créer un sélecteur de date.

Maintenant nous allons voir à quoi servent chacune des options :

- | | | |
|---------------|---|---|
| x et y | : | Définissent la position d'un élément. Attention si un seul élément n'a pas de x ou de y défini, les éléments s'afficheront tous par défaut les uns en dessous des autres. |
| id | : | cette option est obligatoire pour chaque élément, il permettra dans le futur de récupérer les informations rentrées dans le formulaire et doit toujours être de la forme "axx" sachant qu'il ne peut y avoir 2 id identiques dans le même formulaire. |
| label | : | cela permet de donner un nom devant chaque élément et est aussi obligatoire. |
| type | : | permet de définir le type de retour, il est obligatoire pour les éléments texte et tableau. Les valeurs possibles sont : chaine, entier, double, boolean, caractere. |

III. Prévisualisation du formulaire dans un navigateur

Nous offrons la possibilité à l'utilisateur de pouvoir prévisualiser le résultat dans un navigateur du formulaire crée sans passer par le programme java. Pour cela il suffira simplement d'ouvrir le fichier XML avec un navigateur, Mozilla de préférence car chrome ou encore Edge ne prendront pas en compte certaines fonctionnalités.

Vous avez normalement dû télécharger le fichier « **form.xsl** » et « **form.dtd** » se trouvant dans la partie installation. Placez-les ensuite au même niveau que votre fichier XML.

Vous pouvez à présent visualiser votre formulaire dans votre navigateur en cliquant sur le fichier XML. Nous rappelons que cette fonctionnalité ne sert que pour visualiser le formulaire et non pour récupérer des valeurs. De plus il est possible que le formulaire généré soit modifié légèrement lors de l'utilisation du programme en java.

Vous trouverez un exemple d'affichage dans la partie Exemple.

IV. Récupération des données du formulaire dans un code JAVA

1. Création du formulaire

La troisième étape est d'intégrer le formulaire à votre code JAVA. Avant toute chose, vous devez importer notre outil par la commande :

```
import iut.algo.form.FormController;
```

Cette commande se fait en haut de votre fichier.

Pour créer un formulaire, il suffit :

```
FormController          fm          =  
FormController.createAndGetForm("exemple.xml");
```

Cette méthode vous retournant un `FormController` vous devrez désormais utiliser le retour de cette méthode pour utiliser les autres méthodes. (Vous avez d'autres possibilités de créations de formulaire dans le chapitre Pour aller plus loin)

2. Modification des valeurs

Ensuite vous avez la possibilité de modifier les valeurs des éléments comme ceci :

```
fm.setValue("a05", "test");
```

3. Affichage de la fenêtre

Pour afficher la frame, on utilise :

```
fm.showForm();
```

Affiche le formulaire et stoppe le programme utilisateur.

Attention :

Si l'utilisateur a quitté sans remplir le formulaire, vous pouvez obtenir des erreurs lors de la récupération des données. Vous avez donc à votre disposition une méthode permettant de savoir si l'utilisateur a quitté ou validé :

```
boolean b = fm.isValid();
```

4. Récupération des valeurs

Une fois que le formulaire est validé ou quitté, le programme, jusque-là en pause, reprend. Vous pouvez donc procéder à la récupération de vos données. Vous pouvez aussi appuyer sur le bouton effacer pour réinitialiser le formulaire.

Pour récupérer des valeurs, on utilise pour les **int** :

```
int i = fm.getInt("02");
```

Attention : Les « **id** » ne sont que les chiffres présents dans l'id du fichier XML.

Exemple : a02 → 02

Il existe donc autant de commandes qu'il y a de types de retour.

- `getInt(id);`
- `getString(id);`
- `getDouble(id);`
- `getChar(id);`
- `getBoolean(id);`
- `getStringArray(id, tableau* de String)`
- `getCharArray(id, tableau* de char)`
- `getBooleanArray(id, tableau* de Boolean)`
- `getIntArray(id, tableau* de Int)`
- `getDoubleArray(id, tableau* de double)`
- `getArray(id);`

* Les tableaux peuvent être déclarés avec une ou deux dimensions. Les `getArray` renvoient `true` si le tableau a été copié et `false` en cas de problème lors de la création.

Attention : Pour récupérer le bouton qui a été sélectionné dans la balise « boutons », vous devez faire `getInt(id)` car ceux-ci renvoient le numéro, passé en paramètre lors de leur création, de celui sélectionné.

Afin de récupérer les valeurs du tableau, il vous faut lancer cette commande:

```
Object[][] tableau = fm.getArray("04");
```

Pour fermer définitivement le formulaire, on utilisera cette primitive :

```
fm.close();
```

Attention : Il sera impossible d'ouvrir de nouveau le formulaire après le passage de cette commande. Cette fermeture est définitive.

V. Pour aller plus loin...

Notre outil vous offre plusieurs options supplémentaires vous permettant d'aller plus loin.

1. Choix de la langue du formulaire

Premièrement vous avez possibilité de changer la langue de votre formulaire, en choisissant entre **anglais** et **français**.

Dans ce cas il vous suffira juste de changer les mots :

- fenetre → window
- longueur → length
- largeur → width
- titre → title
- texte → text
- chaine → string
- entier → int
- booleen → boolean
- caractere → char
- menu → dropdown
- choix → choice
- case → checkbox
- tableau → array
- nb_lig → nb_row
- bouton → button
- boutons → buttons
- calendrier → calendar

Le reste ne change pas.

2. Alternative à la création du formulaire

Pour ne pas utiliser **FormController.createAndGetForm("")**, vous avez la possibilité d'utiliser ces deux méthodes :

```
String id = FormController.createForm("exemple.xml");
```

Cette méthode vous retourne un id de FormController à utiliser dans la prochaine méthode.

```
FormController fm = FormController.getForm(id);
```

Cette méthode vous retourne alors le FormController à utiliser pour les autres méthodes.

3. Affichage des identifiants et des types

Lorsque votre formulaire est activé, vous pouvez afficher les types de chaque élément ainsi que leurs id.

Pour afficher les types :

```
CTRL + T
```

Pour afficher les ID :

```
CTRL + I
```

VI. Exemples

Demo.java :

```
1  import iut.algo.form.FormController;
2  /**
3   * classe qui permet d'afficher un formulaire de test
4   * @author Team Infotik
5   * @version 2018-01-08
6   */
7  public class Demo
8  {
9      public static void main (String[] args)
10     {
11         /* CREATION ET AFFICHAGE DU FORMULAIRE */
12         FormController fm = FormController.createAndGetForm("exemple.xml");
13         fm.setValue("6", new String[] {"test1","test2","test3"});
14
15         if ( fm.isValid() )
16         {
17             String s = fm.getString("05");
18             System.out.println(s);
19
20             boolean[][] tab = new boolean[2][4];
21             System.out.println( fm.getArrayBoolean("11", tab) );
22             for (boolean[] boolA : tab)
23             {
24                 for (boolean boolB : boolA)
25                 {
26                     System.out.print( boolB + ", " );
27                 }
28             }
29         }
30         else
31         {
32             System.out.println("ERREUR : Vous avez quitté sans valider.");
33         }
34     }
35 }
```


exemple.xml :

```
1  <form>
2      <fenetre longueur="650" titre="test" largeur="1340" x="100" y="100">
3          <label x="36" y="0" label="Un label seul" id="a01"/>
4
5          <case x="36" y="30" label="Choix binaire" id="a02"/>
6
7          <texte x="36" y="60" label="Un entier" id="a03" type ="entier"/>
8
9          <texte x="36" y="90" label="Un caractère" id="a04" type ="caractere"/>
10
11         <texte x="36" y="120" label="Une chaîne" id="a05" type ="chaîne"/>
12
13         <menu x="36" y="150" label="Menu déroulant" id="a06">
14             <choix label="Option 1"/>
15             <choix label="Option 2"/>
16             <choix label="Option 3"/>
17         </menu>
18
19         <boutons x="500" y="30" id="a07" label="Radio boutons">
20             <bouton>Premier choix</bouton>
21             <bouton>Second choix</bouton>
22         </boutons>
23
24         <tableau x="500" y="180" label="Tableau de bool" id="a08" type="entier" nb_lig="7" nb_col="8"/>
25
26         <calendrier x="500" y="120" id="a09" label="Date"/>
27     </fenetre>
28 </form>
29
```

Résultat :

Un label seul

Choix binaire : ☐

Un entier :

Un caractère :

Une chaîne :

Menu déroulant :

Radio boutons

☒ Premier choix
☐ Second choix

Date :

Tableau de bool :

4							
3							
2							
1							
0							
	0	1	2	3	4		

Valeur :

Valider Effacer

Exemple pour la prévisualisation XML :

```
1 <?xml version="1.0" ?>
2 <!DOCTYPE form SYSTEM "Forme.dtd">
3 <?xml-stylesheet type="text/xsl" href="preview.xsl"?>
4
5 <form>
6   <window length="650" title="Exemple de formulaire" width="1340" x="100" y="100">
7     <label x="36" y="0" label="Un label seul" id="a01"/>
8
9     <checkbox x="36" y="30" label="Choix binaire" id="a02"/>
10
11    <text x="36" y="60" label="Un entier" id="a03" type="int"/>
12
13    <text x="36" y="90" label="Un caractère" id="a04" type="char"/>
14
15    <text x="36" y="120" label="Une chaîne" id="a05" type="string"/>
16
17    <dropdown x="36" y="150" label="Menu déroulant" id="a06">
18      <choice label="Option 1"/>
19      <choice label="Option 2"/>
20      <choice label="Option 3"/>
21    </dropdown>
22
23    <buttons x="500" y="30" id="a07" label="Radio boutons">
24      <button ordinal="4">Quatrieme choix</button>
25      <button ordinal="1">Premier choix</button>
26      <button ordinal="3">Troisieme choix</button>
27      <button ordinal="2">Deuxieme choix</button>
28    </buttons>
29
30    <array x="500" y="210" label="Tableau de bool" id="a11" type="boolean" nb_row="4" nb_col="5"/>
31
32    <calendar x="500" y="160" id="a09" label="Date"/>
33  </window>
34 </form>
35
```

Rendu XSL :

Un label seul

Choix binaire : ☐

Un entier :

Un caractère :

Une chaîne :

Menu déroulant :

Radio boutons

Premier choix ☐

Deuxieme choix ☐

Troisieme choix ☐

Quatrieme choix ☐

Date : 16 / 01 / 2018

Tableau de bool :

3	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	0	1	2	3	4

Valeur : ☐