



Tutoriel

Comment générer un fichier java à partir d'un fichier XML



Sommaire

Présentation de l'application :	3
Dans ce tutoriel, on va expliquer exactement comment un fichier XML peut être utilisé afin de générer facilement et automatiquement une classe java.....	3
Avant de commencer ce tutoriel, voici un exemple de la génération d'une classe java à partir d'un fichier XML :.....	3
Comment créer le fichier java à partir de votre fichier XML ?	4



Présentation de l'application :

La version 2.1/A de ZeroCouplage est finalement prête pour emploi. Avant, le codage en utilisant ZeroCouplage sans avoir une bonne connaissance en java mène le développeur en une situation difficile.

Solution : Coder en XML

- Le langage XML a beaucoup d'avantages, citant parmi eux :
- C'est une méthode pour structurer des données
- Compréhensible
- Ressemble un peu à HTML
- Plus puissant
- ...

Dans ce tutoriel, on va expliquer exactement comment un fichier XML peut être utilisé afin de générer facilement et automatiquement une classe java.

Avant de commencer ce tutoriel, voici un exemple de la génération d'une classe java à partir d'un fichier XML :

```
<Page name="page" text="inscription" title="Inscription">
  <GridLayout name="layout" rows="4" cols="1" text="sign in">
    <Label name="labelOfLogin" label="Entrer Login :">
    </Label>

    <Button name="valider" text="Valider" action="goTraitementInscription">
    </Button>

    <Button name="reset" text="Valider" action="goInscription">
    </Button>

  </GridLayout>
</Page>
```

```
package ma.ensao;

import org.zerocouplage.component.api.view.ZCView;

public class Inscription implements ZCView {
    private ZCPage page;
    private ZCGridLayout layout;
    private ZCLabel labelOfLogin;
    private ZCButton valider;
    private ZCButton reset;

    @Override
    public ZCPage display() throws IOException, ZCExceptionConfig, ZCComponentNotFoundException, Exception {
        page = (ZCPage) ZCComponentFactory.newComponent(ZCPage.class);
        page.setName("inscription");
        page.setTitle("Inscription");
        layout = (ZCGridLayout) ZCComponentFactory.newComponent(ZCGridLayout.class);
        page.setBody(layout);
        layout.setName("sign in");
        layout.setRows(4);
        layout.setCols(1);
        labelOfLogin = (ZCLabel) ZCComponentFactory.newComponent(ZCLabel.class);
        layout.addComponent(labelOfLogin);
        labelOfLogin.setLabel("Entrer Login :");
        valider = (ZCButton) ZCComponentFactory.newComponent(ZCButton.class);
        layout.addComponent(valider);
        valider.setAction("goTraitementInscription", this);
        valider.setText("Valider");
        reset = (ZCButton) ZCComponentFactory.newComponent(ZCButton.class);
        layout.addComponent(reset);
        reset.setAction("goInscription", this);
        reset.setText("Valider");
        return page;
    }
}
```

inscription.xml

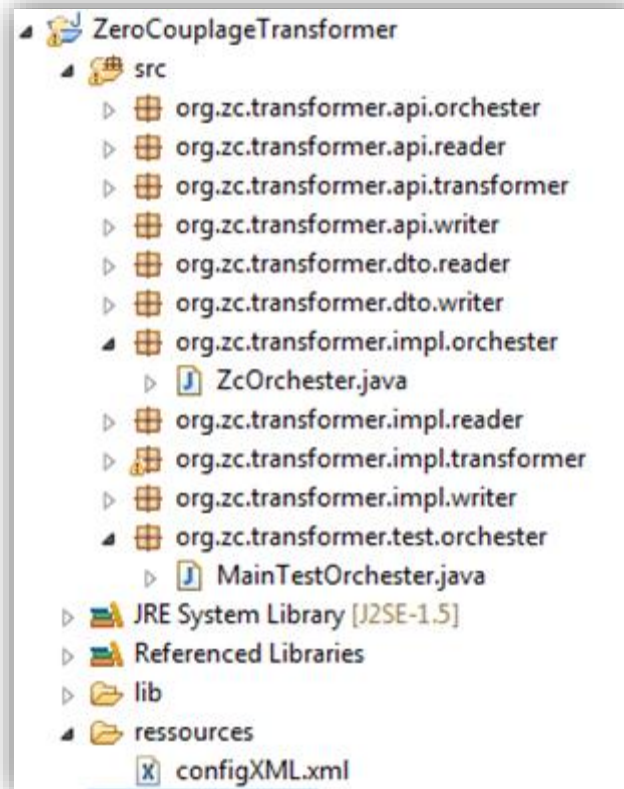
Transformer

inscription.java



Comment créer le fichier java à partir de votre fichier XML ?

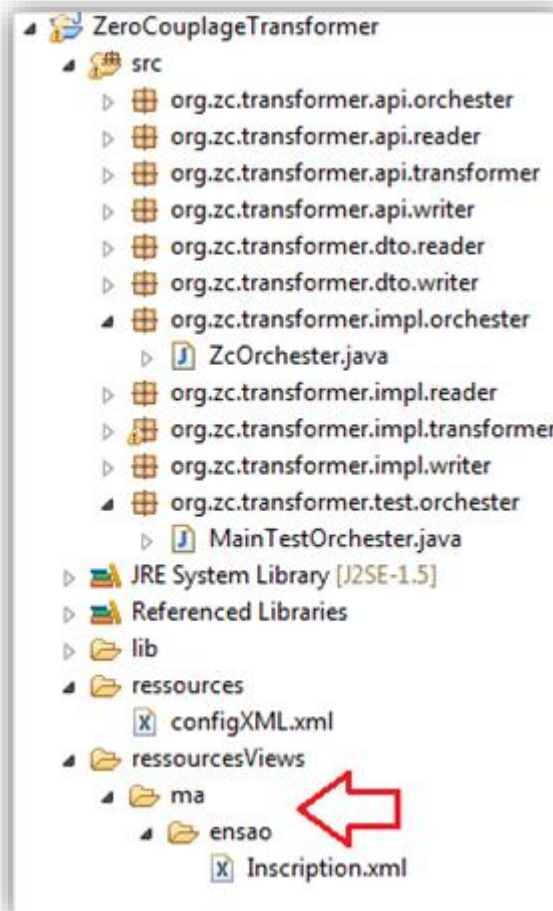
1. Importer le projet de ZeroCouplage à partir du lien suivant :
(Le lien de SVN ou GitHub)
2. La structure du projet doit être comme suit :



3. Le fichier **ConfigXML.xml** : contient tous les ZcComposant avec leurs équivalents en langage java.
4. Le fichier **MainTestOrchester.java** : est la classe main qui instancie et déclenche la classe **ZcOrchester.java**.
5. Créez un dossier, par exemple le dossier « **ressourceViews** », dont vous pouvez créer d'autres dossiers mais il faut prendre en compte que ces dossiers seront générés en tant que le nom du package où la classe java finale sera stockée.

Par exemple :

ressourceViews/ma/ensao → package : ma.ensao



6. Ouvrir la classe **ZcOrchestrer.java** ensuite changer les paths 1 et 2 :

```
ZCOrchestrer.java
package org.zc.transformer.impl.orchester;

import java.io.File;

public class ZcOrchestrer {

    private String[] listpathViewXML;
    //File [] listpathViewXML;
    private ReaderDTO readerDTO;
    private ConfigDTO configDTO;
    private JavaDTO javaDTO;
    private ZCReader zcreader;
    private ZCTransformer zctransformer;
    private ZcWriter zcwriter;
    private ZCReaderConfig zcReaderConfig;

    public void ochester(File pathViewFolder, String outputFolder) {

        listpathViewXML = pathViewFolder.list();
        for(int i=0;i<listpathViewXML.length;i++){
            if(listpathViewXML[i].endsWith(".xml")==true){
                String pathView2 = listpathViewXML[i];
                String path = pathViewFolder+"/"+pathView2;
                String pathZCReader = path.replace("\\", "/");

                zcreader = new ZCReader();
                readerDTO = zcreader.ReadView(pathZCReader, "D:/ENSAO/GI4_S2/PFA/ZC_Projects/ZeroCouplageTransformer/ressourcesViews/");
                zcReaderConfig = new ZCReaderConfig();
                configDTO = zcReaderConfig.read("D:/ENSAO/GI4_S2/PFA/ZC_Projects/ZeroCouplageTransformer/ressources/configXML.xml");

                zctransformer=new org.zc.transformer.impl.transformer.ZCTransformer();
                javaDTO = new JavaDTO();
                javaDTO=zctransformer.transformer(readerDTO, configDTO);
                zcwriter = new ZcWriter();

                zcwriter.write(javaDTO, outputFolder);
            }
        }
    }
}
```



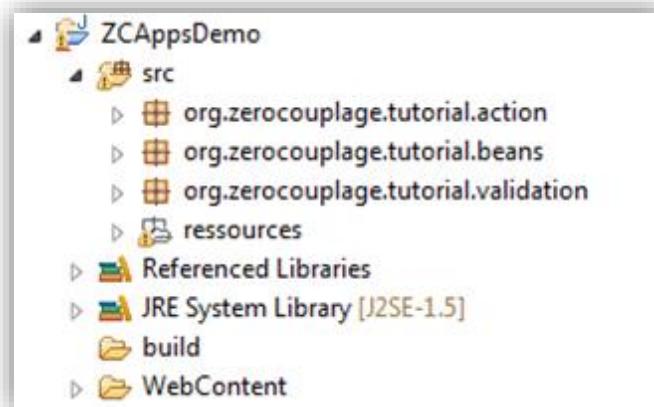
- **Path1** : c'est le chemin du dossier « **ressourcesViews** » dans votre machine.
- **Path2** : c'est le chemin du fichier **configXML.xml** dans votre machine.

Comment récupérer ces chemins ?

Cela en cliquant sur le fichier ou le dossier → propriétés → location → récupérer le chemin.

7. Importez le projet du demo (ZcAppsDemo) à partir du lien suivant :
(Le lien de SVN ou GitHub)

La structure du projet doit être comme suit :



8. Retournez au projet **ZeroCouplageTransformer** et changez dans la classe main **MainTestOrchestrer.java**, les deux chemins 1 et 2.
 - Path1 : chemin vers votre fichier XML « **inscription.xml** »
 - Path2 : chemin vers le dossier « **src** » du projet « **ZCAppsDemo** ».

**MainTestOrchestrer.java**

```
package org.zc.transformer.test.orchestrer;

import java.io.File;

public class MainTestOrchestrer {

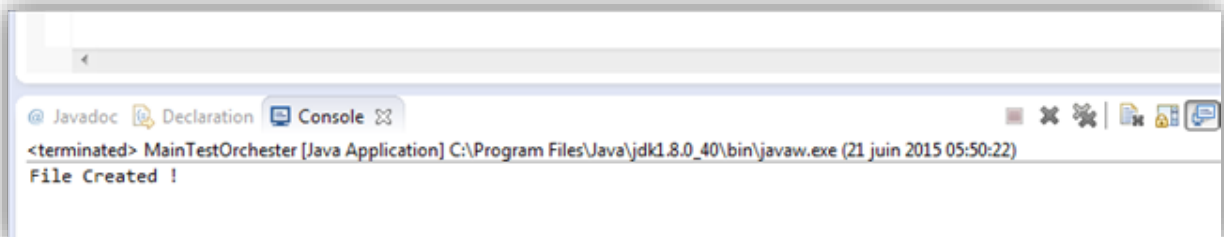
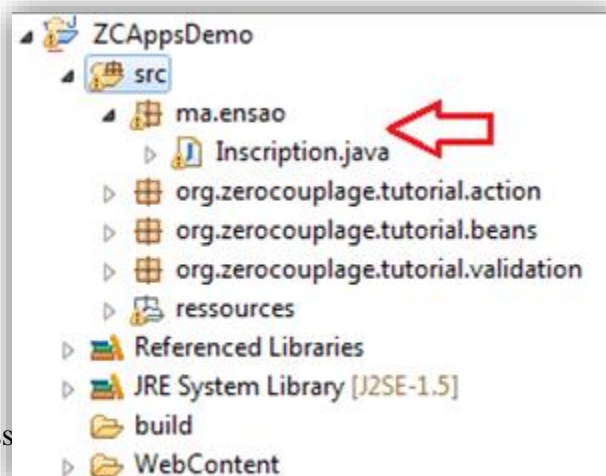
    public static void main(String[] args) {

        ZcOrchestrer zcOrchestrer = new ZcOrchestrer();
        File pathview = new File("D:/ENSAO/GI4_S2/PFA/ZC_Projects/ZeroCouplageTransformer/ressourcesViews/ma/ensao/");
        zcOrchestrer.orchestrer(pathview, "D://ENSAO//GI4_S2//PFA//ZC_Projects//ZCAppsDemo//src");
    }
}
```

9. Exécuter la classe **MainTestOrchestrer.java**.

Si le message « fichier créer » est affiché, passez à l'étape 7.

Si vous avez eu une erreur, essayez de les résoudre d'abord.

10. Rafraîchissez votre projet en sélectionnant le dossier « **src** » de votre projet et cliquer sur F5.

Maintenant, votre class

ao »



```
Inscription.java
package ma.ensao;

import org.zerocouplage.component.api.view.ZCView;

public class Inscription implements ZCView {
    private ZCPage page;
    private ZCGridLayout layout;
    private ZCLabel labelOfLogin;
    private ZCButton valider;
    private ZCButton reset;

    @Override
    public ZCPage display() throws IOException, ZCExceptionConfig, ZCCompNotFoundException, Exception {
        page =(ZCPage) ZCComponentFactory.newComponent(ZCPage.class);
        page.setName("inscription");
        page.setTitle("Inscription");
        layout =(ZCGridLayout) ZCComponentFactory.newComponent(ZCGridLayout.class);
        page.setBody(layout);
        layout.setName("sign in");
        layout.setRows(4);
        layout.setCols(1);
        labelOfLogin =(ZCLabel) ZCComponentFactory.newComponent(ZCLabel.class);
        layout.addComponent(labelOfLogin);
        labelOfLogin.setLabel("Entrer Login :");
        valider =(ZCButton) ZCComponentFactory.newComponent(ZCButton.class);
        layout.addComponent(valider);
        valider.setAction("goTraitementInscription",this);
        valider.setText("Valider");
        reset =(ZCButton) ZCComponentFactory.newComponent(ZCButton.class);
        layout.addComponent(reset);
        reset.setAction("goInscription",this);
        reset.setText("Valider");
        return page;
    }
}
```

FIN