

Manuel d'utilisateur

Ce document décrit la manière d'utiliser et de compiler le générateur de code. Assurez-vous d'avoir installé la dernière version de java et la dernière version de maven pour que le générateur puisse s'exécuter.

Comment utiliser le générateur?

1. Fichier SCXML :

Mettez votre fichier scxml dans le dossier resources fournis à la racine du dossier de ce projet.

2. Squelette d'implémentation du programme :

Créez une classe java dans le package *finite.state.machine*. Cette classe doit étendre la classe *MyWorkspaceImpl*. Vous devez fournir le chemin du fichier scxml lors de l'appel *super()* dans le constructeur dans cette classe. Ensuite, vous devez appeler la méthode *start()* de la classe *MyWorkspaceImpl* en fournissant la classe courante comme argument.

```
package finite.state.machine;

import finite.state.machine.workspace.MyWorkspaceImpl;

public class MyStateMachine extends MyWorkspaceImpl {

    private static final String PATH = "resources/simple.scxml";

    public MyStateMachine(){
        super(PATH);
        start(this);
    }
}
```

Exemple d'implémentation de base du programme

3. Déclaration des événements :

Les événements sont représentés sous forme de méthodes dans ce programme. Les méthodes doivent être du type void et n'acceptent aucun argument. Une fois que les méthodes sont écrites, utilisez l'annotation *@FSMEvent* pour indiquer au programme que cette méthode est un événement et spécifiez le nom de l'événement dans l'argument *event* de l'annotation.

```
@FSMEvent(event="print-hello")
public void print_hello(){
    System.out.println("Hello");
}

@FSMEvent(event="print-bye")
public void print_bye(){
    System.out.println("Bye");
}
```

Exemple de déclaration d'un événement

4. Invocation des transitions :

Pour invoquer des transitions, utilisez la méthode *submitEvent()* de la classe *MyWorkspaceImpl* en donnant le nom de transition visée dans l'argument.

```
public static void main(String[] args){
    MyStateMachine stateMachine = new MyStateMachine();
    stateMachine.submitEvent("Button1");
    stateMachine.submitEvent("Button2");
}
```

Exemple d'invocation d'une transition

Comment compiler le générateur?

Exécutez le fichier *build.sh* pour compiler le programme.

```
Danials-MacBook-Pro:FSMGenerator danial$ ./build.sh
Building FSM Generator ...

-----
T E S T S
-----
Running finite.state.machine.generator.StateGeneratorTest
Tests run: 6, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 1.102 sec

Results :

Tests run: 6, Failures: 0, Errors: 0, Skipped: 0
Danials-MacBook-Pro:FSMGenerator danial$
```

Exemple d'exécution du fichier build.sh

Comment exécuter le générateur?

Exécutez le fichier run.sh pour lancer le programme en donnant le nom de la classe contenant *main*.

```
Danials-MacBook-Pro:FSMGenerator danial$ ./run.sh MyStateMachine
Compiling the FSM project ...
Hello
Bye
Danials-MacBook-Pro:FSMGenerator danial$
```

Exemple d'exécution du fichier run.sh