

Code Inspection

February 5, 2017



**POLITECNICO
DI MILANO**

Version 1.0

- Davide Alessandro Cottica (mat. 806868)
- Stefano Antonino Badalucco (mat. 809322)

Contents

1	Assigned class	3
2	Functional Role	3
3	List of Issues	4
3.1	ModelFormAction class	4
3.2	readSubActions: Line 48 \Rightarrow 71	4
3.3	CallParentAction nested class	4
3.4	accept: Line 101 \Rightarrow 103	4
3.5	runAction: Line 106 \Rightarrow 116	4
3.6	Service nested class	4
3.7	accept: Line 171 \Rightarrow 174	5
3.8	getServiceName: Line 176 \Rightarrow 178	5
3.9	runAction: Line 181 \Rightarrow 246	5
3.10	getAutoFieldMapExdr: Line 248 \Rightarrow 250	5
3.11	getFieldMap: Line 252 \Rightarrow 254	5
3.12	getIgnoreError: Line 256 \Rightarrow 258	5
3.13	getResultMapListNameExdr: Line 260 \Rightarrow 262	5
3.14	getResultMapNameAcsr: Line 264 \Rightarrow 266	5
3.15	getServiceNameExdr: Line 268 \Rightarrow 270	5
4	Other Issues	6
5	Hours of work	7

1 Assigned class

The class assigned to our group for code inspection is the abstract class `ModelFormAction`, which contains nested two classes: **`CallParentActions`** e **`Service`** which extend another class, `AbstractModelAction`.

2 Functional Role

This specific class task is to analyze a specific `ModelForm` and `Element` in order to extract some useful information from them. The `Element` object is parsed as an XML (as evidence we can observe the use of some static methods from `UtilXML` class).

The only method specified in the abstract class returns a List of object extracted from the `Element`, subdivided by type (`Service`, `CallParentAction` or generic action instance). Every different object of the list can perform an action: this is managed separately by each class extending `AbstractModelAction` and is triggered overriding the method `runAction(Map<String, Object> map)`.

3 List of Issues

3.1 ModelFormAction class

3.2 readSubActions: Line 48 \Rightarrow 71

- This method is not commented.
- In the for statement there are three nested if statement that cause a poor readability.
- **NullPointerException** not thrown nor caught from the method.
- **L.48, L.49, L.54** exceed 80 characters.
- **L.49** Not useful variable declaration: only used to extract list length in L.50.

3.3 CallParentAction nested class

- **L.91, L.95** exceed 80 characters.

3.4 accept: Line 101 \Rightarrow 103

- This method is not commented.

3.5 runAction: Line 106 \Rightarrow 116

- This method is not commented.
- **L. 108** the switch statement has not default case.

3.6 Service nested class

- **NullPointerException** not thrown nor caught.
- **L.139, L.140, L.141, L.142, L.145,L.146, L.161** exceed 80 characters.
- **L.142** Not useful variable declaration. Same name attribute was enough to perform the operations.
- **L.143** there are nested if statement that cause a poor readability.
- **L.147 - L.150** Duplicated code. See line 58 - 61.
- **L.152, L.160** Else block code perform the same operations. Recommended use of a String variable to be passed as a parameter for the function.
- **L.156, L.162** missing parenthesis for if statement block code.

3.7 accept: Line 171 \Rightarrow 174

- This method is not commented.

3.8 getServiceName: Line 176 \Rightarrow 178

- This method is not commented.

3.9 runAction: Line 181 \Rightarrow 246

- This method is not commented.
- This method is 66 lines long. If possible the dimension should be lower.
- **L.184, L.192, L.203, L.207, L.209, L.231, L.239** exceed 80 characters.
- **L.220** Caught error not managed by class (no code inside).

3.10 getAutoFieldMapExdr: Line 248 \Rightarrow 250

- This method is not commented.

3.11 getFieldMap: Line 252 \Rightarrow 254

- This method is not commented.

3.12 getIgnoreError: Line 256 \Rightarrow 258

- This method is not commented.

3.13 getResultMapListNameExdr: Line 260 \Rightarrow 262

- This method is not commented.

3.14 getResultMapNameAcsr: Line 264 \Rightarrow 266

- This method is not commented.

3.15 getServiceNameExdr: Line 268 \Rightarrow 270

- This method is not commented.

4 Other Issues

No logical potential issues found for this class.

5 Hours of work

The amount of working hours:

- Davide Cottica: 8 hours
- Stefano Badalucco: 8 hours