



POLITECNICO

MILANO 1863

PowerEnJoy Integration Test Plan Document

Erba Alessandro
Leveni Filippo
Lodi Luca

A.A 2016/2017

Contents

| | | |
|----------|--|-----------|
| 1 | IterateSectionWidget | 3 |
| 1.1 | Functional role | 3 |
| 1.2 | Application of the checklist | 3 |
| 2 | OfbizCatalogAltUrlTransform | 11 |
| 2.1 | Functional role | 11 |
| 2.2 | Application of the checklist | 11 |
| 3 | Effort Spent | 14 |

1 IterateSectionWidget

1.1 Functional role

This class extends “ModelScreenWidget” to provide a renderer for a widget as HTML.

Specifically it render a widget containing multiple “sections”, and iterate over those sections to render them, keeping count of pagination and position in the “page”.

It also inherit the “.accept(ModelWidgetVisitor visitor)” from the “ModelWidget” class, which implement the “visitor” pattern to traverse the tree of widgets and handle them different ways, according to their class type: a .accept(visitor) method in the visited class calls the .visit(this) in the visitor class, which should have a .visit(class-type) for each class-type to visit.

1.2 Application of the checklist

Class attributes

2. line 56

```
public class IterateSectionWidget extends
    ModelScreenWidget {
```

Class name should be nouns, in mixed case, with the first letter of each word in capitalized. ‘Iterate’ is a verb, thus class name should be changed to ‘SectionWidgetIterator’.

7. line 58

```
public static final String module =
    IterateSectionWidget.class.getName();
```

Constant declared without respecting UPPERCASE naming convention.

18. lines *

Lack of comment describing the purpose of the code.

Class constructor

13. lines 70, 87, 88, 95, 99, 101

line exceed recommended 80 characters length.

14. lines 87, 99

line exceed 120 characters length.

18. lines *

Lack of comment describing the purpose of the code.

23. The javadoc is not used as shows at this link:

<https://ci.apache.org/projects/ofbiz/site/javadocs/org/apache/ofbiz/widget/model/IterateSectionWidget>

25. line 55

```
@SuppressWarnings("serial")
```

Between class documentation comment and class statement must not be any code.

33. – line 77

```
String entryName = iterateSectionElement.  
    getAttribute("entry");
```

must be moved to the top

– line 82

```
String keyName = iterateSectionElement.  
    getAttribute("key");
```

must be moved to the top

– line 89

```
int viewSize = DEFAULT_PAGE_SIZE;
```

must be moved to the top

– line 90

```
String viewSizeStr = iterateSectionElement.  
    getAttribute("view-size");
```

must be moved to the top

– line 95

```
List<? extends Element> childElementList =  
    UtilXml.childElementList(iterateSectionElement  
    );
```

must be moved to the top

39. – line 95

```
List<? extends Element> childElementList =  
    UtilXml.childElementList(iterateSectionElement  
    );
```

must be declared using a constructor.

.getSectionList

18. lines *

Lack of comment describing the purpose of the code.

.renderWidgetString

12. line 223-227

```

    }
    /*
     * @return
     */
    public String getPaginateTarget (Map<String , Object>
        context) {

```

should separate sections of file though blank lines and optional comments.

13. lines 113, 122, 129, 134, 150-152, 157, 193, 197, 208, 210-211
line exceed recommended 80 characters length.

14. lines 113, 122
line exceed 120 characters length.

18. lines *
Lack of comment describing the purpose of the code.

27. The method is too long (90 lines), it should be reduced.

33. – line 132

```
List<?> theList = null;
```

must be moved to the top

- line 146

```
int startPageNumber = WidgetWorker.
    getPaginatorNumber ( context );
```

must be moved to the top

- line 171

```
Iterator<?> iter = theList.iterator();
```

must be moved to the top

- line 172

```
int itemIndex = -1;
```

must be moved to the top

- line 173

```
int iterateIndex = 0;
```

must be moved to the top

- line 179

```
Object item = iter.next();
```

must be moved to the top of the "while" block

39. – line 132

```
List<?> theList = null;
```

must be declared using a constructor.

40. line 128

```
if (obj == null)
```

must be replaced using

```
if (obj.equals(null))
```

42. line 129

```
Debug.logError(‘‘No object found for listName:’’+
    listNameExdr.toString(), module);
```

the description of the log could be more precise because in order to make more comprehensible the problem.

42. line 142.

```
Debug.logError(‘‘Object not list or map type’’,
    module);
```

the description of the log could be more precise because in order to make more comprehensible the problem.

42. line 218

```
Debug.logError(e, module);
```

the description of the log could be more precise because in order to make more comprehensible the problem.

.getPaginateTarget

18. lines *

Lack of comment describing the purpose of the code.

.getPaginate

13. lines 232-233

line exceed recommended 80 characters length.

.getViewSize

18. lines *
Lack of comment describing the purpose of the code.

.renderNextPrev

11. line 318-319
- ```
if (linkText.indexOf("?") < 0) linkText.append("?");
else linkText.append("&");
```

must be replaced using

```
if (linkText.indexOf("?") < 0) {
 linkText.append("?");
} else {
 linkText.append("&");
}
```

11. line 340-341
- ```
if (linkText.indexOf("?") < 0) linkText.append("?");
else linkText.append("&");
```

must be replaced using

```
if (linkText.indexOf("?") < 0) {
    linkText.append("?");
} else {
    linkText.append("&");
}
```

13. lines 243, 249, 250, 257, 300, 305-306, 308-309, 311, 320, 324, 327-329, 333-335, 345, 348-350

line exceed recommended 80 characters length.

14. lines 243, 324, 328, 333-334, 345
- line exceed 120 characters length.

18. lines *
Lack of comment describing the purpose of the code.

19. line 282-296

```
/*
int highIndex = -1;
try {
    highIndex = modelForm.getHighIndex();
} catch (Exception e) {
```

```

        highIndex = 0;
    }

    int lowIndex = -1;
    try {
        lowIndex = modelForm.getLowIndex();
    } catch (Exception e) {
        lowIndex = 0;
    }
    */

```

Code commented out without indication of the reason nor of the date by which remove the comment.

19. line 320

```

//if (queryString != null && !queryString.equals("
    null")) linkText += queryString + "&";

```

Code commented out without indication of the reason nor of the date by which remove the comment.

27. The method is too long (100 lines), it should be reduced.

33. – line 249

```

Map<String, Object> inputFields = UtilGenerics.
    checkMap(context.get("requestParameters"));

```

must be moved to the top

– line 250

```

Map<String, Object> queryStringMap = UtilGenerics
    .toMap(context.get("queryStringMap"));

```

must be moved to the top

– line 255

```

String queryString = UtilHttp.urlEncodeArgs(
    inputFields);

```

must be moved to the top

– line 256

```

int paginatorNumber = WidgetWorker.
    getPaginatorNumber(context);

```

must be moved to the top

– line 267

```

int viewIndex = -1;

```



```
must be moved to the top
- line 274
    int viewSize = -1;
must be moved to the top
- line 298
    int lowIndex = viewIndex * viewSize;
must be moved to the top
- line 299
    int highIndex = (viewIndex + 1) * viewSize;
must be moved to the top
- line 305
    HttpServletRequest request = (HttpServletRequest)
        context.get("request");
must be moved to the top
- line 306
    HttpServletResponse response = (
        HttpServletResponse) context.get("response");
must be moved to the top
- line 308
    ServletContext ctx = (ServletContext) request.
        getAttribute("servletContext");
must be moved to the top
- line 309
    RequestHandler rh = (RequestHandler) ctx.
        getAttribute("_REQUEST_HANDLER");
must be moved to the top
- line 317
    StringBuilder linkText = new StringBuilder(
        targetService);
must be moved to the top of the "if" block
- line 328
    String previous = UtilProperties.getMessage("
        CommonUiLabels", "CommonPrevious", (Locale)
        context.get("locale"));
```

must be moved to the top of the "if" block

– line 339

```
StringBuilder linkText = new StringBuilder(
    targetService);
```

must be moved to the top of the "if" block

– line 349

```
String next = UtilProperties.getMessage("
    CommonUiLabels", "CommonNext", (Locale)
    context.get("locale"));
```

must be moved to the top of the "if" block

40. line 245

```
if (targetService == null)
```

must be replaced using

```
if (targetService.equals(null))
```

42. line 263

```
Debug.LogWarning("TargetService is empty.", module
);
```

the description of the log could be more precise because in order to make more comprehensible the problem.

.accept

18. lines *

Lack of comment describing the purpose of the code.

.getListNameExdr

18. lines *

Lack of comment describing the purpose of the code.

.getEntryNameExdr

18. lines *

Lack of comment describing the purpose of the code.

.getKeyNameExdr

18. lines *

Lack of comment describing the purpose of the code.

.getPaginateTarget

- 18. lines *
Lack of comment describing the purpose of the code.

.getPaginate

- 18. lines *
Lack of comment describing the purpose of the code.

2 OfbizCatalogAltUrlTransform

2.1 Functional role

This class implements the “freemarker.template.TemplateTransformModel” interface, thus is used to transform part of the output of a freemarker template before serving the output to the user. In this specific case transforms URLs of catalog items into complete URLs according to the corrent `HttpServletRequest` and other environment variables.

It returns a writer through “`getWriter(final Writer out, final Map args)`” that according to enviroment variables and the argument passed in “args” write to writer “out” the input received, transforming incomplete catalog URLs into correct ones.

2.2 Application of the checklist

Class attributes

- 7. line 52

```
public final static String module =  
    OfbizCatalogUrlTransform.class.getName();
```

Constant declared without respecting UPPERCASE naming convention.

- 13. line 52

```
public final static String module =  
    OfbizCatalogUrlTransform.class.getName();
```

line exceed recommended 80 characters length.

- 18. lines *
Lack of comment describing the purpose of the code.

- 23. The javadoc is not used as shows at this link:
<https://ci.apache.org/projects/ofbiz/site/javadocs/org/apache/ofbiz/product/category/ftl/OfbizCatalogAltUrlTransform.html>

- 25. line 50
There is no class comment. It must be added.

.getStringArg

2. line 55-64

```
Object o = args.get(key);
if (o instanceof SimpleScalar) {
    return ((SimpleScalar) o).getAsString();
} else if (o instanceof StringModel) {
    return ((StringModel) o).getAsString();
} else if (o instanceof SimpleNumber) {
    return ((SimpleNumber) o).getAsNumber().toString();
} else if (o instanceof NumberModel) {
    return ((NumberModel) o).getAsNumber().toString();
}
```

single character name for non ‘throwaway’ variable ‘o’.

18. lines *
Lack of comment describing the purpose of the code.

.checkArg

2. line 55-64

```
Object o = args.get(key);
if (o instanceof SimpleScalar) {
    SimpleScalar s = (SimpleScalar) o;
    return "true".equalsIgnoreCase(s.getAsString());
}
```

single character name for non ‘throwaway’ variables ‘o’ and ‘s’.

18. lines *
Lack of comment describing the purpose of the code.

.getWriter

13. lines 91, 105-106, 116, 119, 121, 125, 131-132, 135-137, 139-141
line exceed recommended 80 characters length.
14. lines 119, 121, 135-137, 139-141
line exceed 120 characters length.
18. lines *
Lack of comment describing the purpose of the code.

various. line 143

```
out.write(url.toString());
```

url is declared as String and as the `.toString()` javadoc for String type suggests

String java.lang.String.toString()

This object (which is already a string!) is itself returned.

3 Effort Spent

- Alessandro Erba $\approx 18\text{h}$
- Filippo Leveni $\approx 18\text{h}$
- Luca Lodi $\approx 18\text{h}$