Software Engineering 2 Glassfish

Inspection document Version 1.0

1/3/2016

Politecnico di Milano A.A. 2015-2016

Milica Jovanovic (mat. 835953);Pavle Vidanovic (mat. 854472)

Contents

1 Assigned class

Our team has been assigned to do the code inspection on class WebappClassLoader. It belongs to the package org.glassfish.web.loader
Functions to be covered are:

- findClassInternal(String name)
- findResourceInternal(String name, String path)
- findResourceInternalFromRepositories(String name, String path)

2 Functional role of assigned class

WebappClassLoader

***findClassInternal(String name)* –** find specified class in local repositories. The function requires a class path as a parameter.

First we check if the class path is invalid (if the path is null or starts with "java."), if so the appropriate exception is thrown. If the path is valid we build absolute path to the compiled version of the java class. Next, we call findResourceInternal(String,String) to find specified ResourceEntry object. If the entry is not found the ClassNotFound exception is thrown, else search for loadedClass concurrently from the entry. If the extracted class is not null we return it as result of function execution, on contrary we check if binary content is not null, we go up to package where class is defined. First we extract package name, then if package is not defined we create it with given package name if manifest of entry is empty, otherwise we create it with additional attributes.

After that provide seal check which means that if securityManger is used then all packages must be sealed. If that is not a case we just return found entry.

***findResourceInternal(String name, String path)* –** find specified resource in local repository. Function requires resource name and resource path as a parameter.

First we check if particular component is starter (if the ".class" file exists), appropriate exception is thrown. After that we check if the function parameters are valid. Next, we check if the resource entry exists in the concurrent hash map, if yes we return it as a result of the function, if not we check if the resource was searched before by checking notFoundResources hash map by his key value (resource name). If neither assumption is true we start searching for the resource in other repositories (findResourceInternalFromRepositories), if resource is not found in repositories, we search it in jar files concurrently (findResourceInternalFromJars). If resource is still not found we put it in notFoundResources hash map and return null as result. Otherwise if the resource was found in repo or jar we first update resourceEntries hash map and then return resource as a hash map.

***findResourceInternalFromRepositories(String name, String path)* –** tries to find and load specified resource from internal repositories of the class. Function requires resource name and resource path as a parameter.

First a check is done if repositories array exists (not null), then we go through repositories array to find the specified resource.

In each iteration we build full path to the file by concatenating the repository path to the relative path of the searched resource, then we check if the resource exists in the DirContext table using the previously created full path. After that we build the resource entry in a privileged way

(AccessController.doPriviledged) or using findResourceInternal(File file,String path) function which builds the ResourceEntry object by setting its source and codeBase attributes. After setting attributes of ResourceEntry object we extract contentLength and lastModified from the resource found in DirContext. lastModified attribute of ResourceEntry is set and we extract binary stream from the found Resource object. In a concurrent way with all permissions we modify two arrays (lastModifiedDates and paths) by adding the entry's lastModified attribute and the fullPath to the particular resource to the arrays, respectively.

After the for loop finishes, we check the reason for its breaking i.e if the resource entry was found or we have search in all repositories and have not found the specified resource (entry!=null). If the entry was found we read the binaryContent from the InputStream parameter of readEntryData function and set the binaryContent attribute of the ResourceEntry.

In the end of the function the build ResourceEntry object is returned.

3 List of issues found

3.1 Method 1 – findClassInternal(String name)

Naming conventions			
Checklist point	Issue	Line	Details
1	Parameter: name	2727	parameter *name* does not give enough information about what it represents. (name -> className)
1	Variable: entry	2736	Does not give enough information (entry -> resourceEntry)
1	Variable: clazz	2742	Inappropriate name (clazz -> classEntry)
1	Variable: pkg	2756	Does not give enough information (pkg -> packageObject)
1	Function: validate	3279	Function is not specific enough
3	Class name: WebappClassLoader	149	First letter of each word sould be capitalized (class WebappClassLoader -> class Web**A**ppClassLoader)
5	Method name: modified()	1036	Method names should be verbs, and modified is not a verb.
5	Method name: *nullInstance*	2237	Method names should be verbs, or should start with a verb.
5	Method name: *loadedByThisOrChild*	2457	Method names should be verbs, or should start with a verb.

Indention

Checklist point	Issue	Line	Details
8	More spaces used for indention	2739	There are more than 7 spaces for indention. It should be reduced to 4.
8	Less spaces used for indention	2761 and 2775	There are just 2 spaces used for indention. It should be 3 or 4.
8	Block in code is not indent	2762-2774	This code should be indented by 3 or 4 spaces.

Braces			
Checklist point	Issue	Line	Details
10	Kernighan and Ritchie style is used	In whole class	First brace is on the same line of the instruction that opens the new block
11	If-statement has one statement	2730	Correct way is that the statement is surrounded by curly brackets
11	If-statement has one statement	2738	Correct way is that the statement is surrounded by curly brackets
11	If-statement has one statement	2743	Correct way is that the statement is surrounded by curly brackets
11	If-statement has one statement	2754	Correct way is that the statement is surrounded by curly brackets
11	If-statement has one statement	2791	Correct way is that the statement is surrounded by curly brackets

File organization			
Checklist point	Issue	Line	Details
12	Blank lines that don't separate sections		There are blank lines that don't separate sections, those are probably present to separate things conceptually different.

			However this requirement is respected which means that sections are correctly separated by blank lines in the entire class.

Wrapping lines			
Checklist point	Issue	Line	Details
15	Line break is not correct	2788	Line is separated before an operator “II”
15	Line break is not correct	2793	Line is separated before an operator “+”
16	High level break is present in this function	2792-2794	Break doesn’t occur after coma or an operator.

Class and interface declaration			
Checklist point	Issue	Line	Details
25	Class implementation comment	112-148	Class implementation comment is in wrong position, it should be after class statement.
25	Wrong order of class static variables	157, 160, 167	In class WebAppClassLoader there are first declared private static variables instead of public ones.
25	Wrong order of instance variables	352-541	Instance variables are declared in wrong order. Order should be: public, protected, package level and private as last. This is not the case in this class.

Initialization and Declarations			
Checklist point	Issue	Line	Details

33	Declarations at the beginning of block	2733, 2734, 2736, 2752, 2753 and 2757	Declarations at those lines are not on the beginning of the block.

Method Calls			
Checklist point	Issue	Line	Details
36	Method returned values are not used properly	2768 and 2771	Should be pkg = definePackage(packageName, null, null, null, null, null, null, null);

Computation, Comparisons and Assignments			
Checklist point	Issue	Line	Details
44	Brutish programming present	2779-2796	Brutish programing is presented in a way that there are too many if checks.
45	Operator precedence problems	2788-2789	Parenthesis are used to avoid operator precedence problems.

Exceptions			
Checklist point	Issue	Line	Details
53	Exception is not caught	2792	SecurityException is thrown and is not properly handled in the function, and it is neither stated in the function declaration.

3.2 Method 2 - findResourceInternal(String name, String path)

Naming conventions			
Checklist point	Issue	Line	Details
1	Parameter: name	2825	parameter *name* does not give enough information about what it represents. (parameter name -> resourceName)
1	Variable: started	2826	Does not give enough information (started -> resourceStarted)
1	Variable: entry	2835	Inappropriate name (entry -> resourceEntry)

Indention			
Checklist point	Issue	Line	Details
8	More spaces used for indention	2827	Line has 4 spaces more that it should have

Braces			
Checklist point	Issue	Line	Details
10	Kernighan and Ritchie style is used	In whole class	First brace is on the same line of the instruction that opens the new block

File organization			
Checklist point	Issue	Line	Details
12	Blank lines that don't separate sections		There are blank lines that don't separate sections, those are probably present to separate things conceptually different.

			However this requirement is respected which means that sections are correctly separated by blank lines in the entire class.

Wrapping lines			
Checklist point	Issue	Line	Details
15	Line break is not correct	2827	Line wrap is in front of open bracket which is not by the proposed rules, also the length of the line is not greater than 80 characters

Comments			
Checklist point	Issue	Line	Details
19	Comments do not contain dates	2855-2858	Comment from line 2855-2858 do not contain date when it should be removed.

Class and interface declaration			
Checklist point	Issue	Line	Details
25	Class implementation comment	112-148	Class implementation comment is in wrong position, it should be after class statement.
25	Wrong order of class static variables	157, 160, 167	In class WebAppClassLoader there are first declared private static variables instead of public ones.
25	Wrong order of instance variables	352-541	Instance variables are declared in wrong order. Order should be: public, protected, package level and private as last. This is not the case in this class.

Initialization and Declarations			
Checklist point	Issue	Line	Details
33	Declarations at the beginning of block	2835 and 2859	Line 2835 – *entry* variable is not declared in the beginning of block of code Line 2859 – *entry2* variable is not declared in the beginning of block of code

Methods Calls			
Checklist point	Issue	Line	Details
35	Wrong function call	2851	function putIfAbsent(name,name) should be used instead of put(name,name) because previously we checked that this key does not exist in the hash map

Output Format			
Checklist point	Issue	Line	Details
42	Error message not descriptive enough	2827	throws exception that the web container has not yet been started but the message displayed will be “AS-WEB-UTIL-00010” which does not give information how to solve the issue

Computation, Comparisons and Assignments			
Checklist point	Issue	Line	Details

44	Brutish programming present	2836-2862	Brutish programing is presented in a way that there are too many if checks.
45	Operator precedence problems	2831	Parenthesis are used to avoid operator precedence problems.

3.3 Method 3 - findResourceInternalFromRepositories(String name, String path)

Naming conventions			
Checklist point	Issue	Line	Details
1	Parameter: name	2874	parameter *name* does not give enough information about what it represents. (name -> resourceName)
1	Parameter: path	2875	Does not give enough information (path -> resourcePath)
1	Variable: repositories	399	Inappropriate name (repositories-> repositoriesPath)
1	Variable: dp	2900	Does not give enough information (dp -> priviledgedAction)
1	Variable: result2	2926	Does not give enough information (result2 -> lastModifiedDatesExtended)
1	Variable: result	2934	Does not give enough information (result -> pathsExtended)

Braces			
Checklist point	Issue	Line	Details
10	Kernighan and Ritchie style is used	In whole class	First brace is on the same line of the instruction that opens the new block

File organization			
Checklist point	Issue	Line	Details

12	Blank lines that don't separate sections		There are blank lines that don't separate sections, those are probably present to separate things conceptually different. However this requirement is respected which means that sections are correctly separated by blank lines in the entire class.

Comments			
Checklist point	Issue	Line	Details
18	Comment not descriptive enough	2897	Comment of the fn is well written, but comment on line 2897 is and explained because it doesn't tell what the code does instead it just inform us that if exception is thrown the resource was found
19	Comments do not contain dates		Comments are not written in the proposed way

Class and interface declaration			
Checklist point	Issue	Line	Details
25	Class implementation comment	112-148	Class implementation comment is in wrong position, it should be after class statement.
25	Wrong order of class static variables	157, 160, 167	In class WebAppClassLoader there are first declared private static variables instead of public ones.
25	Wrong order of instance variables	352-541	Instance variables are declared in wrong order. Order should be: public, protected, package level and private as last. This is not the

			case in this class.

Initialization and Declarations			
Checklist point	Issue	Line	Details
32	Variable not initialized	2924	Variable j is declared as int but not initialized on the spot
33	Declarations at the beginning of block	2881 to 2885, 2907 and 2934	Declarations at those lines are not on the beginning of the block.

Arrays			
Checklist point	Issue	Line	Details
38	Array variable might go out of bound	2901	For loop depends on the length of repositories, and in the for block array variable files is accessed through counter i. Variable files[i], counter i is connected to variable repositories[i], if the length of files is less than length of repositories files will go out of bound

Exceptions			
Checklist point	Issue	Line	Details
53	Exception is not properly handled	2943	Catch block is empty, exception is not handled

4 Other problems

There is an error in the line 2912 of the third function findResourceInternalFromRepositories. Issue is related to the position of ***if*** statement:

```
ResourceAttributes attributes =
        (ResourceAttributes) resources.getAttributes(fullPath);
```

```
contentLength = (int) attributes.getContentLength();
entry.lastModified = attributes.getLastModified();

if (resource != null) {

   try {
      binaryStream = resource.streamContent();
   } catch (IOException e) {
      return null;
   }
...
…
```

Code written this way could give NullException because we use the resource instance before checking if the instance is set (resource!=null). It should be written in following way:

```
if (resource != null) {
   ResourceAttributes attributes =
      (ResourceAttributes) resources.getAttributes(fullPath);
   contentLength = (int) attributes.getContentLength();
   entry.lastModified = attributes.getLastModified();

      try {
         binaryStream = resource.streamContent();
      } catch (IOException e) {
         return null;
      }
   ...
   …
```