**findResourceInternal(String name, String path)**

Naming Conventions

1. Line 2825 – parameter name -> resourceName

Line 2826 – parameter started -> resourceStarted

Line 2835 – parameter entry -> resourceEntry

1. No issues
2. No issues
3. No issues
4. No issues
5. No issues
6. No issues

Indention

1. Line 2827 – has 4 spaces more that it should have
2. No tabs used

Braces

1. Kernighan and Ritchie style is used
2. No issues

File Organization

1. 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.
2. No issues
3. No issues

Wrapping Lines

1. Line 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

Should be like this:

throw new IllegalStateException(getString(NOT\_STARTED, name));

instead of

throw new IllegalStateException(

getString(NOT\_STARTED, name));

1. No issues
2. No issues

Comments

1. Class and constructors are very good commented as well as the function findResourceInternal. Code inside this function is very not good commented.
2. Comment from line 2855-2858 do not contain date when it should be removed.

Java Source Files

1. No issues
2. No issues
3. ?
4. ?

Package and Import Statements

1. No issues

Class and Interface Declerations

* 1. Class/interface documentation comment – No issues
  2. Class or interface statement – No issues
  3. Class/interface implementation comment – No issues
  4. Class (static) variables - In class WebappClassLoader there are first declared private static variables instead of public ones. Lines 157, 161.
  5. Instance variables - Instance variables are declared in wrong order in class WebappClassLoader, first the private instance variables then protected and after that in mixed order starting from line 352
  6. Constructors – No issues
  7. Methods – No issues

1. Methods are properly grouped by functionality
2. No issues

Initialization and Declaration

1. The variables and class members are properly defined and have correct visibility in a way that all of private types has getters and setters implemented for them.
2. No issue
3. No issue
4. No issues
5. No issues
6. 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

1. No issues
2. Line 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
3. No issues

Arrays

1. No issues
2. No issues
3. No issues

Object Comparison

1. No issues

Output Format

1. No issues
2. @LogMessageInfo( message = "This web container has not yet been started",level="WARNING")

public static final String NOT\_STARTED = "AS-WEB-UTIL-00010";

Line 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

1. No issues

Computation, Comparisons and Assignments

1. ? This all code I think is brute forcing especially the code in the end

ResourceEntry entry = resourceEntries.get(name);

if (entry != null) {

return entry;

} else if (notFoundResources.containsKey(name)) {

return null;

}

entry = findResourceInternalFromRepositories(name, path);

if (entry == null) {

synchronized (jarFiles) {

entry = findResourceInternalFromJars(name, path);

}

}

if (entry == null) {

notFoundResources.put(name, name);

return null;

}

// Add the entry in the local resource repository

// Ensures that all the threads which may be in a race to load

// a particular class all end up with the same ResourceEntry

// instance

ResourceEntry entry2 = resourceEntries.putIfAbsent(name, entry);

if (entry2 != null) {

entry = entry2;

}

1. ?
2. No issues
3. No issues
4. No issues
5. No issues
6. No issues
7. No issues

Exceptions

1. No issues
2. No issues

Flow of Control

1. No issues
2. No issues
3. No issues

Files

1. No issues
2. No issues
3. No issues
4. No issues