­



Politecnico di Milano

A.A. 2015-2016

Software Engineering 2: “MyTaxi”

Code inspection

Manzi Giuseppe (mat. 854470) &

Nicolini Alessandro (mat. 858858)

CONTENTS

[1. Assigned classes 4](#_Toc439520413)

[2. Functional role 4](#_Toc439520414)

[3. Found issues (from checklist) 5](#_Toc439520415)

[3.1 Section 0 5](#_Toc439520416)

[3.2 Section 1 5](#_Toc439520417)

[3.3 Section 2 6](#_Toc439520418)

[3.3 Section 3 6](#_Toc439520419)

[4. Other problems 7](#_Toc439520420)

# 1. Assigned classes

All the code that was assigned to us belongs to the same class, that is the class **ExtensionValidator**, located in:

**appserver/web/web-core/src/main/java/org/apache/catalina/util/ExtensionValidator.java**

The code lines we had to check are the following ones:

* **Section 1** (lines **169 to 190**): *if* statement;
* **Section 2** (lines **215 to 292**): *validateApplication* method;
* **Section 3** (lines **317 to 385**): *validateManifestResources* method.

# 2. Functional role

## 2.1 Functional role of the whole class

The class has to check the resources extensions, to ensures that all the needed resources are available.



The class uses two lists that contains the manifest resources, from which we can extract the extensions of the needed resources, and the available extensions. These lists are declared as attributes in the following code:



## 2.2 Functional role of the sections

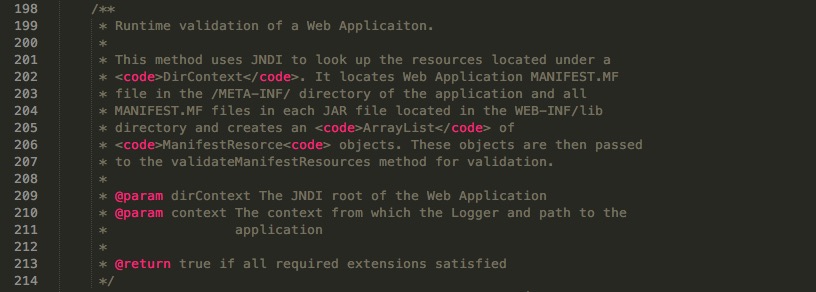
**2.2.1 Section 1**

*Section 1* is part of the static initializer. It contains an *if* statement that get the files of the extension directory. For every directory listed in system properties it checks if it actually is a directory and add the manifest resources of the contained jar files in *containerManifestResources* using *addSystemResource* private static method.



**2.2.2 Section 2**

*Section 2* is the *ValidateApplication* method. It validates at runtime the application and it returns true if all the required extensions are satisfied.



**2.2.3 Section 3**

*Section 3* is the *ValidateManifestResources* method. Using two nested while, it checks if, for every manifest resource, it exists an available resource that has a compatible extension.



# 3. Found issues (from checklist)

## 3.1 Section 0

**3.1.1 Description**

Attributes and imports.

**3.1.2 Lines**

From 59 to 128.

**3.1.3 Location**

appserver/web/web-core/src/main/java/org/apache/catalina/util/ExtensionValidator.java

**3.1.4 Issues**

#### 3.1.4.1

* **Lines:** 103 to 104
* **Rule (Kind of rule):** 7 (Naming conventions)
* **Description:** Static and final attributes should be all capitals
* **Code:**

****

## 3.2 Section 1

**3.2.1 Description**

if(extensionsDir != null) {---}

**3.2.2 Lines**

From 169 to 190.

**3.2.3 Location**

appserver/web/web-core/src/main/java/org/apache/catalina/util/ExtensionValidator.java

**3.2.4 Issues**

#### 3.2.4.1

* **Lines:** 170 to 171
* **Rule (Kind of rule):** 15 (Line Break)
* **Description:** The break is before an operator (=) instead of being after it.
* **Code:**

****

## 3.3 Section 2

**3.3.1 Description**

ValidateApplication(DirContext dirContext, StandardContext context) method.

**3.3.2 Lines**

From 215 to 292.

**3.3.3 Location**

appserver/web/web-core/src/main/java/org/apache/catalina/util/ExtensionValidator.java

**3.3.4 Issues**

#### 3.3.4.1

* **Lines:** 215 to 216
* **Rule (Kind of rule):** 15 (Line Break)
* **Description:** Break after an open parenthesis.
* **Code:**

****

#### 3.3.4.2

* **Lines:** 226
* **Rule (Kind of rule):** 11 (Braces)
* **Description:** If without braces
* **Code:**

****

#### 3.3.4.3

* **Lines:** 230
* **Rule (Kind of rule):** 1 (Name)
* **Description:** Meaningless name
* **Code:**

****

#### 3.3.4.4

* **Lines:** 240
* **Rule (Kind of rule):** 18 (Comments)
* **Description:** Useless Comment
* **Code:**

****

#### 3.3.4.5

* **Lines:** 242
* **Rule (Kind of rule):** 1 (Name)
* **Description:** Meaningless name (mre -> manifestResource)
* **Code:**

****

#### 3.3.4.6

* **Lines:** 242 to 243
* **Rule (Kind of rule):** 15 (Line Break)
* **Description:** The code should be on the same line.
* **Code:**

****

#### 3.3.4.7

* **Lines:** 262
* **Rule (Kind of rule):** 1 (Name)
* **Description:** Meaningless name (ne-> namingEnumeration)
* **Code:**

****

#### 3.3.4.8

* **Lines:** 277
* **Rule (Kind of rule):** 1 (Name)
* **Description:** Meaningless name
* **Code:**

****

#### 3.3.4.9

* **Lines:** 279
* **Rule (Kind of rule):** 15 (Line Break)
* **Description:** The code should be on the same line
* **Code:**

****

## 3.4 Section 3

**3.4.1 Description**

ValidateManifestResources(String appName, ArrayList<ManifestResource> resources) method.

**3.4.2 Lines**

From 317 to 385.

**3.4.3 Location**

appserver/web/web-core/src/main/java/org/apache/catalina/util/ExtensionValidator.java

**3.4.4 Issues**

#### 3.4.4.1

* **Lines:** 325
* **Rule (Kind of rule):** 1 (Name)
* **Description:** Meaningless name (mre -> manifestResource)
* **Code:**

****

#### 3.4.4.2

* **Lines:** 349 to 350
* **Rule (Kind of rule):** 15 (Wrapping lines)
* **Description:** The break is before an operator (&&) instead of being after it.
* **Code:**



#### 3.4.4.3

* **Lines:** 351 to 352
* **Rule (Kind of rule):** 15 (Wrapping lines)
* **Description:** The break is after a casting instead of being between the operator (=) and the casting.
* **Code:**



#### 3.4.4.4

* **Lines:** 357 to 358
* **Rule (Kind of rule):** 15 (Wrapping lines)
* **Description:** The break is before an operator (&&) instead of being after it.
* **Code:**



# 4. Other problems

## 4.1 Section 2

#### 4.1.1

* **Lines:** 240
* **Description:** Initialization should be done using binding.getname()
* **Code:**

****

## 4.1 Section 2

#### 4.1.1

* **Lines:** 378 to
* **Description:** It is possible to combine the if statements
* **Code:**

****