JPass Category Partition

Verificação e Validação de Software 2020–2021

Manuel Tomás 51054
Tiago Varela 51017

2021/03/08

Contents

Selected Functions

The documentation of all the functions on each class on the package $jpass.util$ was analysed. As a result, these functions were selected for being one of the few that had documentation. Most functions simply do not have the necessary documentation to perform Black-Box testing. $SpringUtils$ utilises $SpringLayout$ which is unfamiliar to us, and so was not chosen. $setClipboardContent$, $getClipboardContent$ and $clearClipboardContent$ are all relatively simple functions, but they all perform different tasks, with either different inputs or outputs. For this reason, they were chosen. As we were unfamiliar with $Spring$ and the only other classes that had documented functions were $CriptUtils$ and $StringUtils$, we decided to select one of each. With this in mind, the five methods selected were:

- ClipboardContent class
  - public static void setClipboardContent(String str)
  - public static void clearClipboardContent()
  - public static String getClipboardContent()
- CriptUtils class
  - public static byte[] getSha256Hash(char[] text)
- StringUtils class
  - public static String stripNonValidXMLCharacters(String in)

setClipboardContent(String str)

Purpose

This function takes the System Clipboard and sets its content to be the given str String. A particular case is that empty Strings cannot be set, as according to the Clipboard API.

Category Partition Algorithm

Parameters

- String str
- System Clipboard is empty or not.
- System Clipboard is accessible or not.

Characteristics

- String str - String with N number of char Y (N: number from 0 to +infinity; Y: any char from ASCII table)

Restrictions

- Null str values have no specified behaviour, but should not run. It does not matter what the other inputs are.
- If system clipboard is not accessible, other inputs do not matter.

Categories:

- str - null, empty or !empty.
- isEmptyClipboard or !isEmptyClipboard
- isAccessible or !isAccessible

Combinations

str	isEmptyClipboard	isAccessible	Expected
null	-	-	undefined
-	-	!isAccessible	exception
empty	isEmptyClipboard	isAccessible	clipboard.isEmpty()
empty	!isEmptyClipboard	isAccessible	clipboard.isEmpty()
!empty	isEmptyClipboard	isAccessible	clipboard.equals(str)
!empty	!isEmptyClipboard	isAccessible	clipboard.equals(str)

A total of 6 test cases.

Unit tests

shouldNotSetClipboardExceptionTest()	
What is being done?	Attempts to clear the content from the System Clipboard, given the Clipboard is busy.
What is being tested?	Tests if the correct exception is thrown in the expected exceptional case.
Result	Failed[1]

shouldSetClipboardTest()	
What is being done?	Attempts to set the System Clipboard text content to a given string setClipboard, given the System Clipboard content is empty.
What is being tested?	Tests if the System Clipboard text content matches the set string setClipboard.
Result	Pass

shouldSetClipboardEmptyStringTest()	
What is being done?	Attempts to set the System Clipboard text content to an empty string, given the System Clipboard content is empty.
What is being tested?	Tests if the System Clipboard remains empty upon receiving an empty string.
Result	Pass

shouldSetClipboardOverwriteTest()	
What is being done?	Attempts to set the System Clipboard text content to a given string setClipboard, overwriting the previous content.
What is being tested?	Tests if the System Clipboard text content matches the set string setClipboard.
Result	Pass

shouldSetClipboardEmptyStringOverwriteTest()	
What is being done?	Attempts to set the System Clipboard text content to an empty string, overwriting the previous content.
What is being tested?	Tests if the System Clipboard is now empty upon receiving an empty string.
Result	Pass

[1]The Exception tests all fail because we do not know how to enforce the necessary state on the clipboard so that the exception would be triggered. The clipboard should be "busy" when the functions are called, thus throwing the exception, but we can't set this state.

clearClipboardContent()

Purpose

This function takes the System Clipboard and clears its contents. The System Clipboard should have no content afterwards.

Category Partition Algorithm

Parameters

- System Clipboard is empty or not.
- System Clipboard is accessible or not.

Characteristics

There are no important characteristics to be defined.

Restrictions

- If system clipboard is not accessible, other inputs do not matter.
- In theory, if Clipboard is empty, this function is not necessary and whether or not it does anything the output will be the same.

Categories:

- isEmptyClipboard or !isEmptyClipboard
- isAccessible or !isAccessible

Combinations

isEmptyClipboard	**isAccessible**	**Expected**
-	!isAccessible	exception
isEmptyClipboard	isAccessible	Probably not necessary
!isEmptyClipboard	isAccessible	clipboard.isEmpty()

A total of 2 or 3 test cases.

Unit tests

shouldNotClearClipboardExceptionTest()	
What is being done?	Attempts to clear the content from the System Clipboard, given the Clipboard is busy.
What is being tested?	Tests if the correct exception is thrown in the expected exceptional case.
Result	Failed[1]

shouldClearClipboardTest()	
What is being done?	Attempts to clear the System Clipboard of all content.
What is being tested?	Tests if the System Clipboard is truly empty.
Result	Pass

getClipboardContent()

Purpose

This function returns the text from the System Clipboard's content. If there is no text content, it returns null.

Category Partition Algorithm

Parameters

- System Clipboard is empty or not.
- System Clipboard is accessible or not.

Characteristics

There are no important characteristics to be defined.

Restrictions

- If system clipboard is not accessible, other inputs do not matter.

Categories:

- isEmptyClipboard or !isEmptyClipboard
- isAccessible or !isAccessible

[1]The Exception tests all fail because we do not know how to enforce the necessary state on the clipboard so that the exception would be triggered. The clipboard should be "busy" when the functions are called, thus throwing the exception, but we can't set this state.

Combinations

isEmptyClipboard	isAccessible	Expected
-	!isAccessible	exception
isEmptyClipboard	isAccessible	output == null
!isEmptyClipboard	isAccessible	clipboard.equals(output)

A total of 6 test cases.

Unit tests

shouldNotGetClipboardExceptionTest()	
What is being done?	Attempts to clear the content from the System Clipboard, given the Clipboard is busy.
What is being tested?	Tests if the correct exception is thrown in the expected exceptional case.
Result	Failed[1]

shouldGetClipboardEmptyStringTest()	
What is being done?	Attempts to get the text contents of the System Clipboard when it has no content. getClipboardContent should return null in this case.
What is being tested?	Tests if null is truly the returned value.
Result	Pass

shouldGetClipboardTest()	
What is being done?	Attempts to get the text contents of the System Clipboard.
What is being tested?	Tests if the returned value is the expected clipboard content.
Result	Pass

getSha256Hash(char[] text)

Purpose

This method calculates the SHA-256 hash of a given char[].

Category Partition Algorithm

Parameters

- text

[1]The Exception tests all fail because we do not know how to enforce the necessary state on the clipboard so that the exception would be triggered. The clipboard should be "busy" when the functions are called, thus throwing the exception, but we can't set this state.

Characteristics

- text - null, empty, N number of Y chars (N: number from 1 to +infinity; Y: any char from ASCII table)

Restrictions

There are no special restrictions defined by the method
Categories:

- text - null, empty, !empty

Combinations

text	**Expected**
null	exception
empty	hash(empty) [2]
!empty	hash(text) [2]

A total of 3 test cases.

Unit tests

shouldNotGetSha256HashNullTest()	
What is being done?	Attempts to get the SHA-256 Hash of a null object which should be an exception as it is an invalid operation.
What is being tested?	Tests if the exception is thrown in the expected exceptional case.
Result	Pass

shouldGetSha256HashEmptyTest()	
What is being done?	Attempts to get the SHA-256 of an empty string which should return its hash value.
What is being tested?	Tests if the right value for the hash of empty string is returned.
Result	Pass

shouldGetSha256HashNotEmptyTest()	
What is being done?	Attempts to get the SHA-256 of a string which should return the expected value.
What is being tested?	Tests if the right value for the hash of a string is returned.
Result	Pass

[2]The output of the combinations were tested using a SHA-256 hash calculator tool. [1]

stripNonValidXMLCharacters(String in)

Purpose

This method ensures that the output String has only valid XML unicode characters.

Category Partition Algorithm

Parameters

- in

Characteristics

- in -> null, empty, N number of Y chars (N: number from 1 to +infinity; Y: any char)

Restrictions

There are no special restrictions defined by the method.
Categories:

- text - null, empty, (!empty && !containsInvalid) or (!empty && containsInvalid)

Combinations

text	Expected
null	empty
empty	empty
!empty && !containsInvalid	string without invalid xml characters
!empty && containsInvalid	string without invalid xml characters

A total of 4 test cases.

Unit tests

shouldStripNullTest()	
What is being done?	Attempts to strip a null object which, according to the method documentation, should return an empty string.
What is being tested?	Tests if the returned value really is empty string.
Result	Failed - The function documentation refers that it returns an empty string with a null input, but in reality is returns null. This means either the documentation or the implementation have an error that needs to be fixed.

shouldStripEmptyTest()	
What is being done?	Attempts to strip a empty string which, according to the method documentation, should return an empty string.
What is being tested?	Tests if the returned value really is empty string.
Result	Pass

shouldStripNotEmptyHasNotInvalidTest()	
What is being done?	Attempts to strip a string without invalid xml characters which should return a string without invalid xml characters.
What is being tested?	Tests if the the string returned really has no invalid xml characters.
Result	Pass

shouldStripNotEmptyHasInvalidTest()	
What is being done?	Attempts to strip a string with invalid xml characters which should return a string without invalid xml characters.
What is being tested?	Tests if the the string returned really has no invalid xml characters.
Result	Pass

References

[1] xorbin. Sha-256 hash calculator. [Online]. Available: https://xorbin.com/tools/sha256-hash-calculator