Does the String or implementation of CharSequence Interface have length or not.

CharSequence is a readable sequence of char values. This interface provides read only access to many different kinds of char sequences.

Methods include charAt(int index), length(), subsequence(int start, int end), toString().

String, StringBuffer, StringBuilder, CharBuffer and Segment are class implementing CharSequence Interface.

Now hasLength(..) method is overloaded.

hasLength(CharSequence str) and hasLength(String str).

Below is the code for hasLength(CharSequence str)

**public** **static** **boolean** hasLength(CharSequence str) {

**return** (str != **null** && str.length() > 0);

}

It checks for 2 conditions and both needs to be true.

First the input parameter must not be null

Second the input parameter’s length must be greater than 0.

If both the conditions are true then true is returned

Either of condition is false then false is returned.

Next, the code for hasLength(String str)

**public** **static** **boolean** hasLength(String str) {

**return** *hasLength*((CharSequence) str);

}

If the input parameter is String then it is cast into CharSequence and the overloaded version is called.

Below is the code for testing for length for any implementation of CharSequence.

**package** stringutils;

**import** org.springframework.util.StringUtils;

**public** **class** SpringStringUtilsDemo {

**public** **static** **void** main(String[] args) {

String str = **null**;

*hasLengthDemo*(str);

str = "";

*hasLengthDemo*(str);

str = "Stark";

*hasLengthDemo*(str);

*hasLengthDemo*(**new** StringBuffer(""));

*hasLengthDemo*(**new** StringBuffer("Text"));

}

**public** **static** **void** hasLengthDemo(CharSequence str) {

**boolean** hasLength = StringUtils.*hasLength*(str);

System.***out***.println("str=" + str + " has length? " + hasLength);

}

}

Output:

str=null has length? false

str= has length? false

str=Stark has length? true

str= has length? false

str=Text has length? true