

How are we doing? Please help us improve Stack Overflow. [Take our short survey](#)

# How do I check for null or empty string for many arguments? - C#

Asked 3 years, 7 months ago   Active 3 years, 1 month ago   Viewed 2k times

I have the below method, which I need to check for whether the arguments are empty or null.

3

```
public DB Where(string field, string operat, string value, string andOr, string
field2, string operat2, string value2)
{
    _Where = " WHERE " + field + " " + operat + " @" + field + "1 " + andOr + " " +
field2 + " " + operat2 + " @" + field2 + "2 ";
    _Params.Add(field + "1", value);
    _Params.Add(field2 + "2", value2);
    return this;
}
```

I have found the `string.IsNullOrEmpty` method however this would require this much code:

```
if (string.IsNullOrEmpty(field))
    throw new ArgumentException("field Cannot be null or be empty");

if (string.IsNullOrEmpty(operat))
    throw new ArgumentException("operat Cannot be null or be empty");

if (string.IsNullOrEmpty(value))
    throw new ArgumentException("value Cannot be null or be empty");

if (string.IsNullOrEmpty(andOr))
    throw new ArgumentException("andOr Cannot be null or be empty");

if (string.IsNullOrEmpty(field2))
    throw new ArgumentException("field2 Cannot be null or be empty");

if (string.IsNullOrEmpty(operat2))
    throw new ArgumentException("operat2 Cannot be null or be empty");

if (string.IsNullOrEmpty(value2))
```

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

is there any way or shortening this?

Also, I have tried creating a custom method for this task, however it throws an exception in the custom method instead of the Where() method which makes it tricky to debug.

c# string null isnulloreempty

edited Mar 5 '16 at 17:23

asked Mar 5 '16 at 16:59



Chris

102 13

You could create a static method: `ValidateParameterNotEmpty(string name, string value)` . You would reduce the number of lines to 1/3 – [xanatos](#) Mar 5 '16 at 17:06

The validation you will need to do to prevent SQL injection will be a lot more work so I would make this second priority. – [Crowcoder](#) Mar 5 '16 at 17:07

2 If the field is null or empty, nothing will appear when throwing `ArgumentException` your way. Did you mean to use [nameof](#) ? – [Gediminas Masaitis](#) Mar 5 '16 at 17:07 ✎

I did actually try something like that but found when it threw the exception the debugger went to the method `ValidateParameterNotEmpty()` instead of the `Where()` method. Of course, if I use that method in multiple places it could make it tricky to debug. @xanatos – [Chris](#) Mar 5 '16 at 17:10 ✎

@GediminasMasaitis yes sorry that's a mistake. Although I don't believe I can use `nameof()` as I am not using `c#6`. I will amend it to what I now have. – [Chris](#) Mar 5 '16 at 17:19

## 4 Answers



You could check the value one by one or creating intermediate function to do that.

2

Alternatively, my suggestion is: you could put all the inputs in an array and use LINQ Any to check all of them at once:



```
public DB Where(string field, string operat, string value, string andOr, string field2,
string operat2, string value2)
{
    string[] inputs = {field, operat, value, andOr, field2, operat2, value2}
    if (inputs.Any(x => string.IsNullOrEmpty(x)))
```



By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

}

answered Mar 5 '16 at 17:24



lan

26k 15 52 81

What I can suggest is this:

1

```
private string _nullChecker(string _value){
    if (string.IsNullOrEmpty(_value))
        throw new ArgumentException(_value + "Cannot be null or be empty");
    return _value;
}
```

then, in your Where string statement

```
_Where = " WHERE " + _nullChecker(field) + " " + __nullChecker(operat) + " @" +
_nullChecker(field) + "1 " + _nullChecker(andOr) + " " + _nullChecker(field2) + " " +
_nullChecker(operat2) + " @" + _nullChecker(field2) + "2 ";
```

Not sure with this though. Haven't checked it with actual code. :) Hope this helps

answered Mar 5 '16 at 17:16



Louis Michael

343 1 11

1 Or it might be an extension method. – [tsul](#) Mar 5 '16 at 17:28

You could do this:

0

```
int? GetLength(string s) {
    return s == "" ? -1 : s?.Length;
}
```

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

```
int wholeLength = (s1 + s2).Length; // and so on
if(lengthSum == wholeLength) {
    // No parameter is null or empty
}
```

answered Mar 5 '16 at 17:35



EvilTak

4,101 15 29

Firstly, you can use simple libraries to do argument validation. Check out this one called [Argument Validator](#), which has handy functions that will reduce your overall code by half.

0

Here is an example on how you could do it using the argument validator library:

```
public DB Where(string field, string operat, string value, string andOr, string field2,
string operat2, string value2)
{
    var inputs = new string[] {field, operat, value, andOr, field2, operat2, value2};
    foreach(var input in inputs)
    {
        Throw.IfNullOrEmpty(input, nameof(input));
    }
}
```

edited Aug 21 '16 at 3:19



Cache Staheli

2,016 6 27 36

answered Aug 21 '16 at 0:30



Rezaul Hoque

1 2

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).