How can I make my string property nullable?

I want to make the Middle Name of person optional. I have been using C#.net code first approach. For integer data type its easy just by using ? operator to make in nullable. I am looking for a way to make my sting variable nullable. I tried to search but could not find the way to make it nullable.

33

Below is my code. Please suggest me how to make it nullable.

⭐

5

```
public class ChildrenInfo
{
    [Key]
    public int ChidrenID { get; set; }

    [Required]
    [Display(Name ="First Name")]
    [StringLength(50,ErrorMessage ="First Name cannot exceed more than 50 characters")]
    [RegularExpression(@"^[A-Z]+[a-z]*$",ErrorMessage ="Name cannot have special character,numbers or space")]
    [Column("FName")]
    public string CFName { get; set; }

    [Display(Name ="Middle Name")]
    [RegularExpression(@"^[A-Z]+[a-z]*$",ErrorMessage ="Middle Name cannot have special character,numbers or space")]
    [StringLength(35,ErrorMessage ="Middle Name cannot have more than 35 characters")]
    [Column("MName")]
    public string? CMName { get; set; }
}
```

`c#`   `asp.net-mvc`   `entity-framework`   `ef-migrations`

edited Feb 19 at 9:51
Liam
16.8k 16 79 133

asked Sep 29 '15 at 20:47
Sabin Kumar Sharma
188 1 3 10

---

1   Possible duplicate of [What is the difference between a reference type and value type in c#?](#) – Liam Feb 19 at 9:48

By using our site, you acknowledge that you have read and understand our Cookie Policy, Privacy Policy, and our Terms of Service.

7 Answers

▲

88

▼

✓

String is a reference type and always nullable, you don't need to do anything special. Specifying that a type is nullable is necessary only for value types.
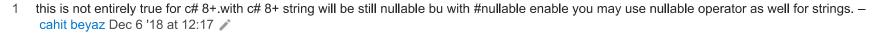
answered Sep 29 '15 at 20:56

aw04
6,891   5   47   75

---

1   this is not entirely true for c# 8+.with c# 8+ string will be still nullable bu with #nullable enable you may use nullable operator as well for strings. –
cahit beyaz Dec 6 '18 at 12:17 ✎

In C# 8.0 reference types may be marked as nullable. One may type `string?` to tell the world this string may be null. Ref: youtube.com/watch?
v=VdC0aoa7ung – Nikola Mar 4 at 23:06

▲

8

▼

System.String is a reference type so you don't need to do anything like

```
Nullable<string>
```

It already has a null value (the null reference):

```
string x = null; // No problems here
```

answered Sep 30 '15 at 5:24

Kashif Mustafa
982   5   19

▲

Strings are nullable in C# anyway because they are reference types. You can just use `public string CMName { get; set; }` and you'll be able to set it to null.
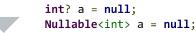
TIM   **1,328**   2   16   31

---

2   Every type is an object in C# (through inheritance). The reason `string`s are nullable is because they're instances of a reference type. — Kapol Sep 29 '15 at 20:58 🖉

1   Thanks, that's a helpful clarification. I've updated my answer accordingly. — Tim Sep 29 '15 at 21:01

▲

1

▼

It's not possible to make reference types Nullable. Only value types can be used in a Nullable structure. Appending a question mark to a value type name makes it nullable. These two lines are the same:

```
int? a = null;
Nullable<int> a = null;
```

answered Sep 29 '15 at 20:59

Vova
1,137   8   24

It's not possible to make reference types Nullable all reference types are implicitly nullable. This is misleading at best — Liam Feb 19 at 9:49

▲

0

▼

As others have pointed out, string is always nullable in C#. I suspect you are asking the question because you are not able to leave the middle name as null or blank? I suspect the problem is with your validation attributes, most likely the RegEx. I'm not able to fully parse RegEx in my head but I **think** your RegEx insists on the first character being present. I could be wrong - RegEx is hard. In any case, try commenting out your validation attributes and see if it works, then add them back in one at a time.

answered Sep 29 '15 at 21:08

Frans
4,740   5   29   51

▲

You don't need to do nothing, the modelbinding will pass null to variable without problems.

519   4   24

`string` type is a reference type, therefore it is nullable by default. You can only use `Nullable<T>` with value types.

0

```
public struct Nullable<T> where T : struct
```

Which means that whatever type is replaced for the generic parameter, it must be a value type.

edited Feb 23 at 14:56

answered Sep 29 '15 at 20:56

Kapol
5,226   3   15   36

I think you meant value type type rather than struct? – Frans Sep 29 '15 at 20:58

@Frans Of course. Thanks. – Kapol Sep 29 '15 at 21:02