How to set enum to null

Asked 8 years, 10 months ago   Active 1 year, 5 months ago   Viewed 196k times

▲

169

▼

★

13

I have an enum

```
string name;

public enum Color
{
    Red,
    Green,
    Yellow
}
```

How to set it to NULL on load.

```
name = "";
Color color = null; //error
```

Edited: My bad, I didn't explain it properly. But all the answers related to nullable is perfect. My situation is What if, I have get/set for the enum in a class with other elements like name, etc. On page load I initiallize the class and try to default the values to null. Here is the scenario (Code is in C#):

```
namespace Testing
{
    public enum ValidColors
    {
        Red,
        Green,
        Yellow
    }

    public class EnumTest
    {
```

```csharp
        public string Name
        {

            get { return name; }
            set { name = value; }
        }

        public ValidColors MyColor
        {

            get { return myColor; }
            set { myColor = value; }
        }

    }

    public partial class _Default : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            EnumTest oEnumTest = new EnumTest();
            oEnumTest.Name = "";
            oEnumTest.MyColor = null; //???
        }
    }

}
```

Then using the suggestions below I changed the above code to make it work with get and set methods. I just need to add "?" in EnumTest class during declaration of private enum variable and in get/set method:

```csharp
public class EnumTest
{
    private string name;
    private ValidColors? myColor; //added "?" here in declaration and in get/set method

    public string Name
    {
        get { return name; }
        set { name = value; }
    }

    public ValidColors? MyColor
    {
        get { return myColor; }
```

Thanks all for the lovely suggestions.

c#   enums   null   nullable

edited Jul 16 '12 at 11:13   asked Dec 2 '10 at 16:20

Rushyo   **Sri Reddy**
6,324   3   27   40   **2,581**   18   57   100

Possible duplicate of [C# Enums: Nullable or 'Unknown' Value?](#) – [Michael Freidgeim](#) Jul 26 '16 at 3:50

5 Answers

You can either use the "?" operator for a nullable type.

335

```
public Color? myColor = null;
```

Or use the standard practice for enums that cannot be null by having the FIRST value in the enum (aka 0) be the default value. For example in a case of color None.

```
public Color myColor = Color.None;
```

edited Mar 11 '15 at 14:45   answered Dec 2 '10 at 16:23

MikeTheLiar   **Rodney S. Foley**
3,614   8   36   60   **6,725**   9   38   62

@equiman no need to cast since the type is specified on the left, you are basically duplicating the first part of my answer that is already accepted. – [Rodney S. Foley](#) Jul 8 '15 at 22:50

4   this answered my question: *Can you add '?' to enumname to make it nullable?* – [LuckyLikey](#) Dec 14 '15 at 8:32

20.07

I don't know what the issue is then, but for my compiler it did give errors until I did the cast. – StefanJanssen Aug 7 at 8:50

If this is C#, it won't work: enums are value types, and can't be `null` .

41

The normal options are to add a `None` member:

```
public enum Color
{
    None,
    Red,
    Green,
    Yellow
}

Color color = Color.None;
```

...or to use `Nullable` :

```
Color? color = null;
```

answered Dec 2 '10 at 16:24

Tim Robinson
45.3k 9 104 122

Make your variable nullable. Like:

22

```
Color? color = null;
```

or

```
Nullable<Color> color = null;
```

9

An enum is a "value" type in C# (means the the enum is stored as whatever value it is, not as a reference to a place in memory where the value itself is stored). You can't set value types to null (since null is used for reference types only).

That being said you can use the built in `Nullable<T>` class which wraps value types such that you can set them to null, check if it `HasValue` and get its actual `Value`. (Those are both methods on the `Nullable<T>` objects.

```
name = "";
Nullable<Color> color = null; //This will work.
```

There is also a shortcut you can use:

```
Color? color = null;
```

That is the same as `Nullable<Color>`;

edited Apr 4 '18 at 13:07   answered Dec 2 '10 at 16:24

Crisfole

4

I'm assuming c++ here. If you're using c#, the answer is probably the same, but the syntax will be a bit different. The enum is a set of int values. It's not an object, so you shouldn't be setting it to null. Setting something to null means you are pointing a pointer to an object to address zero. You can't really do that with an int. What you want to do with an int is to set it to a value you wouldn't normally have it at so that you can tel if it's a good value or not. So, set your colour to -1

```
Color color = -1;
```

Or, you can start your enum at 1 and set it to zero. If you set the colour to zero as it is right now, you will be setting it to "red" because red is zero in your enum.

So.

```
red =1
blue,
green
}
//red is 1, blue is 2, green is 3
Color mycolour = 0;
```

answered Dec 2 '10 at 16:28