Using async to sleep in a thread without freezing [closed]

Asked 6 years, 2 months ago Active 4 years, 7 months ago Viewed 38k times

34

2

So I a label here (""). When the button (button1) is clicked, the label text turns into "Test". After 2 seconds, the text is set back into "". I made this work with a timer (which has an interval of 2000):

```
private void button1_Click(object sender, EventArgs e)
{
    label1.Text = "Test";
    timer.Enabled = true;
}

private void timer_Tick(object sender, EventArgs e)
{
    label1.Text = "";
}
```

This works; however though, I am curious about making it work in an async method.

My code looks like this currently:

```
private void button1_Click(object sender, EventArgs e)
{
    label1.Text = "Test";
    MyAsyncMethod();
}

public async Task MyAsyncMethod()
{
    await Task.Delay(2000);
    label1.Text = "";
}
```

This doesn't work though.

c# asynchronous sleep freeze

edited Dec 22 '14 at 17:48 asked Jun 1 '13 at 23:52

jacobz
1,462 11 26 49

closed as not a real question by Robert Harvey ♦ Jun 2 '13 at 2:14

It's difficult to tell what is being asked here. This question is ambiguous, vague, incomplete, overly broad, or rhetorical and cannot be reasonably answered in its current form. For help clarifying this question so that it can be reopened, visit the help center.

If this question can be reworded to fit the rules in the help center, please edit the question.

I just tried your method and it works fine, on click it changes to "Test" 2 seconds later it changes to "" – sa_ddam213 Jun 2 '13 at 0:01

Could you expound on "This doesn't work"? What were you expecting and what did you observe? Compiler errors? Exception stack traces? – Stephen Cleary Jun 2 '13 at 0:55

1 Answer

75

As I mentioned your code worked fine for me, But perhaps try setting your handler to `async` and running the `Task.Delay` in there.

```
private async void Button_Click_1(object sender, RoutedEventArgs e)
{
    label1.Text = "Test";
    await Task.Delay(2000);
    label1.Text = "";
}
```

answered Jun 2 '13 at 0:04

sa_ddam213
35.7k 6 78 97

1 Yeah, I think it was really difficult to tell what my problem was. However though, this seems to work without any problems :) Thank you and I'll try to word future questions more careful. – jacobz Jun 2 '13 at 17:26

1 For those who do not wish to create an async function: `Task.Delay(2000).Wait();` – Josh Mc Dec 30 '16 at 1:57

2 @JoshMc You are right, but that would cause the GUI thread to be blocked for 2 seconds. That's not a good solution. – sighol Jan 17 '17 at 10:01

True - only recently realized this, good point .Wait() is not preferable, for the vast majority of circumstances. – Josh Mc Jan 18 '17 at 3:33