

Forum: [Hack Forums](#)

Thread: [Loop through richtextbox](#)

Perileos - May 1, 2020, 1:46 pm

Hey there,

Let's see if I can explain this properly.

Basically I have a richtextbox connected with a webbrowser. Each line navigates to a new webpage but I want it to loop.

Right now, each button click is a new line and thus, new page. However, I want it that by 1 press on the button all pages get visited, preferably with a small delay in between.

Current code:

```
Code WebBrowser1.Navigate(RichTextBox1.Lines(i))
```

```
i = i + 1
```

Please excuse me, I'm a big noob if it comes to VB.net but curious to learn.

MavenGradle - May 1, 2020, 1:58 pm

Look into the basic principles of for loops, while loops and loops in general.

Instead of using a WebBrowser component, you could opt to utilize something that's a little less performance hungry

You should make sure that your variable naming is consistent and meaningful. As it stands, the name RichTextBox1 doesn't really mean anything.

A shorter way to increment a variable with 1 is to use "+= 1"

Why would you want to introduce an artificial delay?

Resources regarding loops in VB.NET:

<https://docs.microsoft.com/en-us/dotnet/visual-basic/programming-guide/language-features/control-flow/loop-structures>

Perileos - May 1, 2020, 2:04 pm

(May 1, 2020 01:58 PM)MavenGradle Wrote: [\[CITE\]](#) Look into the basic principles of for loops, while loops and loops in general.

Instead of using a WebBrowser component, you could opt to utilize something that's a little less performance hungry

You should make sure that your variable naming is consistent and meaningful. As it stands, the name RichTextBox1 doesn't really mean anything.

A shorter way to increment a variable with 1 is to use "+= 1"

Why would you want to introduce an artificial delay?

Resources regarding loops in VB.NET:

<https://docs.microsoft.com/en-us/dotnet/visual-basic/programming-guide/language-features/control-flow/loop-structures>

Regarding the webbrowser, I need to use it to sign into a website. Considering the website uses randomised ids/names in their elements, I thought it'd be easier to make it like this.

I have altered the increment to the variable, thanks for the tip.

The artificial delay needs to be there because I want it to browse each X seconds.

MavenGradle - May 1, 2020, 2:12 pm

(May 1, 2020 02:04 PM)Perileos Wrote: [\[CITE\]](#) Regarding the webbrowser, I need to use it to sign into a website. Considering the website uses randomised ids/names in their elements, I thought it'd be easier to make it like this.

You should probably be using something like Selenium from what I understand:

<https://www.selenium.dev/>

<https://www.youtube.com/watch?v=8Ok7AoZceEE>

<https://www.toolsqa.com/selenium-c-sharp/>

You should also look into the possibility of making use of the API system of whatever service it is that you're trying to automate something for. RESTful-API's are available for most major sites (e.g. Facebook has them, Reddit has them, etc.). Using the API system has advantages such as direct communication with the backend server, instead of having to load the website and essentially downloading a bunch of files that you don't really need. Additionally, error handling can be performed in a much cleaner way, reducing the complexity of your application in general.

(May 1, 2020 02:04 PM)Perileos Wrote: [\[CITE\]](#) The artificial delay needs to be there because I want it to browse each X seconds.

That's not really a valid reason in my opinion. You are artificially slowing something down and making your application more complex. The only reason why I could see you doing this would be to circumvent some kind of spam filter, in which case you're doing something that you're not supposed to be doing. Regardless of that, it is never a good idea to introduce delays by using something like: Code Thread.Sleep(x)

This is just bad practice and the only time it could arguable be useful is when looking at unit tests.

Perileos - May 1, 2020, 2:18 pm

(May 1, 2020 02:12 PM)MavenGradle Wrote: [\[CITE\]](#) You should probably be using something like Selenium from what I understand:

<https://www.selenium.dev/>

<https://www.youtube.com/watch?v=8Ok7AoZceEE>

<https://www.toolsqa.com/selenium-c-sharp/>

You should also look into the possibility of making use of the API system of whatever service it is that you're trying to automate something for. RESTful-API's are available for most major sites (e.g. Facebook has them, Reddit has them, etc.). Using the API system has advantages such as direct communication with the backend server, instead of having to load the website and essentially downloading a bunch of files that you don't really need. Additionally, error handling can be performed in a much cleaner way, reducing the complexity of your application in general.

That's not really a valid reason in my opinion. You are artificially slowing something down and making your application more complex. The only reason why I could see you doing this would be to circumvent some kind of spam filter, in which case you're doing something that you're not supposed to be doing. Regardless of that, it is never a good idea to introduce delays by using something like: Code Thread.Sleep(x)

This is just bad practice and the only time it could arguable be useful is when looking at unit tests.

I'm going to research Selenium more.

Basically I intend to scrape information from a website and if that information is present I want the

program to do an action. Not necessarily meant for spam. I'm a noob when it comes to programs hence why I can't/don't know how to use an api.

MavenGradle - May 1, 2020, 2:25 pm

(May 1, 2020 02:18 PM)Perileos Wrote: [\[CITE\]](#) I'm going to research Selenium more.

Basically I intend to scrape information from a website and if that information is present I want the program to do an action. Not necessarily meant for spam. I'm a noob when it comes to programs hence why I can't/don't know how to use an api.

Feel free to ask any questions you have here. Using an API system can be as simple as calling using a [WebRequest](#) to make a [HTTP POST](#) call with (serialized) [JSON](#) data. You could make use of [Json.NET](#) to serialize .NET objects into JavaScript Object Notation (JSON).

In any case, using the API approach will save you a lot of system resources and speed up your development process in general when you get the hang of it and it should always be your first choice of action when looking at automation in regards to webdevelopment. It's very useful to look into those things using the links I've included throughout this post. Happy learning!

Perileos - May 1, 2020, 2:27 pm

(May 1, 2020 02:25 PM)MavenGradle Wrote: [\[CITE\]](#) Feel free to ask any questions you have here. Using an API system can be as simple as calling using a [WebRequest](#) to make a [HTTP POST](#) call with (serialized) [JSON](#) data. You could make use of [Json.NET](#) to serialize .NET objects into JavaScript Object Notation (JSON).

In any case, using the API approach will save you a lot of system resources and speed up your development process in general when you get the hang of it and it should always be your first choice of action when looking at automation in regards to webdevelopment. It's very useful to look into those things using the links I've included throughout this post. Happy learning!

Thank you again. You're a very helpful member.

I'm trying to learn from this instead of hiring someone.

Who Soup - May 2, 2020, 3:25 pm

(May 1, 2020 02:27 PM)Perileos Wrote: [\[CITE\]](#) Thank you again. You're a very helpful member. I'm trying to learn from this instead of hiring someone.

Don't ever listen to MavenGradle, this guy is known to not help people he's here to spew a bunch of shît he doesn't know and stroke his ego.

Here's the solution to your question OP.

```
Code    For i As Integer = 0 To RichTextBox1.Lines.Count
        WebBrowser1.Navigate(RichTextBox1.Lines(i))
        Threading.Thread.Sleep(1000)
    Next
```

End Sub

Code Explained:

Spoiler ([Click to View](#))

First you want to create a basic loop and have it finish when there aren't any more lines to read.

```
Code    For i As Integer = 0 To RichTextBox1.Lines.Count
```

Next

Next, within the loop you want your web browser to go through each line in the rich textbox (defined by an index; aka which number line it is which starts at 0) and navigate to that line. Every time "i"

goes around in the loop it will run this code to navigate to said web page. It will continue counting until there are no more lines which will continually progress you to the next URL in the list.

```
Code WebBrowser1.Navigate(RichTextBox1.Lines(i))
```

Finally you stated you wanted to have some time between each request. You can easily achieve this with `threading.thread.sleep`. the number inside of the parenthesis (in this case 1000) means 1000 milliseconds. If you want to change this number just know there's 1000 milliseconds in 1 second.

```
Code Threading.Thread.Sleep(1000)
```

If you need any further explanation please let me know ?

Perileos - May 2, 2020, 3:35 pm

(May 2, 2020 03:25 PM)Who Soup Wrote: [\[CITE\]](#) Don't ever listen to MavenGradle, this guy is known to not help people he's here to spew a bunch of shít he doesn't know and stroke his ego.

Here's the solution to your question OP.

```
Code      For i As Integer = 0 To RichTextBox1.Lines.Count
           WebBrowser1.Navigate(RichTextBox1.Lines(i))
           Threading.Thread.Sleep(1000)
           Next
```

```
End Sub
```

Code Explained:

Spoiler ([Click to View](#))

First you want to create a basic loop and have it finish when there aren't any more lines to read.

```
Code      For i As Integer = 0 To RichTextBox1.Lines.Count
```

```
           Next
```

Next, within the loop you want your web browser to go through each line in the rich textbox (defined by an index; aka which number line it is which starts at 0) and navigate to that line. Every time "i" goes around in the loop it will run this code to navigate to said web page. It will continue counting until there are no more lines which will continually progress you to the next URL in the list.

```
Code WebBrowser1.Navigate(RichTextBox1.Lines(i))
```

Finally you stated you wanted to have some time between each request. You can easily achieve this with `threading.thread.sleep`. the number inside of the parenthesis (in this case 1000) means 1000 milliseconds. If you want to change this number just know there's 1000 milliseconds in 1 second.

```
Code Threading.Thread.Sleep(1000)
```

If you need any further explanation please let me know ?

Cheers.

That was something I was looking for. It's something I understand hahaha.

Thanks man! ?

itcorp - May 2, 2020, 4:26 pm

actually in this case it is better to use foreach loop instead of for loop especially when it comes to the larger amount of data contained in RichTextBox. In for loop you accessing lines on every

iteration,so i recommend foreach loop in this case unless you just want to work with indexes.

MavenGradle - May 2, 2020, 8:17 pm

(May 2, 2020 03:25 PM)Who Soup Wrote: [\[CITE\]](#) Don't ever listen to MavenGradle, this guy is known to not help people he's here to spew a bunch of shít he doesn't know and stroke his ego.

Here's the solution to your question OP.

```
Code    For i As Integer = 0 To RichTextBox1.Lines.Count
        WebBrowser1.Navigate(RichTextBox1.Lines(i))
        Threading.Thread.Sleep(1000)
    Next
```

End Sub

Code Explained:

Spoiler ([Click to View](#))

First you want to create a basic loop and have it finish when there aren't any more lines to read.

```
Code    For i As Integer = 0 To RichTextBox1.Lines.Count
```

Next

Next, within the loop you want your web browser to go through each line in the rich textbox (defined by an index; aka which number line it is which starts at 0) and navigate to that line. Every time "i" goes around in the loop it will run this code to navigate to said web page. It will continue counting until there are no more lines which will continually progress you to the next URL in the list.

```
Code WebBrowser1.Navigate(RichTextBox1.Lines(i))
```

Finally you stated you wanted to have some time between each request. You can easily achieve this with `threading.thread.sleep`. the number inside of the parenthesis (in this case 1000) means 1000 milliseconds. If you want to change this number just know there's 1000 milliseconds in 1 second.

```
Code Threading.Thread.Sleep(1000)
```

If you need any further explanation please let me know ?

As pointed out, the code you shared is exactly what you're supposed to avoid doing. You do not make use of any viable variable names, you have no error handling and you make an entire application unresponsive for 1000 milliseconds on every iteration, not to mention that you make use of a component that utilizes Internet Explorer in its core, of all things. A component that, by the way, has a very clear remark on the official documentation that it is resource heavy.

All of this garbage could have been avoided if you simply wrote clean code.

Unfortunately, some people who lack the required skill for writing decent code or simple comprehension skills will opt for a solution like this but if anyone's genuinely interested in learning proper programming techniques, this is a prime example on how not to write code and how to lose your job as a programmer in no time. That is, if you work for a respectable company that actually verifies (which is called a code review) the code its employees produce.

The reasoning behind this has been explained in this and in previous posts.

Who Soup - May 2, 2020, 8:55 pm

(May 2, 2020 08:17 PM)MavenGradle Wrote: [\[CITE\]](#) As pointed out, the code you shared is exactly what you're supposed to avoid doing. You do not make use of any viable variable names, you have

no error handling and you make an entire application unresponsive for 1000 milliseconds, not to mention that you make use of a component that utilizes Internet Explorer in its core, of all things. A component that, by the way, has a very clear remark on the official documentation that it is resource heavy.

All of this garbage could have been avoided if you simply wrote clean code.

Unfortunately, some people who lack the required skill for writing decent code or simple comprehension skills will opt for a solution like this but if anyone's genuinely interested in learning proper programming techniques, this is a prime example on how not to write code and how to lose your job as a programmer in no time. That is, if you work for a respectable company that actually verifies (which is called a code review) the code its employees produce.

The reasoning behind this has been explained in this and in previous posts.

Your posts seriously bring down the quality of this forum it's honestly a shame that you're not banned yet. People will take notice that your posts do nothing for them while mine provides a precise working answer. If you absolutely can't help being an ass hole at least put more effort into your posts and post working code. No one wants to be bombarded with links that don't even apply to what's being asked.

The fact that you write an entire essay in your post but can't write 5 lines of code says everything about you and your credibility.

MavenGradle - May 3, 2020, 4:49 am

(May 2, 2020 08:55 PM)Who Soup Wrote: [\[CITE\]](#) Your posts seriously bring down the quality of this forum it's honestly a shame that you're not banned yet. People will take notice that your posts do nothing for them while mine provides a precise working answer. If you absolutely can't help being an ass hole at least put more effort into your posts and post working code. No one wants to be bombarded with links that don't even apply to what's being asked.

The fact that you write an entire essay in your post but can't write 5 lines of code says everything about you and your credibility.

Your personal inquiries and insults are laughable and besides the point.

Your code provides the OP with a solution that is not appropriate, as I've already pointed out.

Instead of this post you should have been posting a proper solution to the question at hand or at the very least correct your previous post.

If anything brings down the quality of this subforum, it would be the code you've posted above, but apparently you lack the insight to realize that it is garbage code. If you took the time of day to read, you'd understand exactly why it is garbage code.

Dotexe - May 3, 2020, 4:58 am

(May 2, 2020 03:25 PM)Who Soup Wrote: [\[CITE\]](#) Don't ever listen to MavenGradle, this guy is known to not help people he's here to spew a bunch of shít he doesn't know and stroke his ego.

LMFAO, when i started with React, he was the only one who helped me thru the process. So saying that he is "known to not help people" is just irrelevant. I would've tell you this or that, you're just not worth my time.

Joel - May 6, 2020, 8:08 am

(May 2, 2020 08:17 PM)MavenGradle Wrote: [\[CITE\]](#) As pointed out, the code you shared is exactly what you're supposed to avoid doing. You do not make use of any viable variable names, you have

no error handling and you make an entire application unresponsive for 1000 milliseconds on every iteration, not to mention that you make use of a component that utilizes Internet Explorer in its core, of all things. A component that, by the way, has a very clear remark on the official documentation that it is resource heavy.

All of this garbage could have been avoided if you simply wrote clean code.

Unfortunately, some people who lack the required skill for writing decent code or simple comprehension skills will opt for a solution like this but if anyone's genuinely interested in learning proper programming techniques, this is a prime example on how not to write code and how to lose your job as a programmer in no time. That is, if you work for a respectable company that actually verifies (which is called a code review) the code its employees produce.

The reasoning behind this has been explained in this and in previous posts.

Not everyone can write the correct thing on their first try man. Your code evolves as you get better.

MavenGradle - May 6, 2020, 8:13 am

(May 6, 2020 08:08 AM)Joel Wrote: [\[CITE\]](#) Not everyone can write the correct thing on their first try man. Your code evolves as you get better.

Fully agree with that statement, which is exactly why the suggestions in previous posts have, in fact, been posted; in order to help some of the readers evolve. It's ultimately up to them whether or not they follow the advice that is given to them, but said advice is given in good will.

It's another thing entirely to completely disregard proper advice and not acknowledge that your way of working is incorrect. Learning and evolving in a certain branch can only happen if the person is willing to admit to their mistakes and actually implement a proper solution.

Joel - May 6, 2020, 8:20 am

(May 6, 2020 08:13 AM)MavenGradle Wrote: [\[CITE\]](#) Fully agreed with that statement, which is exactly why the suggestions in previous posts have, in fact, been posted; in order to help some of the readers evolve. It's ultimately up to them whether or not they follow the advice that is given to them, but said advice is given in good will.

It's another thing entirely to completely disregard proper advice and not acknowledge that your way of working is incorrect. Learning and evolving in a certain branch can only happen if the person is willing to admit to their mistakes and actually implement it properly.

Agree'd, people have to want to actively learn. I myself genuinely throw together sloppy projects for the sake of speed and then go back and fix them once I have a stable shitty version.

Xogg - May 7, 2020, 6:24 pm

Or, instead of the integer based loop (For i As Integer = 0 To RichTextBox1.Lines.Count) you can use a string based loop

For Each line as String in RichTextBox1.Lines

Who Soup - May 9, 2020, 6:02 pm

(May 7, 2020 06:24 PM)Xogg Wrote: [\[CITE\]](#) Or, instead of the integer based loop (For i As Integer = 0 To RichTextBox1.Lines.Count) you can use a string based loop

For Each line as String in RichTextBox1.Lines

Good suggestion, I have a bad habit of doing integer loops for everything lol string loop for a string based function makes more sense.

Xogg - May 10, 2020, 6:32 am

(May 9, 2020 06:02 PM)Who Soup Wrote: [\[CITE\]](#) Good suggestion, I have a bad habit of doing integer loops for everything lol
string loop for a string based function makes more sense.
I also find this a lot easier for beginners to understand.

itcorp - May 10, 2020, 12:48 pm

As I stated earlier using for loops and its variations they are simply not the best solution here in this case,you can check it very easily by a simple unit test which you will determine for yourself how much is the for loop slower than foreach loop.What is very important when you enter the world of programming is to have an open mind and be prepared for suggestions that can help you in your personal development and that's what @MavenGradle tried to say in my opinion because someone tried to do this before you and saw that it was not the best solution.I wish you all the best buddy.

Who Soup - May 10, 2020, 9:15 pm

(May 10, 2020 12:48 PM)itcorp Wrote: [\[CITE\]](#) As I stated earlier using for loops and its variations they are simply not the best solution here in this case,you can check it very easily by a simple unit test which you will determine for yourself how much is the for loop slower than foreach loop.What is very important when you enter the world of programming is to have an open mind and be prepared for suggestions that can help you in your personal development and that's what @MavenGradle tried to say in my opinion because someone tried to do this before you and saw that it was not the best solution.I wish you all the best buddy.

That's what you and all your other butt buddies don't understand. There definitely are better solutions but if you're not actively contributing why are you even attempting to help someone? OP asked a question and I gave him a working solution. Especially if you know of a better way why would you not share code? Isn't that the point of trying to help someone? You know how they say a picture is worth 1000 words? Code is worth 1000 explanations.

Until that happens, I don't have any respect for any of your opinions.

itcorp - May 11, 2020, 1:31 am

(May 10, 2020 09:15 PM)Who Soup Wrote: [\[CITE\]](#) That's what you and all your other butt buddies don't understand. There definitely are better solutions but if you're not actively contributing why are you even attempting to help someone? OP asked a question and I gave him a working solution. Especially if you know of a better way why would you not share code? Isn't that the point of trying to help someone? You know how they say a picture is worth 1000 words? Code is worth 1000 explanations.

Until that happens, I don't have any respect for any of your opinions.

I didn't share the code because you gave it a solution, so I upgraded to your existing answer therefore, I did not feel the need to give a code.

You misunderstood me,I'm not taking anyone's side because you are both right in some ways therefore, I will share a simple code to see what I'm talking about.I wish you all the best buddy.

here are pictures of 1000 words ;)

Code Public Class Form1

Private stopWatch As Stopwatch

Private Sub btnForLoop_Click(sender As Object, e As EventArgs) Handles btnForLoop.Click
stopWatch = New Stopwatch()
stopWatch.Start()

```

        For i As Integer = 0 To RichTextBox1.Lines.Count() - 1
#If DEBUG Then
            System.Diagnostics.Debug.WriteLine(RichTextBox1.Lines(i))
#End If
        Next

        stopWatch.[Stop]()
        Dim ts As TimeSpan = stopWatch.Elapsed
        Dim elapsedTime As String = String.Format("{0:00}:{1:00}:{2:00}.{3:00}", ts.Hours, ts.Minutes,
ts.Seconds, ts.Milliseconds / 10)
        lblTime.Text = elapsedTime
    End Sub

    Private Sub btnForLoop1_Click(sender As Object, e As EventArgs) Handles btnForLoop1.Click
        stopWatch = New Stopwatch()
        stopWatch.Start()

        For i As Integer = 0 To RichTextBox1.Lines.Length - 1
#If DEBUG Then
            System.Diagnostics.Debug.WriteLine(RichTextBox1.Lines(i))
#End If
        Next

        stopWatch.[Stop]()
        Dim ts As TimeSpan = stopWatch.Elapsed
        Dim elapsedTime As String = String.Format("{0:00}:{1:00}:{2:00}.{3:00}", ts.Hours, ts.Minutes,
ts.Seconds, ts.Milliseconds / 10)
        lblTime0.Text = elapsedTime
    End Sub

    Private Sub btnForeachLoop_Click(sender As Object, e As EventArgs) Handles
btnForeachLoop.Click
        stopWatch = New Stopwatch()
        stopWatch.Start()

        For Each i In RichTextBox1.Lines
#If DEBUG Then
            System.Diagnostics.Debug.WriteLine(i)
#End If
        Next

        stopWatch.[Stop]()
        Dim ts As TimeSpan = stopWatch.Elapsed
        Dim elapsedTime As String = String.Format("{0:00}:{1:00}:{2:00}.{3:00}", ts.Hours, ts.Minutes,
ts.Seconds, ts.Milliseconds / 10)
        lblTime1.Text = elapsedTime
    End Sub

End Class

```

Who Soup - May 11, 2020, 10:05 am

(May 11, 2020 01:31 AM)itcorp Wrote: [CITE](#) I didn't share the code because you gave it a solution,

so I upgraded to your existing answer therefore, I did not feel the need to give a code.

You misunderstood me, I'm not taking anyone's side because you are both right in some ways therefore, I will share a simple code to see what I'm talking about. I wish you all the best buddy.

here are pictures of 1000 words ;)

Code Public Class Form1

```
Private stopWatch As Stopwatch
Private Sub btnForLoop_Click(sender As Object, e As EventArgs) Handles btnForLoop.Click
    stopWatch = New Stopwatch()
    stopWatch.Start()

    For i As Integer = 0 To RichTextBox1.Lines.Count() - 1
#If DEBUG Then
        System.Diagnostics.Debug.WriteLine(RichTextBox1.Lines(i))
#End If
    Next

    stopWatch.[Stop]()
    Dim ts As TimeSpan = stopWatch.Elapsed
    Dim elapsedTime As String = String.Format("{0:00}:{1:00}:{2:00}.{3:00}", ts.Hours, ts.Minutes,
ts.Seconds, ts.Milliseconds / 10)
    lblTime.Text = elapsedTime
End Sub

Private Sub btnForLoop1_Click(sender As Object, e As EventArgs) Handles btnForLoop1.Click
    stopWatch = New Stopwatch()
    stopWatch.Start()

    For i As Integer = 0 To RichTextBox1.Lines.Length - 1
#If DEBUG Then
        System.Diagnostics.Debug.WriteLine(RichTextBox1.Lines(i))
#End If
    Next

    stopWatch.[Stop]()
    Dim ts As TimeSpan = stopWatch.Elapsed
    Dim elapsedTime As String = String.Format("{0:00}:{1:00}:{2:00}.{3:00}", ts.Hours, ts.Minutes,
ts.Seconds, ts.Milliseconds / 10)
    lblTime0.Text = elapsedTime
End Sub

Private Sub btnForeachLoop_Click(sender As Object, e As EventArgs) Handles
btnForeachLoop.Click
    stopWatch = New Stopwatch()
    stopWatch.Start()

    For Each i In RichTextBox1.Lines
#If DEBUG Then
        System.Diagnostics.Debug.WriteLine(i)
#End If
```

Next

```
stopWatch.[Stop]()  
Dim ts As TimeSpan = stopWatch.Elapsed  
Dim elapsedTime As String = String.Format("{0:00}:{1:00}:{2:00}.{3:00}", ts.Hours, ts.Minutes,  
ts.Seconds, ts.Milliseconds / 10)  
lblTime1.Text = elapsedTime  
End Sub
```

End Class

I take it back, you're high quality :)

Alien - May 11, 2020, 9:21 pm

In my early days of coding in this language, I was very stubborn about only using a webbrowser component. I completely recommend against it now. Web requests is the best approach to any website scraping application. You should look into get and post requests as well as find a basic regex extract function that will help you grab specific info on the website you are trying to scrape.

The benefits of using web requests can be game changing. Your application won't be frozen for a second at a time, and you can easily multi thread it the method/function you are creating to make it run 10x smoother. You have so much more control over the application using this approach