Chat GPT Can (and maybe should) Change the Way You Code

Chat GPT is all over the news. Most of the stories are about Microsoft's investment, or schools wanting to ban it because it can generate papers that teachers can't tell were generated by AI and not the student.

Here's a different story… Chat GPT helped recently helped me VASTLY improve some code, shrinking the solution from 27 lines to 6! Granted, the 27-line solution was written to be very obvious to the potential users of the code. But the 6-line solution is equally readable. What's more, the code is in a relatively-obscure language called DataWeave which is used by MuleSoft developers (and recently open sourced so you, too, can use it!).

Back Story

One of my colleagues recently discovered that Chat GPT can write code for you. Yup, type in a phrase describing what you need in Java, C, even DataWeave, and Chat GPT will spit out code. Often, the code is even correct and does what you ask, though not always.

Today, another colleague asked in a forum if there was a DataWeave function to generate a string of 8 random alpha-numeric characters suitable for using in a password (so, ideally using upper and lower case letters). There is no such function in the standard modules. So, I wrote one. You can view the code in the section First Solution.

I'd been playing with Chat GPT for a different reason, and thought

> *"I wonder what code it would write if I asked it to solve this same problem?"*

Here is what I asked Chat GPT:

> *Write dataweave to generate a string of random letters and numbers 8 characters long*

Here is what it wrote:

```
randomString: '#[(0..9 ++ 'ABCDEFGHIJKLMNOPQRSTUVWXYZ' ++
'abcdefghijklmnopqrstuvwxyz')[((Math.random()*62) as Number) as Integer] as String][8]'
```

As it turns out, that code doesn't work (go ahead and put it into DataWeave playground at this website and see for yourself). But, it made me think:

> *I am already creating the set of possible characters (chars), and I'm already selecting N characters from that set by randomly picking one, and I'm already using a range to drive the generation, why not just do it in one real step like Chat GPT is trying to do?*

The Solution – generateString(n)

Given the above, here is the code that I wrote:

```
var generateString = (n: Number = 8) ->
do {
    var x = ('0123456789' ++ 'ABCDEFGHIJKLMNOPQRSTUVWXYZ' ++ 'abcdefghijklmnopqrstuvwxyz')
    ---
    0 to n reduce (item, acc="") -> acc ++ x[ceil(random()*sizeOf(x))]
}
```

While I *could* have done it in one line, this is more readable, and also more generic:

1. You can add characters into the variable “x” to include special characters like !, -, *, etc.
2. You can generate strings of any length by setting passing that number to override the default length of 8 characters

You are all free to use this code as you wish in your MuleSoft / DataWeave projects.

Summary

While generateString() is one take-away from this blog post, a larger takeaway is that you can use Chat GPT to help you think about your code, and maybe you should.

First Solution:

Here is the code I wrote for the first solution. NOTE: I was deliberate in setting up variable after variable to make it clear what the code did. Otherwise, I could have compressed out several lines of code. But, it still would have had three functions.

```
import substringAfter from dw::core::Strings
import withMaxSize from dw::core::Strings
import repeat from dw::core::Strings
var arrayToString = (arr: Array) ->
    arr reduce (char, acc="") -> acc ++ (char as String)
var randomChar = () ->
    do {
        var chars = [ 'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q',
'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z']
        var n = randomInt(sizeOf(chars)-1)
        var char1 = chars[n]
        var n2 = randomInt(2)
        var char2 = if(n2 == 0) char1 else upper(char1)
        ---
        char2
    }

var generateString = (size: Number) ->
        do {
                var str1size = size/2
                var str2size = size - str1size
                var string1 = random() as String substringAfter  "." withMaxSize  str1size
                var arr1 = (1 to str2size) map randomChar()
                var string2 = arrayToString(arr1)
                ---
                string1 ++ string2
        }
```

Legal