

# Lab 3: Web Application with Genie

Mohamed Raed Boukari  
Dept. of EE (AII21)  
ISET Bizerte — Tunisia  
raedboukari2018@gmail.com

Sana Ben Said  
Dept. of EE (AII21)  
ISET Bizerte — Tunisia  
bensaidsana06@gmail.com

## I. INTRODUCTION

In this Lab we used Genie Framework in julia , Genie is a full-stack MVC web framework that provides a streamlined and efficient workflow for developing modern web applications. It builds on Julia's strengths (high-level, high-performance, dynamic, JIT compiled), exposing a rich API and a powerful toolset for productive web development.

## II. EXERCICES

- In the first task we add extra slide that modify the behaviour of the sine wave graph by adding : **Phase ranging between  $-\pi$  and  $\pi$ , changes by a step of  $\frac{\pi}{100}$**

```
appjl x appjl.html
appjl > ...
1 using GenieFramework
2 @genietools
3
4 @app begin
5
6     @in N::Int32 = 1000
7     @in amp::Float32 = 0.25
8     @in freq::Int32 = 1
9     @in ph::Float64 = 3.14/100
10
11
12     @out my_sine = PlotData()
13
14     @onchange N, amp, freq, ph begin
15         x = range(0, 1, length=N)
16         y = amp*sin.(2*pi*freq*x.+ph)
17
18         my_sine = PlotData(x=x,
19                             y=y,
20                             plot=StipplePlotly.Charts.PLOT_TYPE_LINE)
21     end
22
23 end
24
25 @page["/", "app.jl.html"]
```

Figure 1: Adding the phase function in julia

```
appjl appjl.html
appjl.html > div.row > div.st-col.col-12.col-sm.st-module > q-slider
1 <header class="st-header q-pa-sm">
2   <h1 class="st-header__title text-h3" Sinewave Dashboard </h1>
3 </header>
4
5 <div class="row">
6   <div class="st-col col-12 col-sm st-module">
7     <p><b># Samples</b></p>
8     <q-slider v-model="N"
9       :min="10" :max="1000"
10      :step="10" :label="true">
11   </q-slider>
12 </div>
13
14   <div class="st-col col-12 col-sm st-module">
15     <p><b>Amplitude</b></p>
16     <q-slider v-model="amp"
17       :min="0" :max="3"
18      :step=".5" :label="true">
19   </q-slider>
20 </div>
21   <div class="st-col col-12 col-sm st-module">
22     <p><b>phase</b></p>
23     <q-slider v-model="ph"
24       :min="3.14" :max="3.14"
25      :step="3.14/100" :label="true">
26   </q-slider>
27 </div>
28
29   <div class="st-col col-12 col-sm st-module">
30     <p><b>Frequency</b></p>
31     <q-slider v-model="freq"
32       :min="0" :max="10"
33      :step="1" :label="true">
34   </q-slider>
35 </div>
36 </div>
37
38 <div class="row">
39   <div class="st-col col-12 col-sm st-module">
40     <p><b>Sinewave</b></p>
41     <plotly :data="my_sine"> </plotly>
42   </div>
43 </div>
```

Figure 2: Adding the phase function in HTML

- Then in the second task we add the offset function : **Offset varies from  $-0.5$  to  $1$ , by a step of  $0.1$**

```

app.jl x app.jl.html
1 using GenieFramework
2 @genietools
3
4 @app begin
5
6     @in N::Int32 = 1000
7     @in amp::Float32 = 0.25
8     @in freq::Int32 = 1
9     @in ph::Float64 = 3.14/100
10    @in offset::Int32 = 0
11
12
13
14    @out my_sine = PlotData()
15
16    @onchange N, amp, freq, ph, offset begin
17        x = range(0, 1, length=N)
18        y = amp*sin.(2*pi*freq*x.+ph).+offset
19
20        my_sine = PlotData(x=x,
21                          y=y,
22                          plot=StipplePlotly.Charts.PLOT_TYPE_LINE)
23    end
24
25 end
26 @page["/", "app.jl.html"]

```

Figure 3: Adding the offset function in Julia

```

app.jl x app.jl.html x
1 app.jl.html > div.row > div.st-col.col-12.col-sm.st-module > q-slider
2
3 <div class="row">
4
5 <div class="st-col.col-12.col-sm.st-module">
6 <q-slider v-model="N">
7
8 :step="10" :label="true">
9 </q-slider>
10 </div>
11
12 <div class="st-col.col-12.col-sm.st-module">
13 <p><b>Amplitude</b></p>
14 <q-slider v-model="amp">
15 :min="0" :max="3"
16 :step=".5" :label="true">
17 </q-slider>
18 </div>
19 <div class="st-col.col-12.col-sm.st-module">
20 <p><b>Phase</b></p>
21 <q-slider v-model="ph">
22 :min="3.14" :max="3.14"
23 :step="3.14/100" :label="true">
24 </q-slider>
25 </div>
26 <div class="st-col.col-12.col-sm.st-module">
27 <p><b>Offset</b></p>
28 <q-slider v-model="offset">
29 :min="0.5" :max="1"
30 :step="0.1" :label="true">
31 </q-slider>
32 </div>
33
34 <div class="st-col.col-12.col-sm.st-module">
35 <p><b>Frequency</b></p>
36 <q-slider v-model="freq">
37 :min="0" :max="10"
38 :step="1" :label="true">
39 </q-slider>
40 </div>
41
42 <div class="row">
43 <div class="st-col.col-12.col-sm.st-module">
44 <p><b>Sine wave</b></p>
45 <plotly :data="my_sine"> </plotly>
46 </div>
47
48 </div>
49
50 </div>

```

Figure 4: Adding the offset function in HTML

- After that we open the Julia REPL in order to open the web app



Figure 5: Julia REPL

- Now , we have the sine wave and we can change every parameter thanks to the graphic interface

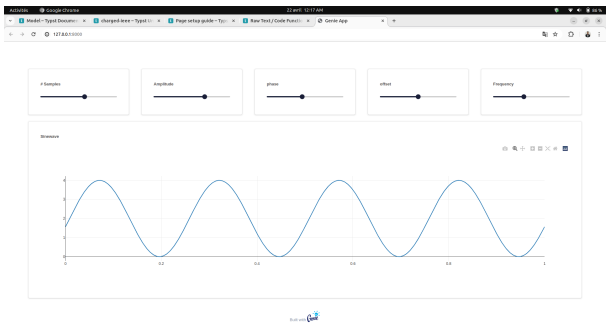


Figure 6: Sine Wave

### III. CONCLUSION

In this Lab we have the ability to use Genie Lab in julia to design a mathematical web app

### REFERENCES