

LAB #3: WEB APPLICATION WITH GENIE

Amdouni Aziz
Dept. of EE
ISET Bizerte — Tunisia
azizamdouni007@gmail.com
 Amdouni Aziz

I. INTRODUCTION

In this report, I will explain what i had done to add to the previous basic web application two extra sliders to change the phase and the offset which also modify the behaviour of the sine wave graph.

II. SINE WAVE CONTROL

A. Julia coding

To the previous app.jl file , i had add two inputs **phase** and **offset**. There types are Float64 and Float32 and default values are 0. Also,i had added their names after **onchange** so we can control them as we wish. This work is shown in code below :

```
using GenieFramework
@genietools

@app begin

    @in N::Int32 = 1000
    @in amp::Float32 = 0.25
    @in freq::Int32 = 1

    @out my_sine = PlotData()

    @onchange N, amp, freq begin
        x = range(0, 1, length=N)
        y = amp*sin.(2*pi*freq*x)

        my_sine = PlotData(x=x,
                           y=y,

    plot=StipplePlotly.Charts.PLOT_TYPE_LINE)
    end

end

@page("/", "app.jl.html")
```

B. HTML coding

For app.jl.html file, i had add two sliders :

1) The phase:

- Firstly , i had linked the slider's value to a variable named **ph**.
- Secondly , i had set the minimum value of the slider to $-\pi$.
- Thridly , i had set the maximum value of the slider to π .
- Fourthly , i had set the step increment of the slider to $\frac{\pi}{100}$.
- In the end ,we specified that labels should be displayed on the slider.

2) The offset:

- Firstly , i had linked the slider's value to a variable named **off**.
- Secondly , i had set the minimum value of the slider to -0.5 .
- Thridly , i had set the maximum value of the slider to 1.
- Fourthly , i had set the step increment of the slider to 0.1.
- In the end ,i specified that labels should be displayed on the slider.

This is shown in the html code below :

```
<header class="st-header q-pa-sm">
  <h1 class="st-header__title text-h3" Sinewave
Dashboard </h1>
</header>

<div class="row">
  <div class="st-col col-12 col-sm st-module">
    <p><b># Samples</b></p>
    <q-slider v-model="N"
:min="10" :max="1000"
:step="10" :label="true">
  </q-slider>
</div>

  <div class="st-col col-12 col-sm st-module">
    <p><b>Amplitude</b></p>
    <q-slider v-model="amp"
:min="0" :max="3"
:step=".5" :label="true">
  </q-slider>
</div>

  <div class="st-col col-12 col-sm st-module">
    <p><b>Frequency</b></p>
```

```

<q-slider v-model="freq"
  :min="0" :max="10"
  :step="1" :label="true">
</q-slider>
</div>

<div class="row">
  <div class="st-col col-12 col-sm st-module">
    <p><b>Sinewave</b></p>
    <plotly :data="my_sine"> </plotly>
  </div>
</div>

```

C. Graphical interface

After checking the app.jl and app.jl.html codes, I had opened the terminal of vs code and opened Julia and typed the commands below to use the GenieFramework to develop a web application.

```

julia> using GenieFramework
julia> cd
julia> Genie.loadapp()
julia> up()

```

```
using GenieFramework
```

This line imports the GenieFramework module into the Julia environment, allowing you to access the functionality provided by the Genie web framework.

```
cd()
```

This chooses the current working directory in Julia to the specified path where your web application is located.

```
Genie.loadapp()
```

This command loads the web application defined in the current directory into the Genie framework. It sets up the necessary configurations and initializes the application.

```
up()
```

This command starts the web server, allowing our web application to be accessible through a web browser. Once the server is up and running, we can navigate to the specified URL to interact with our web application and control any parameters. We can now open the browser and navigate to the link <http://127.0.0.1:8000>. We will get the updated graphical interface where we can control now plus the amplitude and frequency of the sine wave the phase and the offset as in Figure 2.

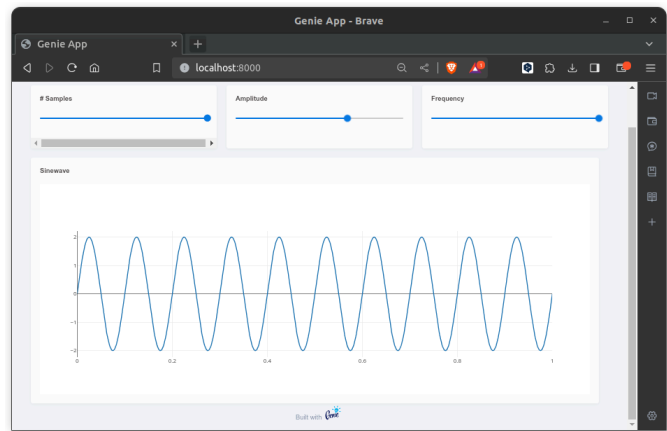


Figure 1: Genie -> Old Sine Wave



Figure 2: Genie -> Updated Sine wave