




# LAB #3: WEB APPLICATION WITH GENIE

Abdelbacet Mhamdi  
Senior-lecturer, Dept. of EE  
ISET Bizerte — Tunisia  
 a-mhamdi

Mohamed Bejaoui  
Dept. of EE  
ISET Bizerte — Tunisia  
 bejaouimohamed

Jamila Gharbi  
Dept. of EE  
ISET Bizerte — Tunisia  
 Gharbijamila

## I. INTRODUCTION

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

## II. SINE WAVE CONTROL

### A. Julia coding

To the previous app.jl file, we have add two inputs **phase** and **offset**. Its types are Float64 and Float32 and default value is 0. Also, we have add 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
    @in ph::Float64 = 0
    @in off::Float32 = 0
    @out my_sine = PlotData()
    @onchange N, amp, freq, ph, off begin
        x = range(0, 1, length=N)
        y = amp*sin.(2*pi*freq*x.+ph).+off

        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, we have add two sliders :

#### 1) The phase:

- Firstly, we have links the slider's value to a variable named **ph**.

- Secondly, we have sets the minimum value of the slider to  $-\pi$ .
- Thridly, we have sets the maximum value of the slider to  $\pi$ .
- Fourthly, we have sets the step increment of the slider to  $\frac{\pi}{100}$ .
- In the end, we specifies that labels should be displayed on the slider.

#### 2) The offset:

- Firstly, we have links the slider's value to a variable named **off**.
- Secondly, we have sets the minimum value of the slider to  $-0.5$ .
- Thridly, we have sets the maximum value of the slider to 1.
- Fourthly, we have sets the step increment of the slider to 0.1.
- In the end, we specifies 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="st-col col-12 col-sm st-module">
  <p><b>Phase</b></p>
<q-slider v-model="ph"
  :min="-3.14" :max="3.14"
  :step="0.314" :label="true">
</q-slider>
</div>
<div class="st-col col-12 col-sm st-module">
  <p><b>off</b></p>
<q-slider v-model="off"
  :min="-0.5" :max="1"
  :step="0.1" :label="true">
</q-slider>
</div>
</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>
</div>

```

### C. Graphical interface

After checking the app.jl and app.jl.html codes, we have open the terminal of vs code and open julia and tap the commands below to use the GenieFramework to develop a web application.

```

julia> using GenieFramework
julia> cd("C:/Users/bejao/OneDrive/Bureau/infodev-main/Codes/web-app")
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("C:\\Users\\bejao\\OneDrive\\Bureau\\infodev-main\\Codes\\web-app")
```

This choose 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.

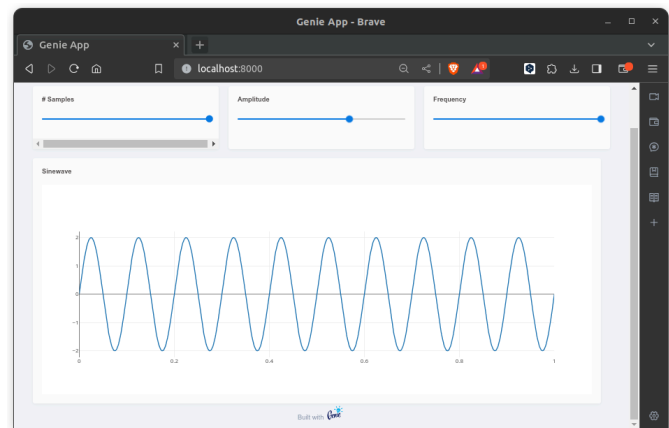


Figure 1: Genie -> Old Sine Wave

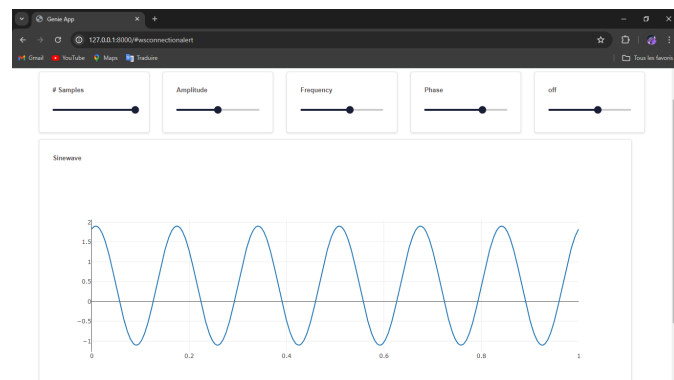


Figure 2: Genie -> Updated Sine wave