BANanoVuetifyAD3 for Dummies

Table of contents

Introduction 3
Setting up the IDE 3
Back-Ends 8
Facts 8
Tutorials 9
Part 01 9
Part 02 9
Part 03 9
Part 04 10
Part 05 10
Part 06 10
Part 07 10
Part 08 10
Part 09 10
Part 10 11
Part 11 11
Part 12 11
Part 13 11
Part 14 11
Part 15 11
Part 16 12
Part 17 12
Part 18 12
Part 19 12
Part 20 12
Part 21 13
Part 22 13

Introduction

Welcome to **BANanoVuetifyAD3 aka BVAD3**

- BANanoVuetifyAD3 is a B4J (Basic4Java) library that helps one create Web Apps / Web Sites using BANano and Vuetify. It is the first VueJS UX based library for BANano. This library is created by Anele Mbanga. It seeks to bring the power of Vuetify to BANano. With BVAD3 one codes their UX to life and then using BANano functionalities, one can build and then publish their web application.

- B4J is created by Anywhere Software. With it one is able to write Java applications in B4X, a VB (Visual Basic) like syntax code base and it produces native java apps that can run on Windows, Linux and Mac.

- BANano is created by Alain Bailleul. This helps anyone create websites and or webapps using VB-syntax. It generates pure JavaScript, CSS and HTML for the website/webapp. Apps created with it are SPAs (Single Page Applications) and or PWA(Progressive Web Apps) with the inclusion of web service workers (optional). BANano itself is UX framework independent and this means one can use their own framework of choice.

- Vuetify is a complete UI framework built on top of Vue.js. The goal of the project is to provide developers with the tools they need to build rich and engaging user experiences. Unlike other frameworks, Vuetify is designed from the ground up to be easy to learn and rewarding to master with hundreds of carefully crafted components from the Material Design specification

Things to remember

- BANanoVuetifyAD3 = **BVAD3**
- Single Page Applications = **SPA**
- Progressive Web App = **PWA**
- Basic4Java = **B4J**
- Visual Basic **= VB**

Things about Anywhere Software

1. There is a video based learning channel. Check it here.
2. With their B4A (Basic4Android) Now FREE, one can create completely native Android Apps.
3. With their B4i (Basic4iPhone), one can create completely native iPhone, iPad Apps
4. There are some coding booklets that have been written that can help you out with the B4X eco-system, thanks to Klaus.

Let us prepare our development environment first

Setting up the IDE

Developing on Windows PC

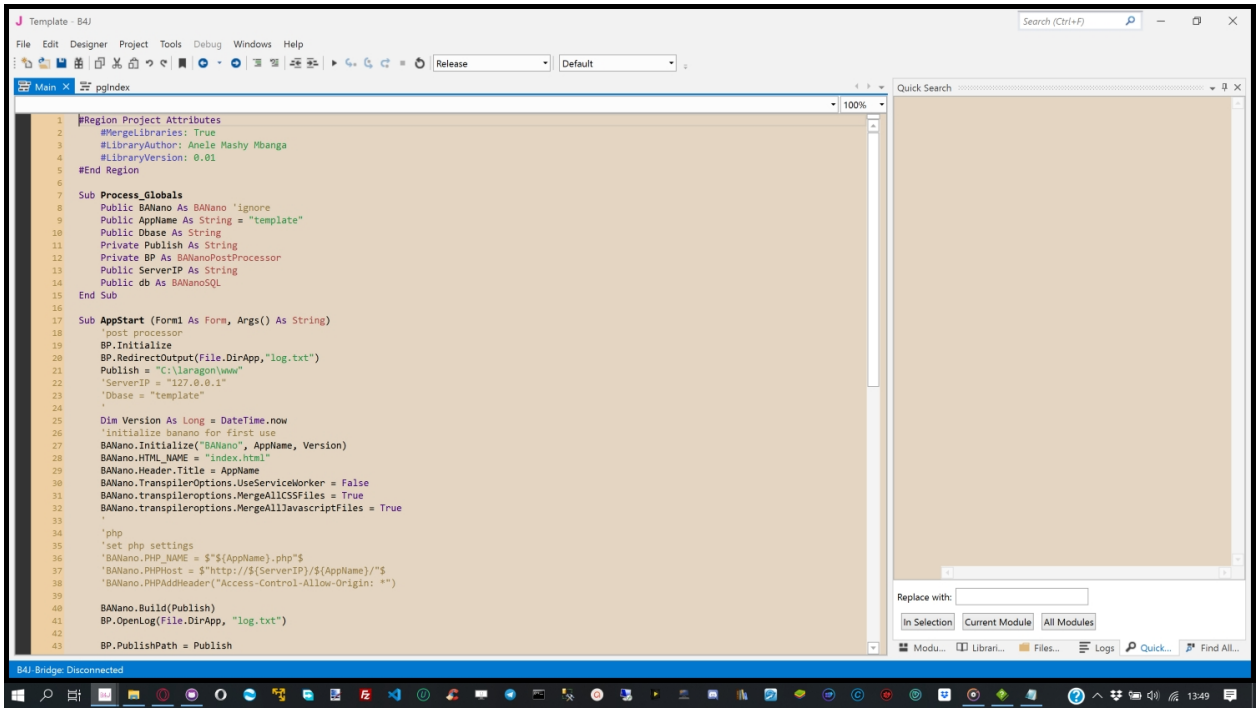
At the time of writing B4J only runs on Windows PC.

To be able to develop BVAD3 apps, you will need the following: You can click on each to download

1. B4J

There are instructions on the website on how to install and configure B4J. With it you will need Java JDK

8+. Just follow the instructions on how to set up your IDE and get it ready.



Create a folder structure: You can skip this step if your IDE is already set up

1. Create a folder named B4X in your C: drive, and then create the respective sub folders

C:\B4X\B4J\Shared - we will call this folder "**shared**"
C:\B4X\B4J\Libraries - we will call this folder "**external libraries**"
C:\B4X\B4J\Workspace - we will call this "**workspace**"

- The **shared** folder will store all code modules that have sharable code
- The **external libraries** will store all libraries from others users e.g. BANano & BVAD3 library.
- The **workspace** will store your project folders, e.g. projects we will create with BVM

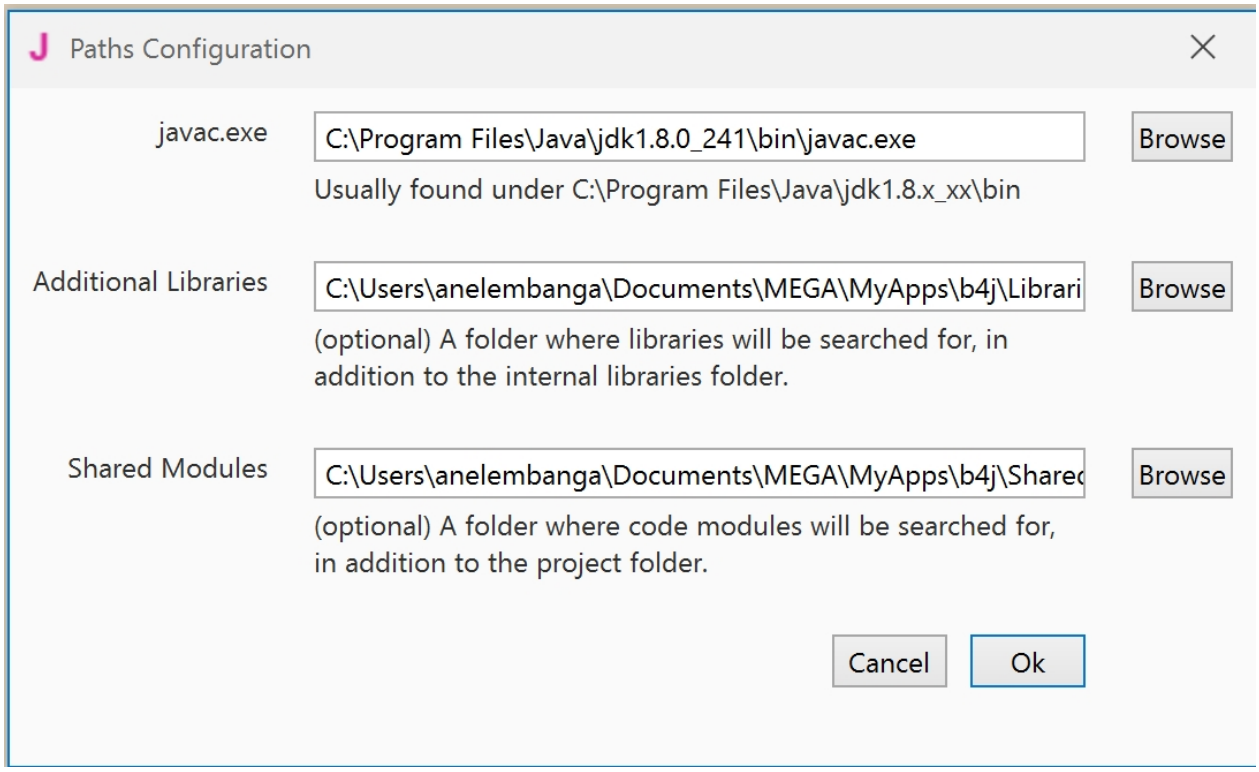
We have B4X master folder because we can create the same structure for **B4A** and or **B4I** IDEs.

Test the readiness of your IDE

To test the readiness of your IDE, we will do 3 things.

1. Start B4J, in the menu click Tools > Configure Paths. A screen like this will appear.

Figure 2



- Ensure that the specified paths point to the correct locations.
- Additional Libraries should point to your **external libraries** path you created before
- Shared Modules should point to the **shared** path you created before. You click Ok to save any changed details.

2. BANano

Once downloaded, copy the contents of the **Library** folder to your external libraries folder e.g.C:\B4X\B4J\Libraries This library comes with some code examples on the usage of BANano. I have also written a nice tutorial on how one can get started with BANano. That will help you with the basics and also further experience on how to use BANano. As an example, one of the things you will see when writing BVAD3 code is the ***BANanoEvent***.

NB: I greatly recommend that you go through this tutorial so that at least you have some understanding of BANano and what it does.

3. BANanoVuetifyAD3

Download the github repo and extract the contents to your working folder, e.g. C:\B4X\B4J\Workspace Open the Library folders inside BVAD3, double click the BANanoVuetifyAD3.b4j file to open b4j. Run the project. This depends on #2 above. Close the project. Your library will be compiled.

The structure of the BVAD3 github repo.

1. Library - this contains the source code for the BVAD3 b4x library.

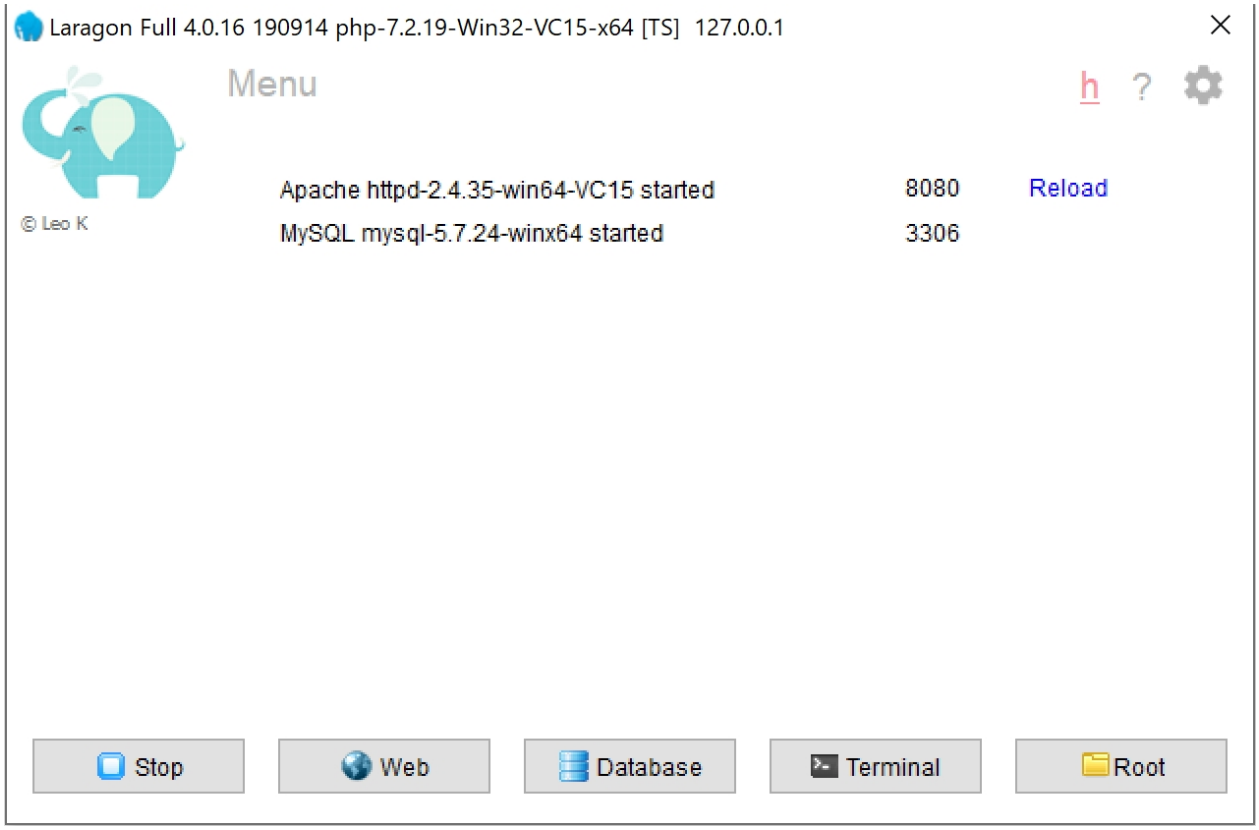
2. Demos - a collection of demo projects created with BVAD3

3. Templates - various BVAD3 templates

I am assuming you have briefed yourself about BANano (my tutorial and others) and now you are ready to explore BVAD3 code and its output. Going forward we will use our VB know how to create apps.

4. A webserver. I am using the laragon development web server for all my examples here. One can also use XAMPP

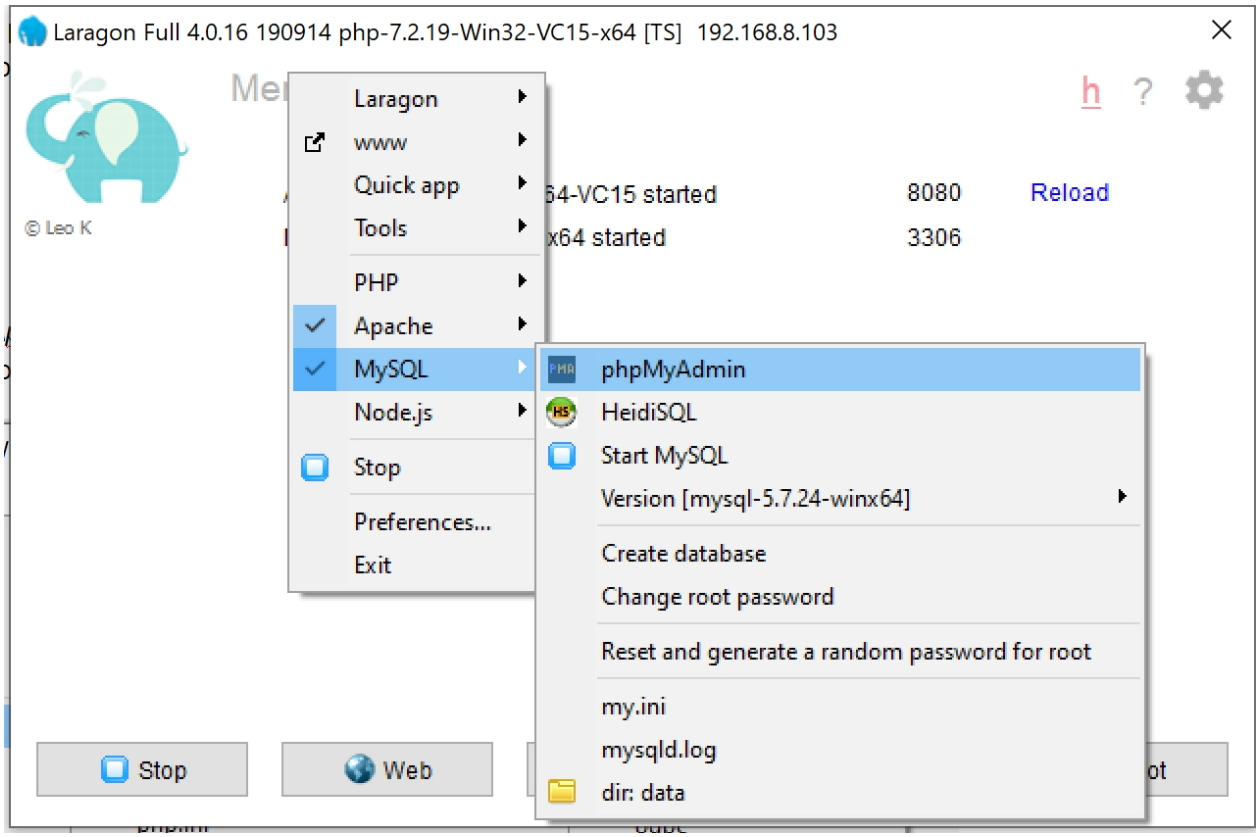
I am yet to test the USBWebServer.



MySQL Usage

- Laragon does not come installed with phpMyAdmin, thus, download phpMyAdmin
- Extract the folder to c:\laragon\etc\apps**phpMyAdmin**
- The password is ***root***.

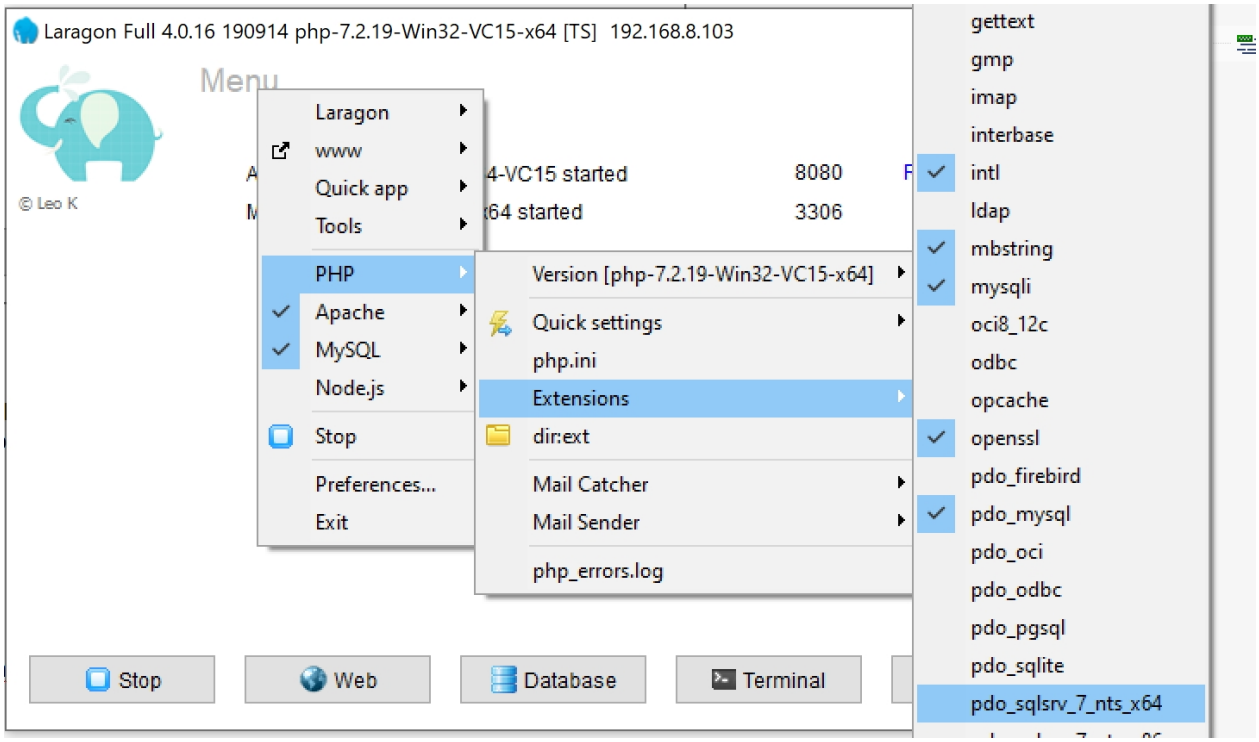
Check that phpMyAdmin works



MSSQL Usage

[Install PHP drivers for MSSQL]

- Extract the files to *C:\laragon\bin\php\php-7.2.19-Win32-VC15-x64\ext*. This is the php extensions folder
- Activate the nts (non-thread-safe option)



Internet Information Server

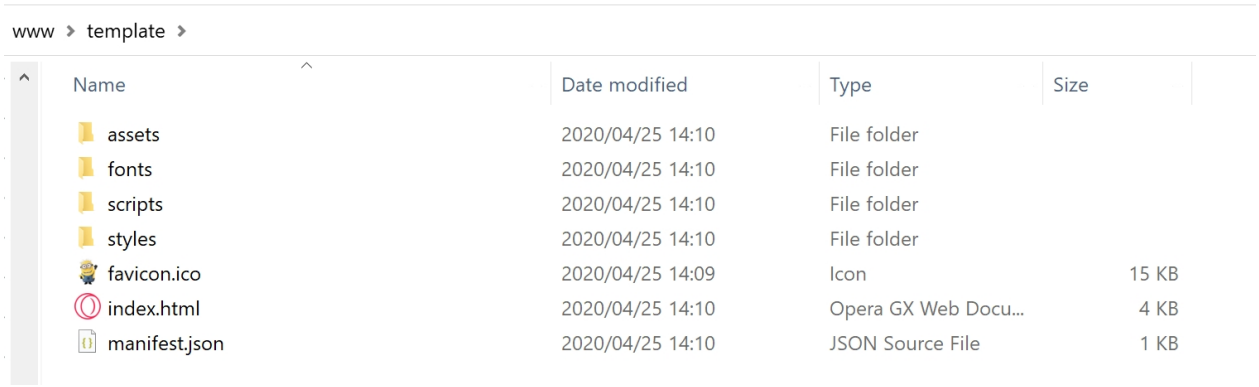
1. Install Web Platform Installer
2. Install PHP Manager

Install IIS from WPI.

2.5. Install an FTP tool

I am using FileZilla to upload my BVM apps to the interweb. The output of your website, will be saved to the folder that you told banano to publish on.

This structure will follow this pattern.



Explaining the folders

1. assets - this stores all assets for the app e.g. images, json, and other files
2. fonts - (optional for storing font fies)
3. scripts - this folder has all your .js files
4. styles - this folder has all your .css files

Back-Ends

By default, when creating apps with the designer, BANanoSQL is the default backend. You can change your backend so that your app works with:

1. BANAnoSQL (IndexedDB via AlaSQL)
2. SQLite
3. MySQL
4. MSSQL
5. FireBase

For the first 4, we have created a library called BANanoVueConnect and fore Firebase we have created a library called BANanoFireStoreDB. You can check the MealPrep demo project on how Firebase storage was used.

Facts

Important things to know:

- Referencing state in BVAD3 should always be in lowercase e.g. {{ antotherone }} and DO NOT use {{ anotherOne }}

- States CANNOT be hyphened e.g. "my-name" should be "myname"
- States CANNOT have spaces or special characters
- **VModel** should not have spaces.

A good example would be

Set the state...

```
Dim items As List = vuetify.NewList
items.Add("Anele Mbanga (Mashy)")
vuetify.SetData("items", items)
```

Get the state

```
Dim items As List = vuetify.GetData("items")
Log(items)
```


Tutorials

Our blog

Part 01

Youtube Link

Source Code

This is a skeleton project

Part 02

Youtube Link

Source Code

Adding navigation bar, hamburger, spacer, and a button. Binding abstract designer components to vuetify app.
Firing events.

Part 03

Youtube Link

Source Code

Using individual blocks, we build and run our app. We create a dynamic title for our toolbar title and change this on button click by updating its state.

Part 04

Youtube Link

Source Code

We start with a blank template and create routers, load layouts to the router components and link these to the vuetify app. For more details of how routers are used, see the MealPrep app in the Demos folder.

Part 05

Youtube Link

Source Code

Based on our previous example, we create a menu that is activated by a button link. We also set an active class for each menu item being selected.
We apply a transition to the menu and link routers per menu item. We use state binding for the menu items.

Part 06

Youtube Link

Source Code

We continue from part 5 and add a logout button and an empty navigation drawer with a background image. We will add navigation items in part 7.

Part 07

Youtube Link

Source Code

We continue from part 6 and add a a list to the drawer, use the same links we created the menu items with. We then use a v-for loop and binding to ensure each drawer item can navigate to its page.

Part 08

Youtube Link

Part 09

Youtube Link

Source Code

In this part we create 3 types of avatars, text, icon and image. We also create a grid layout to set these at row 1, column 1 to 3 respectively.
We also add a user profile just above the list in the navigation drawer

Part 10

Youtube Link

Source Code

We create different alerts and toggle visibility

Part 11

Youtube Link

Source Code

We create dynamic dialogs and dynamic snackbar controls.

Part 12

Youtube Link

Source Code

We create an input dialog prompt.

Part 13

Youtube Link

Source Code

We create text-field layouts and feed these to the grid layout we have created. As we set v-models for each of the text-fields we call .GetData to read the values of the text fields.

Part 14

Youtube Link

Source Code

We create badges and increment and decrement these also changing their color. We also create a user status indicator.

Part 15

Youtube Link

Source Code

Instead of creating multiple layouts, we re-use our layouts and use BANanoLoadLayoutArray to load, extract and update them via code.

Part 16

Youtube Link

Source Code

In this example, we have created date and time pickers for input. Both are placed inside a menu so that they are activated when a text field is active.

Part 17

Youtube Link

Source Code

We start our journey with v-data-table

Part 18

Youtube Link

Source Code

We extend our tables and add color coded chips and color coded action buttons. We link these action buttons to events, events that are passed the row being processed.

Part 19

Youtube Link

Source Code

Here we add interactive user input components to the v-data-table, these are switches, rating, progress indicators and are able to display an avatar.

Part 20

Youtube Link

Source Code

We add lazy loaded images to the table, mail to links, colored icons, and format dates and numbers with day.js and numeral.js.

Part 21

Youtube Link

Source Code

We also obfuscate our javascript files for the project for protection (run in release mode)

Part 22

Youtube Link

Source Code

We build the grid using the abstract designer during debug to experience BANano #LiveCodeSwapping.
We add elements to the grid matrix using re-usable layouts, this includes a button with a click event.