

Migrating to SithasoDaisy2

Power to Choose Edition



SithasoDaisy
Anything is possible

Table of contents

Introduction	3
Web Crawling	7
String Compression	8
Breaking Changes	8
DataBases	9
Useful Links	10

Introduction

Welcome to SithasoDaisy 2 - Power to Choose Edition.

The Sithaso 2 Power to Choose Edition is an upgrade of version 1, however unlike version 1, everything in Version 2 follows a modular approach, where you choose which components to use. These components have been made as b4xlibs, specific to the **Plugins**, the **DataSource** and the database back-end functionalities.

To include the individual b4xlibs, just copy them to your additional libraries folder.

The additional b4xlibs are all prefixed with SithasoDaisy and most address the plugins available for SithasoDaisy. These are the following:

- AES
- AES4PHP
- AlaSQLE
- Axios
- BarCodeReader
- ChartKick
- CollectJS
- CSV
- DataSource
- Device
- Docs
- DocxTemplator
- EnjoyHint
- EvoCalendar
- Excel
- FireBase
- FlatPicker
- FlipPage
- FluidMeter
- FontAwesome
- FrappeGantt
- FullCalendar
- GMap
- GridNav
- HtmlParser
- JQuery
- JustGage
- KanBan
- LottiePlayer
- LZString
- MockCode
- MongoDB
- MSAccess
- MSSQLE
- MySQLAPI
- MySQLLE
- PDF
- PHP

- PocketBase
- PostGres
- QRCode
- Relax
- RollDate
- SignaturePad
- Swal
- Table
- TimeLine
- ToastChart
- Gijgo (TreeView & Editor)
- UploadToPHP
- WebCam

All of these libraries css & js files are marked for **later** loading. This means to use them, one needs to call the respective calls to load the css & js files. This has been done on the demo.

The calls to load the files you need can be done on pgIndex.Initialize method o on the Show sub of each page.

One should include the Uses clause for the components needed on that page. The calls ae the following:

- BANano.Await(app.UsesEnjoyHint)
- BANano.Await(app.UsesFlipPage)
- BANano.Await(app.UsesTimeLine)
- BANano.Await(app.UsesGridNav)
- BANano.Await(app.UsesChartKick)
- BANano.Await(app.UsesQRCode)
- BANano.Await(app.UsesBarCodeReader)
- BANano.Await(app.UsesWebCam)
- BANano.Await(app.UsesAxios)
- BANano.Await(app.UsesFlatPickDateTime)
- BANano.Await(app.UsesLottiePlayer)
- BANano.Await(app.UsesRollDate)
- BANano.Await(app.UsesHTMLParser)
- BANano.Await(app.UsesLZCompressString)
- BANano.Await(app.UsesTable)
- BANano.Await(app.UsesPocketBase)
- BANano.Await(app.UsesFullCalendar)
- BANano.Await(app.UsesSignaturePad)
- BANano.Await(app.UsesCSV)
- BANano.Await(app.UsesDevice)
- BANano.Await(app.UsesKanBan)
- BANano.Await(app.UsesAES)
- BANano.Await(app.UsesAESPHP)
- BANano.Await(app.UsesEncryption)
- BANano.Await(app.UsesToastChart)
- BANano.Await(app.UsesMockupCode)
- BANano.Await(app.UsesPDF)
- Banano.Await(app.UsesDocxTemplator)
- BANano.Await(app.UsesExcel)
- BANano.Await(app.UsesFireBase)
- BANano.Await(app.UsesSupaBase)

- BANano.Await(app.UsesRelax)
- BANano.Await(app.UsesGijgo)
- BANano.Await(app.UsesJustGage)
- BANano.Await(app.UsesFluidMeter)
- BANano.Await(app.UsesFrappeGantt)
- BANano.Await(app.UsesEvoCalendar)
- BANano.Await(app.UsesDropZone)
- BANano.Await(app.UsesCollectJS)

For example in the demoADTable code module, we have a SDUITable, we will also execute export functionality, so our code for Sub Show is:

```

20 'sub to show the page
21 Sub Show(duiapp As SDUIApp) 'ignore
22 'get the reference to the app
23 app = duiapp
24 app.PagePause
25 banana.Await(app.UsesEnjoyHint)
26 banana.Await(app.UsesTable)
27 banana.Await(app.UsesCSV)
28 banana.Await(app.UsesExcel)
29 banana.Await(app.UsesPDF)
30 banana.LoadLayout(app.PageViewer, "adtablelayout")
31 pgIndex.ShowDesign
32 pgIndex.ClearDesign
33 pgIndex.AddDesignKV("", "--Nothing Selected--")
34 pgIndex.AddDesignKV1("SDUIPage.page", "page", page.CustProps)
35 pgIndex.AddDesignKV1("SDUITable.SDUITable1", "SDUITable1", SDUITable1.CustProps)
36
37 'build the page, via code or loadlayouts
38 BuildPage
39
40 End Sub

```

NB: These Uses calls should be done after the **app =** assignment.

As you might have noticed, the Uses names are closely linked to the component names, e.g. UsesEvoCalendar ~ SDUIEvoCalendar etc. So one needs to apply the Uses load call for the component used on that page. Some Uses calls though do not apply to components but to helper calls. For example AES, which the js files are used for these methods.

```

55 Private Sub txtOriginal_Append_Click (event As BANanoEvent)
56     txtEncrypted.Value = SDUIShared.AESEncrypt(txtOriginal.Value, "SithasoDaisy")
57 End Sub
58
59 Private Sub txtEncrypted_Append_Click (event As BANanoEvent)
60     txtDecrypted.Value = SDUIShared.AESDecrypt(txtEncrypted.Value, "SithasoDaisy")
61 End Sub

```

The aim is for the developer to choose which b4xlibs to use when developing their apps, and leave those ones not used. For example in the Demo Example, we have used all the b4xlibs available for SithasoDaisy, so we select all of them.

<input checked="" type="checkbox"/>	SithasoDaisy	2.00	
<input checked="" type="checkbox"/>	SithasoDaisyAES	2.00	
<input checked="" type="checkbox"/>	SithasoDaisyAES4PHP	2.00	
<input checked="" type="checkbox"/>	SithasoDaisyAlaSQLE	2.00	
<input checked="" type="checkbox"/>	SithasoDaisyAxios	2.00	
<input checked="" type="checkbox"/>	SithasoDaisyBarCodeReader	2.00	
<input checked="" type="checkbox"/>	SithasoDaisyChartKick	2.00	
<input checked="" type="checkbox"/>	SithasoDaisyCollectJS	2.00	
<input checked="" type="checkbox"/>	SithasoDaisyCSV	2.00	
<input checked="" type="checkbox"/>	SithasoDaisyDataSource	2.00	
<input checked="" type="checkbox"/>	SithasoDaisyDevice	2.00	
<input checked="" type="checkbox"/>	SithasoDaisyDocs	2.00	
<input checked="" type="checkbox"/>	SithasoDaisyDocxTemplator	2.00	
<input checked="" type="checkbox"/>	SithasoDaisyEditor	2.00	
<input checked="" type="checkbox"/>	SithasoDaisyEnjoyHint	2.00	
<input checked="" type="checkbox"/>	SithasoDaisyEvoCalendar	2.00	
<input checked="" type="checkbox"/>	SithasoDaisyExcel	2.00	
<input checked="" type="checkbox"/>	SithasoDaisyFireBase	2.00	
<input checked="" type="checkbox"/>	SithasoDaisyFlatPicker	2.00	
<input checked="" type="checkbox"/>	SithasoDaisyFlipPage	2.00	
<input checked="" type="checkbox"/>	SithasoDaisyFluidMeter	2.00	
<input checked="" type="checkbox"/>	SithasoDaisyFontAwesome	2.00	
<input checked="" type="checkbox"/>	SithasoDaisyFrappeGantt	2.00	
<input checked="" type="checkbox"/>	SithasoDaisyFullCalendar	2.00	
<input checked="" type="checkbox"/>	SithasoDaisyGMap	2.00	

Upgrading your Existing Project

1. In **pgIndex.Initialize**, find this code:

```
app.AddApp(Me, Main.AppName)
```

2. Add the following code

```
BANano.Await(app.UsesPocketBase)
BANano.Await(app.UsesTable)
BANano.Await(app.UsesFlatPickDateTime)
BANano.Await(app.UsesRollDate)
BANano.Await(app.UsesExcel)
BANano.Await(app.UsesPDF)
```

3. Ensure that the following b4xlibs are selected

<input checked="" type="checkbox"/>	SithasoDaisy	2.00
<input checked="" type="checkbox"/>	SithasoDaisyExcel	2.00
<input checked="" type="checkbox"/>	SithasoDaisyFlatPicker	2.00
<input checked="" type="checkbox"/>	SithasoDaisyFontAwesome	2.00
<input checked="" type="checkbox"/>	SithasoDaisyJQuery	2.00
<input checked="" type="checkbox"/>	SithasoDaisyRollDate	2.00
<input checked="" type="checkbox"/>	SithasoDaisySwal	2.00
<input checked="" type="checkbox"/>	SithasoDaisyTable	2.00
<input checked="" type="checkbox"/>	SithasoDaisyUploadToPHP	2.00

You can add / remove the **Uses...** code depending on what you need. The FontAwesome, Swal and JQuery are compulsory and should be selected. You can add / remove other b4xlibs as you wish.

Below we will just touch on some few items (already in the demo), just to explain the concepts.

Web Crawling

To run a web-crawler on you page do the following:

1. Add this on Sub Show

```
BANano.Await(app.UsesHTMLParser)
```

2. We have a button click event to crawl the page:

```

226 Sub btncode_click (e As BANanoEvent)
227     'get the elemnt html
228     Dim allCode As String = BANano.GetElement("#pageview").GetHTML
229     'execute web scraping
230     Dim treeItems As List = app.parseHTML(allCode)
231     Log(treeItems)
232 End Sub

```

String Compression

We need to compress strings in our page:

We add this on Sub Show

```
banano.Await(app.UsesLZCompressString)
```

Our code to compress the string is.

```

Private Sub txtMessage_Append_Click (event As BANanoEvent)
    If txtMessage.IsBlank Then Return
    txtConvert.Value = ""
    txtMessage.AppendButton.Loading = True
    Dim res As String = app.LZCompressString(txtMessage.Value)
    txtConvert.Value = res
    txtConvert.HintCaption = $"Length: ${txtConvert.Count}"$
    txtMessage.HintCaption = $"Length: ${txtMessage.Count}"$
    txtMessage.AppendButton.Loading = False
End Sub

```

Breaking Changes

Below are way to add a few components (plugins) that has changed when adding components via code.

1. SDUIBarCodeReader

```

Private bcScanner As SDUIBarCodeReader
bcScanner.AddBarCodeReader(Me, page.Cell(2, 1).Name, "bcreader1")

```

2. SDUIFullCalendar

```

Private fc As SDUIFullCalendar
fc.AddFullCalendar(Me, page.Cell(1, 1).Name, "fc")

```

3. SDUIQRCode

```

Private qrcode1 As SDUIQRCode
qrcode1.AddQRCode(Me, page.Cell(3, 1).Name, "qrcode1")

```

4. SDUISignaturePad

```

Private signp As SDUISignaturePad
signp.Initialize(Me, "signp", "signp")
signp.AddToParent(page.Cell(1, 1).Name, CreateMap())

```


5. SDUITimeLine

```
Dim bt1 As SDUITimeLine
bt1.AddBeautifulTimeLine(Me, page.Cell(1, 1).Name, "bt1")
```

```
Dim bt2 As SDUITimeLine
bt2.AddTimeLine1(Me, page.Cell(3, 1).Name, "bt2")
```

6. SDUIWebcam

```
Private bcScanner As SDUIWebcam
bcScanner.AddWebCam(Me, page.Cell(1, 1).Name, "webcam1")
```

7. SDUILottiePlayer

```
Dim lp As SDUILottiePlayer
lp.AddLottiePlayer(Me, page.Cell(1, 1).Name, "lp1")
lp.JsonUrl = "/assets/liftoff.json"
lp.Width = "250px"
lp.Height = "250px"
lp.AutoPlay = False
lp.IsLoop = True
lp.Root.m(4)
```

8. Exporting Table To Excel (SDUITable1 is the table component)

```
Dim excel As SDUIExcel
excel.Initialize(Me, "SDUITable1", SDUIShared.DateTimeNowBackUp & " " & SDUITable1.ExcelTitle & ".xlsx")
excel.ExportTable(SDUITable1.ExcelTitle, True, SDUITable1.GetFieldNames, SDUITable1.GetTitleNames, SDUITable1.Originals)
```

9. Exporting Table To PDF (SDUITable1 is the table component)

```
Dim pdf As SDUIJSPDF
pdf.Initialize(Me, "SDUITable1", SDUIShared.DateTimeNowBackUp & " " & SDUITable1.ExcelTitle & ".pdf")
pdf.ExportTable(SDUITable1.ExcelTitle, SDUITable1.Orientation, SDUITable1.PageSize, _
SDUITable1.Unit, SDUITable1.FontSize, SDUITable1.PdfTheme, SDUITable1.PageBreakRepeat, _
SDUITable1.GetFieldNames, SDUITable1.GetTitleNames, SDUITable1.Originals)
```

DataBases

The DataSource has been separated to be standing on its own. The only example that was included was the MySQL example in PHP (from your downloads)

All BackEnd helpers can be used individually without having to use the DataSource. You can read about the DS here.

<https://www.b4x.com/android/forum/threads/bananodatasource-a-crud-database-abstract-layer.131838/#content>

How each of these connections works was discussed here:

<https://www.b4x.com/android/forum/threads/bananoconnect-bananosql-sqlite-mysql-mssql-library.117956/#content>

PocketBase (the most used back-end in our apps)

<https://www.b4x.com/android/forum/threads/sithasodaisy-pockethost-vs-pocketbase-what-is-the-difference.148439/#content>

<https://www.b4x.com/android/forum/threads/banano-using-pocketbase-firebase-alternative-for-your-apps.143589/#content>

<https://www.b4x.com/android/forum/threads/pocketbase-crud-rest-api-sse-with-postman-and-then-httputils.144227/#content>

<https://www.b4x.com/android/forum/threads/sithasodaisy-tailwindcss-pocketbase-chatroom-using-sse.144720/#content>

<https://www.b4x.com/android/forum/threads/sithasodaisy-crud-webapp-on-java-jetty-webserver-with-pocketbase-back-end-hosted-on-pockethost-io.147680/#content>

SupaBase

<https://www.b4x.com/android/forum/threads/banano-supabase-firebase-alternative-user-signup-signin-process.144850/#content>

<https://www.b4x.com/android/forum/threads/banano-supabase-firebase-alternative-crud-example-with-sse.144817/#content>

You can connect to:

- FireBase
- MySQL PHP (crud web app already attached)
- MySQL PHP API
- MSSQL
- MSAccess
- SQLite

Apps featuring these back-ends will be made available soon.

CouchDB (SDUIRelax)

<https://www.b4x.com/android/forum/threads/bananorelax-1-perform-crud-offline-2-sync-to-cloud-and-3-relax.134767/#content>

Key-Value Store

<https://www.b4x.com/android/forum/threads/bananokeystore-a-keyvalue-store-for-banano-on-top-of-indexeddb.141816/#content>

Useful Links

Just to refresh, here are some useful links

SithasoDaisy Forum Link:

<https://www.b4x.com/android/forum/threads/sithasodaisy-tailwindcss-daisyui-toolbox.143549/>

SithasoDaisy Library Link

<https://www.b4x.com/android/forum/threads/sithasodaisy-tailwindcss-webapps-using-b4x.143950/>

SithasoDaisy Learn Link (the most up to date source is on the demo app)

<https://www.b4x.com/android/forum/threads/sithasodaisy-mashy-teaches-tailwindcss-using-b4x-with-ebook.143971/>

SithasoDaisy WebApps

<https://www.b4x.com/android/forum/threads/sithasodaisy-webapps-5-each.145871/#content>

Buy SithasoDaisy

<https://sithasoholdings.gumroad.com/l/wewsu>