

```

-----
-- calltab.lua
-- by Stewart Bracken http://stewart.bracken.bz stew.bracken@gmail.com
-- Call any number with your twilio phone number
-- Lots of code leveraged from WidgetDemo.
-- Known critical bug: text labels and buttons are recreated on every scene
-- enter but never destroyed, adding up a lot of widgets and whatnot.
-----

local widget = require( "widget" )
local storyboard = require( "storyboard" )
local scene = storyboard.newScene()
local Util = require("Twilio.Util")

-- create a constant for the left spacing of the row content
local LEFT_PADDING = 10
local textMode = false

local toTextField = nil
local fromTextField = nil

local tHeight = 30
local tLeft = 80
local tWidth = 200
local tTop = 80

local function createTextFields(self)
    local function fieldHandler( event )
        if ( "began" == event.phase ) then
            -- This is the "keyboard has appeared" event
            -- In some cases you may want to adjust the interface when the keyboard
            -- appears.

            -- Show Dismiss Keyboard button if in portrait mode
            if isPortrait then
                clrKbButton.isVisible = true
            end

            textMode = true

        elseif ( "ended" == event.phase ) then
            -- This event is called when the user stops editing a field: for
            -- example, when they touch a different field

        elseif ( "submitted" == event.phase ) then

```

```

-- This event occurs when the user presses the "return" key (if
-- available) on the onscreen keyboard

-- Hide keyboard
native.setKeyboardFocus( nil )
textMode = false
clrKbButton.isVisible = false -- Hide the Dismiss KB button
end
end

toTextField = native.newTextField(tLeft,tTop,tWidth, tHeight)
toTextField.inputType = "number"
toTextField.text = CALLER_TO
toTextField:addEventListener("userInput",fieldHandler)
self.view:insert(toTextField)

fromTextField = native.newTextField(tLeft,tTop+tHeight+10,tWidth, tHeight)
fromTextField.inputType = "number"
fromTextField.text = CALLER_FROM
fromTextField:addEventListener("userInput",fieldHandler)
self.view:insert( fromTextField )
end

function scene:enterScene(e)
    if fromTextField == nil then
        createTextFields(self)
    end
end

function scene:exitScene(e)
    fromTextField:removeSelf()
    toTextField:removeSelf()
    fromTextField=nil
    toTextField=nil
end

function scene:createScene( event )

    local group = self.view

    -- Display a background
    local background = display.newImage( "assets/background.png", true )
    group:insert( background )

    -- Status text box

```

```

local statusBox = display.newRect( 70, 290, 210, 120 )
statusBox:setFillColor( 0, 0, 0 )
statusBox.alpha = 0.4
group:insert( statusBox )

-- Status text
local statusText = display.newText( "Enter Twilio validated numbers to begin",
80, 300, 200, 0, native.systemFont, 20 )
statusText.x = statusBox.x
statusText.y = statusBox.y - ( statusBox.contentHeight * 0.5 ) + (
statusText.contentHeight * 0.5 )
group:insert( statusText )

-----
-----

-- widget.newSpinner()
-----

-- Create a spinner widget
local spinner = widget.newSpinner
{
    left = 274,
    top = 55,
}
group:insert( spinner )
spinner.isVisible = false

createTextFields(self)

local toTextFieldLabel = display.newText( "To:", LEFT_PADDING, 80,
native.systemFont, 16 )
toTextFieldLabel:setTextColor( 0 )
group:insert( toTextFieldLabel )

local fromTextFieldLabel = display.newText( "From:", LEFT_PADDING,
tTop+tHeight+10, native.systemFont, 16 )
fromTextFieldLabel:setTextColor( 0 )
group:insert( fromTextFieldLabel )

local function makeCall( event )
    spinner:start()
    spinner.isVisible = true
    statusText.text = "Requesting call..."

```

```

local function request_listener(e)
    spinner:stop()
    spinner.isVisible=false
    if e.success then
        statusText.text = "Call success!"
    else
        statusText.text = e.message
    end
    print(Util.to_string(e.response))
end

local vars = {Type="Calls", To=toTextField.text, From = fromTextField.text,
Url="http://demo.twilio.com/docs/voice.xml" }
R:request(vars, "POST", request_listener)
end

local callButton = widget.newButton
{
    left = tLeft,
    top = tTop+tHeight*2+10,
    width = tWidth,
    height = tHeight,
    id = "callButton",
    label = "CALL",
    onPress = makeCall,
}
group:insert( callButton )

end

scene:addEventListener( "createScene" )
scene:addEventListener( "enterScene" )
scene:addEventListener( "exitScene" )

return scene

```