```lua
--loginwindow.lua
--Login window works but I haven't completed it, ignore this file.

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


local TwilioRestClient = require "Twilio.TwilioRestClient"
local auth = require "tests.auth" or {}
--Plug in your twilio account credentials here in the XXXX's
local ACCOUNT_SID = auth.ACCOUNT_SID or "XXXXXXXX" -- your Account SID
local ACCOUNT_TOKEN = auth.ACCOUNT_TOKEN or "XXXXXX" --your account token

--Outgoing Caller phone number
local CALLER_TO = auth.CALLER_TO or "+1NNNNNNNNNN"
--Incoming Caller Phone Number, previously validated with Twilio
local CALLER_FROM = auth.CALLER_FROM or "+1NNNNNNNNNN"

--Lets just make our TwilioRestClient global because it's easier to use across scenes
R = TwilioRestClient.create(ACCOUNT_SID, ACCOUNT_TOKEN)


local Util = require("Twilio.Util")

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

function scene:exitScene(event)
    --storyboard.purgeScene(self)
end

function scene:destroyScene(e)
    storyboard.purgeScene("loginwindow")
end


function scene:createScene( event )
    local group = self.view

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

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

    local sidTextField = native.newTextField(tLeft,tTop,tWidth, tHeight)
    --toTextField.inputType = "number"
    sidTextField.text = auth.ACCOUNT_SID
    sidTextField:addEventListener("userInput",fieldHandler)
    local toTextFieldLabel = display.newText( "SID:", LEFT_PADDING, 80,
native.systemFont, 16 )
    sidTextField:setTextColor( 0 )
    group:insert( sidTextField )

    local tokenTextField = native.newTextField(tLeft,tTop+tHeight+10,tWidth,
tHeight)
```

```
    --fromTextField.inputType = "number"
    tokenTextField.text = auth.ACCOUNT_TOKEN
    tokenTextField:addEventListener("userInput",fieldHandler)
    local fromTextFieldLabel = display.newText( "Token:", LEFT_PADDING,
tTop+tHeight+10, native.systemFont, 16 )
    tokenTextField:setTextColor( 0 )
    group:insert( tokenTextField )



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

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

    -- Start the spinner animating
    --spinner:start()
    spinner.isVisible = false


    local function createTabBar()


    end


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

        local function request_listener(e)
            spinner:stop()
            spinner.isVisible=false
            if e.success then
                --login
                group.isVisible=false
                createTabBar()
                storyboard.gotoScene( "calltab" )
            end
        end

        --If twilio accepts get request, credentials are valid and we login
        local vars = {}
        R:request(vars, "GET", request_listener)
    end

    local loginButton = widget.newButton
    {
        left = tLeft,
        top = tTop+tHeight*2+10,
        width = tWidth,
        height = tHeight,
        id = "loginButton",
        label = "Login",
        onRelease = authenticate,
    }
    group:insert( loginButton )

end

scene:addEventListener( "createScene" )

scene:addEventListener( "destroyScene" )
return scene
```