```lua
--------------------------------------------------------------------------
-- sendsms.lua
-- by Stewart Bracken  http://stewart.bracken.bz  stew.bracken@gmail.com
-- Send text message with rest api.
-- 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 bodyTextField = 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
```

```lua
        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 )

    bodyTextField = native.newTextField(tLeft,tTop+(tHeight+10)*2,tWidth, tHeight*2)
    --bodyTextField.inputType = "number"
    bodyTextField.text = "Your message goes here"
    bodyTextField:addEventListener("userInput",fieldHandler)
    self.view:insert( bodyTextField )
end

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

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

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

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

    createTextFields(self)

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

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

    local smsTextFieldLabel = display.newText( "Message:", LEFT_PADDING,
tTop+(tHeight+10)*2, native.systemFont, 16 )
    smsTextFieldLabel:setTextColor( 0 )
    group:insert( smsTextFieldLabel )


    local function sendSMS( event )
        spinner:start()
        spinner.isVisible = true
        statusText.text = "Sending sms..."

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

        --Send a message
        local vars = {Type="Messages", From=fromTextField.text, To =
toTextField.text, Body=bodyTextField.text}
        R:request(vars,"POST", request_listener)
    end

    local sendButton = widget.newButton
    {
        left = tLeft,
        top = tTop+(tHeight+10)*4,
        width = tWidth,
        height = tHeight,
        id = "sendButton",
        label = "Send SMS",
        onRelease = sendSMS,
    }
```

```lua
        group:insert( sendButton )

    end


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

    return scene
```