

Lua Directory example

About

Takes input of the first 3 chars of a FreeSWITCH user's last name and lets the caller choose which returned result to connect. You will need [mod_lua](#) working for this.

Example

```
-- directory.lua
-- written by Brian Snipes <bsnipes@_snipes.org>
--   on 2008/07/03
-- Purpose
-- to create a 'directory' app that takes input of the first 3 chars of a persons last name
-- and lets the user choose which returned result to transfer to
-- Notes
-- inside the custom_sounds_path are 5 sound files taken from an * box and converted to wav
-- FS currently has no equivalent of these files
-- the files are: dir-instr.wav dir-intro-fn.wav dir-intro.wav dir-nomatch.wav dir-nomore.wav
-- the following must be set correctly
-- used for logging info
script_name = "directory.lua"
-- extension the script runs on... we call back to this if no matches
directory_extension = 799
custom_sounds_path = "/usr/local/freeswitch/sounds/custom/"
voicemail_path = "/usr/local/freeswitch/storage/voicemail/default/192.168.5.4/"
-- the 'names' table contains names in a CamelCase format with the first and last names separated
-- by an underscore
-- the table contains only dictionary entries
-- the key is the name and the value is the extension number
names = {
    Ash_George= 1001,
    Jones_Jerry= 1002,
    Smith_John= 1003
}
-- the following are used to generate digits sequences for the name/extension lookup
alphabet = "abcdefghijklmnopqrstuvwxy"
numbers = "22233344455566677778889999"
-- function used to check for the existence of a file
-- returns true if fname exists
function exists(fname)
    local f = io.open(fname, "r")
    if (f and f:read()) then return true end
end
-- function used to spell a name letter by letter
function sayname(full_name)
    -- find the center space then reverse the first and last names
    middle = string.find(full_name, ' ')
    first_name = string.sub(full_name, middle + 1, string.len(full_name))
    last_name = string.sub(full_name, 1, middle -1)
    freeswitch.consoleLog("info", script_name .. " : Now spelling out " .. first_name .. " " .. last_name ..
"\n")
    session:execute("phrase", "spell," .. first_name .. " " .. last_name);
end
-- function used to check if an extension already has a recorded name
function checkforname(extension)
    --check the actual voicemail directory for recorded_name.wav
    vmpath = voicemail_path .. extension .. "/recorded_name.wav"
    if exists(vmpath) then
        freeswitch.consoleLog("info", script_name .. " : recorded_name.wav for " .. extension .. " exists.\n")
        return vmpath
    else
        freeswitch.consoleLog("info", script_name .. " : recorded_name.wav for " .. extension .. " does not
exist.\n")
        return false
    end
end
end
-- create an all lowercase names table
```

```

lower_names = {}
for k,v in pairs(names)
do
    lower_names[string.lower(k)] = v
end
-- create lookup tables
code_to_names = {}
for k,v in pairs(lower_names)
do
    first_three_chars = string.sub(k, 1, 3)
    code = ""
    -- for each char in first_three_chars find the matching number
    for i=1,3
    do
        code = code .. string.sub(numbers, string.find(alphabet, string.sub(first_three_chars,i,i)))
    end
    -- at this point code will be 3 numbers
    -- see if the 3 digits are in our lookup table or not
    if code_to_names[code] == nil then
        code_to_names[code] = { k }
    else
        table.insert(code_to_names[code], k)
    end
end
-- code_to_names will look like the following at this point
-- code_to_names[566]={ "jones_jerry" }
-- next we need to start the session and play the greeting
session:answer()
session:execute( "sleep", "2000" )
freeswitch.consoleLog("info", script_name .. " : playing intro message and waiting for digits.\n")
digits = session:playAndGetDigits(3, 3, 10, 5000, "##", custom_sounds_path .. "dir-intro.wav", "", "");
-- log to console what was input
freeswitch.consoleLog("info", script_name .. " : captured the following digits - " .. digits .. "\n")
-- if it was 3 digits then process it
if (string.len(digits) == 3) then
    if (code_to_names[digits] ~= nil) then -- we have matches
        freeswitch.consoleLog("info", script_name .. " : found " .. digits .. " in the code_to_names table\n")
        -- sort the table so it comes out in good order
        table.sort(code_to_names[digits])
        -- get the # of items in code_to_names[digits]
        for count = 1,table.getn(code_to_names[digits])
        do
            freeswitch.consoleLog("info", script_name .. " : processing match found for " .. digits .. ". It is
            .. code_to_names[digits][count] .. "\n")
            full_name = string.gsub(code_to_names[digits][count], "_", " ")
            extension = lower_names[code_to_names[digits][count]]
            recorded_name = checkforname(extension)
            if (recorded_name == false) then
                sayname(full_name)
            else
                freeswitch.consoleLog("info", script_name .. " : playing the recorded_name.wav for extension "
                .. extension .. "\n")
                session:streamFile(recorded_name)
            end
            session:execute("phrase", "spell," .. extension)
            session:execute( "sleep", "1000" )
            -- give the option press 1 to accept this person or * to go to the next
            freeswitch.consoleLog("info", script_name .. " : playing 'press 1 or *' acceptance message and
            waiting for digit\n")
            accept_digit = session:playAndGetDigits(1, 1, 3, 2000, "##", custom_sounds_path .. "dir-instr.wav",
            "", "")
            freeswitch.consoleLog("info", script_name .. " : acceptance digit is " .. accept_digit .. "\n")
            if (accept_digit == "1") then
                accepted_extension = true
                session:transfer(extension, "XML", "default")
                break
            end
        end
        -- we've played them all and if nothing was accepted
        -- play the nomore matches wav and transfer back to the ivr
        if (accepted_extension ~= true) then

```

```
        freeswitch.consoleLog("info", script_name .. " : no more matches found. playing message and
transferring back to directory\n")
        session:streamFile( custom_sounds_path .. "dir-nomatch.wav" )
        session:transfer(directory_extension, "XML", "default")
    end
else
    -- no matches found at all
    -- transfer back when done
    freeswitch.consoleLog("info", script_name .. " : did not find " .. digits .. " in the code_to_names
table\n")
    freeswitch.consoleLog("info", script_name .. " : transferring back to directory\n")
    session:streamFile( custom_sounds_path .. "dir-nomatch.wav" )
    session:transfer(directory_extension, "XML", "default")
end
end
end
```