```
--------------------------------------------------------------------------------
-- Util.lua
-- by Stewart Bracken  http://stewart.bracken.bz  stew.bracken@gmail.com
-- Helper functions required for TwilioRestClient.
--------------------------------------------------------------------------------
local Util = setmetatable({}, nil)

--Set to false for less assert checks and better performance
local doTypeCheck = true

--DeepCopy is the only function in this file necessary for TwilioRestClient.
function Util.DeepCopy(object)
    if type(object) ~= "table" then
        return object
    end
    local new_table = {}
    for index, value in pairs(object) do
        new_table[Util.DeepCopy(index)] = Util.DeepCopy(value)
    end
    return setmetatable(new_table, getmetatable(object))
end

--Taken from http://lua-users.org/wiki/TableSerialization
function Util.table_print (tt, indent, done)
  done = done or {}
  indent = indent or 0
  if type(tt) == "table" then
    local sb = {}
    for key, value in pairs (tt) do
      table.insert(sb, string.rep (" ", indent)) -- indent it
      if type (value) == "table" and not done [value] then
        done [value] = true
        table.insert(sb, "{\n");
        table.insert(sb, Util.table_print (value, indent + 2, done))
        table.insert(sb, string.rep (" ", indent)) -- indent it
        table.insert(sb, "}\n");
      elseif "number" == type(key) then
        table.insert(sb, string.format("\"%s\"\n", tostring(value)))
      else
        table.insert(sb, string.format(
            "%s = \"%s\"\n", tostring (key), tostring(value)))
       end
    end
    return table.concat(sb)
  else
```

```
    return tt .. "\n"
  end
end

--Also taken from the above URL
function Util.to_string( tbl )
    if  "nil"       == type( tbl ) then
        return tostring(nil)
    elseif  "table" == type( tbl ) then
        return Util.table_print(tbl)
    elseif  "string" == type( tbl ) then
        return tbl
    else
        return tostring(tbl)
    end
end
--Returns true if checkMe is one of the types passed in to arg
-- Ex multiTypeCheck({1,2,3}, "string", "number") == false
local function multiTypeCheck(checkMe, ...)
    for i, v in ipairs(arg) do
        if type(checkMe) == v then
            return true
        end
    end
    return false
end

--For performance, turn off typechecking at top of Util
Util.typechk = doTypeCheck and multiTypeCheck or function(...) return true end


--Unused because Corona network function encodes URLs as far as I know
function Util.url_encode(str)
  if (str) then
    str = string.gsub (str, "\n", "\r\n")
    str = string.gsub (str, "([^%w %-%_%.%~])",
        function (c) return string.format ("%%%02X", string.byte(c)) end)
    str = string.gsub (str, " ", "+")
  end
  return str
end
return Util
```