

RESTful Verto

mod_verto has a builtin HTTP server so it is possible to build restful APIs.

To use the HTTP interface, put the following config in your verto profile

```
<vhosts>
  <vhost domain="localhost">
    <param name="alias" value="seven.local freeswitch.org"/>
    <!-- <param name="root" value="/usr/local/freeswitch/htdocs"/> -->
    <!-- <param name="script_root" value="/usr/local/freeswitch/scripts"/> -->
    <param name="index" value="index.html"/>
  <!--
    <param name="auth-realm" value="FreeSWITCH"/>
    <param name="auth-user" value="freeswitch"/>
    <param name="auth-pass" value="rocks"/>
  -->

  <rewrites>
    <rule expression="^/api" value="/my_custom_api.lua"/>
    <rule expression="^/channels" value="/rest.lua"/>
  </rewrites>
</vhost>
```

so it looks like

```
<profiles>
  <profile name="mine">
    <vhosts>
      <vhost>
      </vhost>
    </vhosts>
  </profile>
</profiles>
```

Note: Only one vhost is currently supported.

Put rest.lua to your scripts dir: (demo only, edit for your needs)

rest.lua

```
--[[
Restful by Seven Du.

GET    /channels
GET    /channels/uuid
POST   /channels
PUT    /channels/uuid
DELETE /channels/uuid
DELETE /channels
]]

function headers()
  stream:write("HTTP/1.0 200 OK\r\n")
  if (accept == "application/json") then
    stream:write("Content-Type: application/json\r\n")
  else
    stream:write("Content-Type: text/plain\r\n")
  end
  stream:write("\r\n")
end

-- print(env:serialize())
```

```

api = freeswitch.API()
method = env:getHeader("Request-Method")
http_uri = env:getHeader("HTTP-Request-URI")
http_query = env:getHeader("HTTP-QUERY")
accept = env:getHeader("Accept")
uuid = string.sub(http_uri, "11") -- remove /channels/ from uri
freeswitch.consoleLog("ERR", "[" .. method .. "]\n")
-- freeswitch.consoleLog("ERR", http_uri .. "\n")
-- freeswitch.consoleLog("ERR", http_query .. "\n")

if (method == "GET") then
    if not (uuid == "") then
        format = ""
        if accept == "application/json" then
            format = " json"
        end
        ret = api:execute("uuid_dump", uuid .. format)
    else
        format = ""
        if (accept == "application/json") then
            format = " as json"
        end
        ret = api:execute("show", "channels" .. format)
    end
elseif (method == "POST") then
    dest = env:getHeader("destNumber")
    app = "echo"
    dialstr = "user/" .. dest .. " &" .. app
    print(dialstr)
    ret = api:execute("originate", dialstr)
elseif method == "PUT" then
    if action == "nomedia" then
        cmd = "uuid_media"
        arg = uuid .. " off"
    elseif action == "media" then
        cmd = "uuid_media"
        arg = uuid .. " on"
    elseif action == "hold" then
        cmd = "uuid_hold"
        arg = "on " .. uuid
    elseif action == "media" then
        cmd = "uuid_media"
        arg = "off" .. uuid
    end
    ret = api:execute(cmd, arg)
elseif method == "DELETE" then
    if not (uuid == "") then
        cmd = "uuid_kill"
        arg = uuid
    else
        cmd = "hupall"
        arg = ""
    end
    ret = api:execute(cmd, arg)
end
headers()
stream:write(ret .. "\n")

```

reload mod_verto and try:

```

curl -0 localhost:8081/channels
curl -0 -XPOST -d "destNumber=1000" localhost:8081/channels
curl -0 localhost:8081/channels/1f0802b7-3568-4eb6-b372-182861b56d9b
curl -0 -H "Accept: application/json" localhost:8081/channels/1f0802b7-3568-4eb6-b372-182861b56d9b
curl -0 -XDELETE localhost:8081/channels

```

