This repository Search **Pull requests** **Issues** **Gist**



openresty / **stream-lua-nginx-module**

Watch 60 Star 230 Fork 69

Code Issues 15 Pull requests 10 Projects 0 Wiki Pulse Graphs

Embed the power of Lua into NGINX TCP servers

193 commits **2** branches **1** release **6** contributors

Branch: **master** **New pull request** **Create new file** **Upload files** **Find file** **Clone or download**

agentzh committed on **GitHub** Merge pull request #59 from rainingmaster/119_to_stream ... Latest commit `ed7c0ea` on Dec 19, 2016

src	fixed coding style issues found by ngx-releng. alas.	5 months ago
t	transform test case 119-config-prefix.t from ngx_http_lua to ngx_stre...	2 months ago
util	tests: added 128-duplex-tcp-socket.t.	a year ago
.gitattributes	added .gitattribute.	a year ago
.gitignore	feature: implemented ngx.encode_args() and ngx.decode_args(). thanks ...	10 months ago
.travis.yml	travis-ci: added nginx-1.11.2 to build matrix, moved package manageme...	3 months ago
README.md	doc: updated copyright notice.	2 months ago
config	feature: implemented ngx.encode_args() and ngx.decode_args(). thanks ...	10 months ago
valgrind.suppress	updated valgrind.suppress for nginx 1.11.2+ on linux i386.	6 months ago

README.md

Name

ngx_stream_lua_module - Embed the power of Lua into Nginx stream/TCP Servers.

This module is not distributed with the Nginx source. See the installation instructions.

Table of Contents

Status

Experimental.

Synopsis

```
events {
    worker_connections 1024;
}

stream {
    # define a TCP server listening on the port 1234:
    server {
        listen 1234;

        content_by_lua_block {
            ngx.say("Hello, Lua!")
        }
    }
}
```

Set up as an SSL TCP server:

```
stream {
    server {
```

```
        listen 4343 ssl;

        ssl_protocols       TLSv1 TLSv1.1 TLSv1.2;
        ssl_ciphers         AES128-SHA:AES256-SHA:RC4-SHA:DES-CBC3-SHA:RC4-MD5;
        ssl_certificate     /path/to/cert.pem;
        ssl_certificate_key /path/to/cert.key;
        ssl_session_cache   shared:SSL:10m;
        ssl_session_timeout 10m;

        content_by_lua_block {
            local sock = assert(ngx.req.socket(true))
            local data = sock:receive()  -- read a line from downstream
            if data == "thunder!" then
                ngx.say("flash!")  -- output data
            else
                ngx.say("boom!")
            end
            ngx.say("the end...")
        }
    }
}
```

Listening on a UNIX domain socket is also supported:

```
stream {
    server {
        listen unix:/tmp/nginx.sock;

        content_by_lua_block {
            ngx.say("What's up?")
            ngx.flush(true)  -- flush any pending output and wait
            ngx.sleep(3)  -- sleeping for 3 sec
            ngx.say("Bye bye...")
```

```
        }
    }
}
```

Description

This is a port of the ngx_http_lua_module to the NGINX "stream" subsystem so as to support generic stream/TCP clients in the downstream.

Lua APIs and directive names rename the same as the `ngx_http_lua_module`.

Back to TOC

Directives

The following directives are ported directly from `ngx_http_lua_module`. Please check the documentation of `ngx_http_lua_module` for more details about their usage and behavior.

- lua_code_cache
- lua_regex_cache_max_entries
- lua_package_path
- lua_package_cpath
- init_by_lua_block
- init_by_lua_file
- init_worker_by_lua_block
- init_worker_by_lua_file
- content_by_lua_block

- content_by_lua_file
- lua_shared_dict
- lua_socket_connect_timeout
- lua_socket_buffer_size
- lua_socket_pool_size
- lua_socket_keepalive_timeout
- lua_socket_log_errors
- lua_ssl_ciphers
- lua_ssl_crl
- lua_ssl_protocols
- lua_ssl_trusted_certificate
- lua_ssl_verify_depth
- lua_check_client_abort
- lua_max_pending_timers
- lua_max_running_timers

In addition, `ngx_stream_lua_module` provides the following directives:

- `lua_resolver`

  Just an equivalent to the resolver directive in the NGINX "http" subsystem.

- `lua_resolver_timeout`

  Just an equivalent to the resolver_timeout directive in the NGINX "http" subsystem.

- `lua_lingering_close`

Just an equivalent to the lingering_close directive in the NGINX "http" subsystem.

- `lua_lingering_time`

  Just an equivalent to the lingering_time directive in the NGINX "http" subsystem.

- `lua_lingering_timeout`

  Just an equivalent to the lingering_timeout directive in the NGINX "http" subsystem.

The send_timeout directive in the NGINX "http" subsystem is missing in the "stream" subsystem. So `ngx_stream_lua_module` uses the `lua_socket_send_timeout` for this purpose.

Back to TOC

Nginx API for Lua

Many Lua API functions are ported from the `ngx_http_lua_module`. Check out the official manual of `ngx_http_lua_module` for more details on these Lua API functions.

- ngx.var.VARIABLE

  Unlike the "http" subsystem, the "stream" subsystem of the NGINX core does not support NGINX variables at all. But still we emulate some of the NGINX builtin variables in this `ngx.var` API for compatibility with `ngx_http_lua_module` and ease of extracting certain session-wise information. Currently the following "variables" are supported:

  - `ngx.var.pid` for $pid
  - `ngx.var.connection` for $connection
  - `ngx.var.remote_addr` for $remote_addr
  - `ngx.var.binary_remote_addr` for $binary_remote_addr

    - `ngx.var.remote_port` for $remote_port
    - `ngx.var.nginx_version` for $nginx_version
    - `ngx.var.server_addr` for $server_addr
    - `ngx.var.server_port` for $server_port

- Core constants

  `ngx.OK`, `ngx.ERROR`, and etc.

- Nginx log level constants

  `ngx.ERR`, `ngx.WARN`, and etc.

- print
- ngx.ctx
- ngx.req.socket

  Only raw request sockets are supported, for obvious reasons. The `raw` argument value is ignored and the raw request socket is always returned. Unlike `ngx_http_lua_module`, you can still call output API functions like `ngx.say`, `ngx.print`, and `ngx.flush` after acquiring the raw request socket via this function.

- ngx.print
- ngx.say
- ngx.log
- ngx.flush

  This call currently ignores the `wait` argument and always wait for all the pending output to be completely flushed out (to the system socket send buffers).

- ngx.exit
- ngx.eof
- ngx.sleep
- ngx.escape_uri
- ngx.unescape_uri
- ngx.encode_args
- ngx.decode_args
- ngx.encode_base64
- ngx.decode_base64
- ngx.crc32_short
- ngx.crc32_long
- ngx.hmac_sha1
- ngx.md5
- ngx.md5_bin
- ngx.sha1_bin
- ngx.quote_sql_str
- ngx.today
- ngx.time
- ngx.now
- ngx.update_time
- ngx.localtime
- ngx.utctime
- ngx.re.match
- ngx.re.find
- ngx.re.gmatch

- ngx.re.sub
- ngx.re.gsub
- ngx.shared.DICT
- ngx.socket.tcp
- ngx.socket.udp
- ngx.socket.connect
- ngx.get_phase
- ngx.thread.spawn
- ngx.thread.wait
- ngx.thread.kill
- ngx.on_abort
- ngx.timer.at
- ngx.timer.running_count
- ngx.timer.pending_count
- ngx.config.debug
- ngx.config.subsystem

  Always takes the Lua string value `"stream"` in this module.

- ngx.config.prefix
- ngx.config.nginx_version
- ngx.config.nginx_configure
- ngx.config.ngx_lua_version
- ngx.worker.exiting
- ngx.worker.pid

- ngx.worker.count
- ngx.worker.id
- coroutine.create
- coroutine.resume
- coroutine.yield
- coroutine.wrap
- coroutine.running
- coroutine.status

Back to TOC

TODO

- Add new directives `access_by_lua_block` and `access_by_lua_file`.
- Add new directives `log_by_lua_block` and `log_by_lua_file`.
- Add new directives `balancer_by_lua_block` and `balancer_by_lua_file`.
- Add new directives `ssl_certificate_by_lua_block` and `ssl_certificate_by_lua_file`.
- Add `ngx.semaphore` API.
- Add `ngx_meta_lua_module` to share as much code as possible between this module and `ngx_http_lua_module` and allow sharing of `lua_shared_dict`.
- Add support for lua-resty-core.
- Add `lua_postpone_output` to emulate the postpone_output directive.

Back to TOC

Nginx Compatibility

The latest version of this module is compatible with the following versions of Nginx:

- 1.9.x >= 1.9.7 (last tested: 1.9.7)

Nginx cores older than 1.9.7 (exclusive) are *not* supported.

Back to TOC

Installation

This module can be manually compiled into Nginx or OpenResty:

1. Install LuaJIT 2.0 or 2.1 (recommended) or Lua 5.1 (Lua 5.2+ are *not* supported yet). LuaJIT can be downloaded from the the LuaJIT project website and Lua 5.1, from the Lua project website. Some distribution package managers also distribute LuaJIT and/or Lua.
2. Download the latest version of ngx_stream_lua HERE.
3. Download the latest supported version of NGINX HERE (See Nginx Compatibility) or the OpenResty bundle from HERE.

Build the source of NGINX or OpenResty with this module, like below:

```
wget 'http://nginx.org/download/nginx-1.9.7.tar.gz'
tar -xzvf nginx-1.9.7.tar.gz
cd nginx-1.9.7/

# tell nginx's build system where to find LuaJIT 2.0:
export LUAJIT_LIB=/path/to/luajit/lib
export LUAJIT_INC=/path/to/luajit/include/luajit-2.0
```

```
# tell nginx's build system where to find LuaJIT 2.1:
export LUAJIT_LIB=/path/to/luajit/lib
export LUAJIT_INC=/path/to/luajit/include/luajit-2.1

# or tell where to find Lua if using Lua instead:
#export LUA_LIB=/path/to/lua/lib
#export LUA_INC=/path/to/lua/include

# Here we assume Nginx is to be installed under /opt/nginx/.
./configure --prefix=/opt/nginx \
        --with-ld-opt="-Wl,-rpath,/path/to/luajit-or-lua/lib" \
        --with-stream \
        --with-stream_ssl_module \
        --add-module=/path/to/stream-lua-nginx-module
```

Back to TOC

Community

Back to TOC

English Mailing List

The openresty-en mailing list is for English speakers.

Back to TOC

Chinese Mailing List

The openresty mailing list is for Chinese speakers.

Back to TOC

Code Repository

The code repository of this project is hosted on github at openresty/stream-lua-nginx-module.

Back to TOC

Bugs and Patches

Please submit bug reports, wishlists, or patches by

1. creating a ticket on the GitHub Issue Tracker,
2. or posting to the OpenResty community.

Back to TOC

Copyright and License

Back to TOC

See Also

- ngx_http_lua_module
- ngx_stream_echo_module
- OpenResty

Back to TOC