



Модули Си  
Скрипты Lua  
Скрипты JS

# О модулях

- API gateway (логи, статистика, кеширование)
  - Авторизация через IdP. Чтобы не авторизовываться на каждом бэке.
- Врезка контента, реклама, статистика, логи, фильтрация
- Не простой язык

# O Lua

- Openresty, luaJIT
- Одна виртуальная машина (3 Gb) на worker. Нехватка ОЗУ свалит worker
- Есть мусорщик
- Язык прост, но не популярен

# Работа с данными

```
apt-get install lua-nginx-redis
```

```
server {  
    location / {  
        content_by_lua '  
            local redis = require "nginx.redis"  
            local red = redis:new()  
            local ok, err = red:connect("127.0.0.1", 6379)  
            ok, err = red:incr("test")  
            local res, err = red:get("test")  
            ngx.say("hits: ", res)  
        ',  
    }  
}
```

<https://ruhighload.com/%D0%98%D1%81%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5+lua+%D0%B2+nginx>

# Получение тела запроса

```
local data = ngx.req.get_body_data()
if ( data == nil ) then
    local handle = io.open(ngx.req.get_body_file(), "rb")
    data = handle:read("*all")
    handle:close()
end
```

<https://ruhighload.com/%D0%98%D1%81%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5+lua+%D0%B2+nginx>

# О JS

- Модуль `njs` - позволяет писать обработчики на JS
- `Njs` - родной скриптинг, от разработчиков `nginx`
- Почему JS: Синтаксис хорошо смотрится в конфиге (фигурные скобки). Быстро работает в один поток, без блокировок, как `worker` в `nginx`. Очередь событий JS = очередь соединений `nginx`. Популярен. Развивается.

- Зачем им собственный интерпретатор?  
V8/SpiderMonkey медленно стартуют.
- Njs - не application сервер. Не замена Node.js
- Почему быстрый
  - Компиляция в байт код при старте nginx
  - Для каждого запроса создается клон VM
  - Нет сборщика мусора

- При работе через WebSocket может скопиться мусор
- Примеры <https://github.com/xeioex/njs-examples>



## Результаты двух асинхронных подзапросов объединяются в один ответ

```
http {  
    js_include example.js;  
    server {  
        listen 80;  
        location /join {  
            js_content join;  
        }  
        location /foo {  
            proxy_pass http://localhost:8080;  
        }  
        location /bar {  
            proxy_pass http://localhost:8090;  
        }  
    }  
}
```

<https://github.com/xeioex/njs-examples>

## example.js

```
function join(r) {
  join_subrequests(r, ['/foo', '/bar']);
}

function join_subrequests(r, subs) {
  var parts = [];

  function done(reply) {
    parts.push({ uri: reply.uri,
                  code: reply.status,
                  body: reply.responseBody });
  }

  if (parts.length == subs.length) {
    r.return(200, JSON.stringify(parts));
  }
}

for (var i in subs) {
  r.subrequest(subs[i], done);
}
}
```

Спасибо за внимание!

Каримов Азат  
[a.karimov@modulbank.ru](mailto:a.karimov@modulbank.ru)