

ghttp

A tiny, go based backend framework

Why another web server?

- Because it is RESTful?
 - Because it gets going in less than 5 minutes.
 - Fast? Build on top of the Go HTTP server.
 - gottp has been designed keeping APIs & backend servers in mind.
 - Does all the fancy tricks that your backend server (not you) should be doing.
 - Supports Unix Domain Sockets and CORS out-of-the-box.
 - Concurrent? The Go HTTP server runs each request in its own goroutine.
 - Supports constant running and graceful restarts of background workers.
 - Support for Batch requests like Facebook's Graph API.
-

Sample Implementation

```
package main

import "gopkg.in/simversity/gottp.v2"

type hello struct {
    gottp.BaseHandler
}

func (self *hello) Get(req *gottp.Request) {
    req.Write("hello world")
}

func main() {
    gottp.NewUrl("hello", "/hello/\\w{3,5}/?$",
new(hello))
    gottp.DefaultServer()
}
```

```
piyush:shortener [master] $ curl -I http://localhost:9010/hello/world
```

```
HTTP/1.1 200 OK
Content-Type: application/json
Server: Gottp Server
Date: Sun, 01 Feb 2015 12:21:24 GMT
Content-Length: 61
```

RESTful by design

```
package main

import (
    "gopkg.in/simversity/gottp.v1"
)

type helloHandler struct {
    gottp.BaseHandler
}

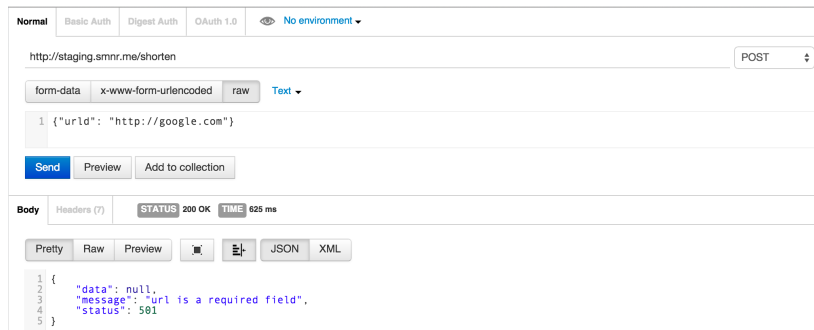
func (self *helloHandler) Post(req *gottp.
Request) {
    req.Write("hello POST")
}

func (self *helloHandler) Get(req *gottp.Request)
{
    req.Write("hello Get")
}
```

Inbuilt request Argument validation

```
type UrlModel struct {
    Url      string `json:"url" required:"true"`
}

model := new(UrlModel)
req.ConvertArguments(&model)
errors := utils.Validate(&model)
```



The screenshot shows a REST client interface with the following details:

- URL:** `http://staging.smnr.me/shorten`
- Method:** `POST`
- Form:** `form-data` (selected), `x-www-form-urlencoded`, `raw`
- Body:** `1 { "url": "http://google.com" }`
- Buttons:** `Send`, `Preview`, `Add to collection`
- Response:** `STATUS 200 OK`, `TIME 625 ms`
- Body Tab:** `Pretty` (selected), `Raw`, `Preview`
- Response Body:**

```
1 {
2   "data": null,
3   "message": "url is a required field",
4   "status": 501
5 }
```

Supports Domain Sockets

```
piyush:GOPATH [] $ ./bin/shortener.v2
2015/02/01 16:11:32 No config file supplied. Using defaults.
2015/02/01 16:11:32 local_server.go:85: Listening on
/tmp/shortener.sock
```

...and gracefully cleans them on exit

```
sigchan := make(chan os.Signal, 10)
signal.Notify(sigchan, os.Interrupt, syscall.
SIGTERM)

//NOTE: Capture every Signal right now.
//signal.Notify(sigchan)

s := <-sigchan
log.Println("Exiting Program. Got Signal: ", s)

//perform cleanups & wait for all operations to end

cleanAddr(addr)
os.Exit(0)
```

Response data compression

```
piyush:shortener [master] $ curl -I http://staging.smnr.me/urls
```

```
HTTP/1.1 200 OK
Date: Wed, 18 Feb 2015 13:24:32 GMT
Content-Type: application/json
Content-Length: 61
Connection: keep-alive
Access-Control-Allow-Origin: *
Server: Gottp Server
```

...based on Content-Type request headers

```
piyush:shortener [master] $ curl -H Accept-Encoding:
gzip -I http://staging.smnr.me/urls
```

```
HTTP/1.1 200 OK
Date: Wed, 18 Feb 2015 13:25:04 GMT
Content-Type: application/json
Content-Length: 85
Connection: keep-alive
Access-Control-Allow-Origin: *
Content-Encoding: gzip
Server: Gottp Server
```

```
piyush:shortener [master] $ curl -H Accept-Encoding:
deflate -I http://staging.smnr.me/urls
```

```
HTTP/1.1 200 OK
Date: Wed, 18 Feb 2015 13:25:30 GMT
Content-Type: application/json
Content-Length: 73
Connection: keep-alive
Access-Control-Allow-Origin: *
Content-Encoding: deflate
Server: Gottp Server
```

Email and extendible tracebacks

<http://gopkg.in/simversity/gotracer.v1> + Extra Information

Jan 1, 2015 at 11:12pm (UTC)

Host: [127.0.0.1:9011](#)
Method: POST
Protocol: HTTP/1.1
RemoteIP: @
URI: /publish
Referer:
Arguments: {"content_id":"142015381728364370447976","user_id":"141989498101811860663702"}

Traceback

```
goroutine 50811 [running]:
github.com/Simversity/gottp.Exception(0xc210101780)
    /var/lib/jenkins/jobs/siminars-usher-prod/workspace/
gopath/src/github.com/Simversity/gottp/httpserver.go:74 +0xa5
runtime.panic(0x6cd740, 0xc21012d710)
    /usr/local/go/src/pkg/runtime/panic.c:248 +0x106
gorm.(*MConn).Insert(0xc2100eafb0, 0x74fe80, 0x4, 0x7f4311bd2968,
0x1, ...)
    /var/lib/jenkins/jobs/siminars-usher-prod/workspace/
src/gorm/con.go:317 +0x425
subpub/feed.makeAndSaveFeed(0xc2100c62d0, 0xc2100c62d0, 0x764b30)
    /var/lib/jenkins/jobs/siminars-usher-prod/workspace/
src/subpub/feed/feed.go:41 +0xf3
subpub/feed.responseFeeder.addFeed(0xc210109280, 0xc2100c74c0,
0x18, 0xc21011fdd0, 0xc210109280, ...)
    /var/lib/jenkins/jobs/siminars-usher-prod/workspace/
src/subpub/feed/responses.go:55 +0x796
subpub/feed.PushFeed(0xc2100c74c0, 0x18, 0xc210106c20, 0x18,
0xc21011fdd0, ...)
    /var/lib/jenkins/jobs/siminars-usher-prod/workspace/
```

Cross Origin & Named URLs

```
$.ajax({  
  url: "http://staging.smnr.me/urls/",  
  type: "GET"  
}).done(function(ret) {  
  console.log(ret.data)  
})
```

```
third_party.js?server=siminars.com:13 XHR finished  
loading: GET "http://staging.smnr.me/urls/".
```

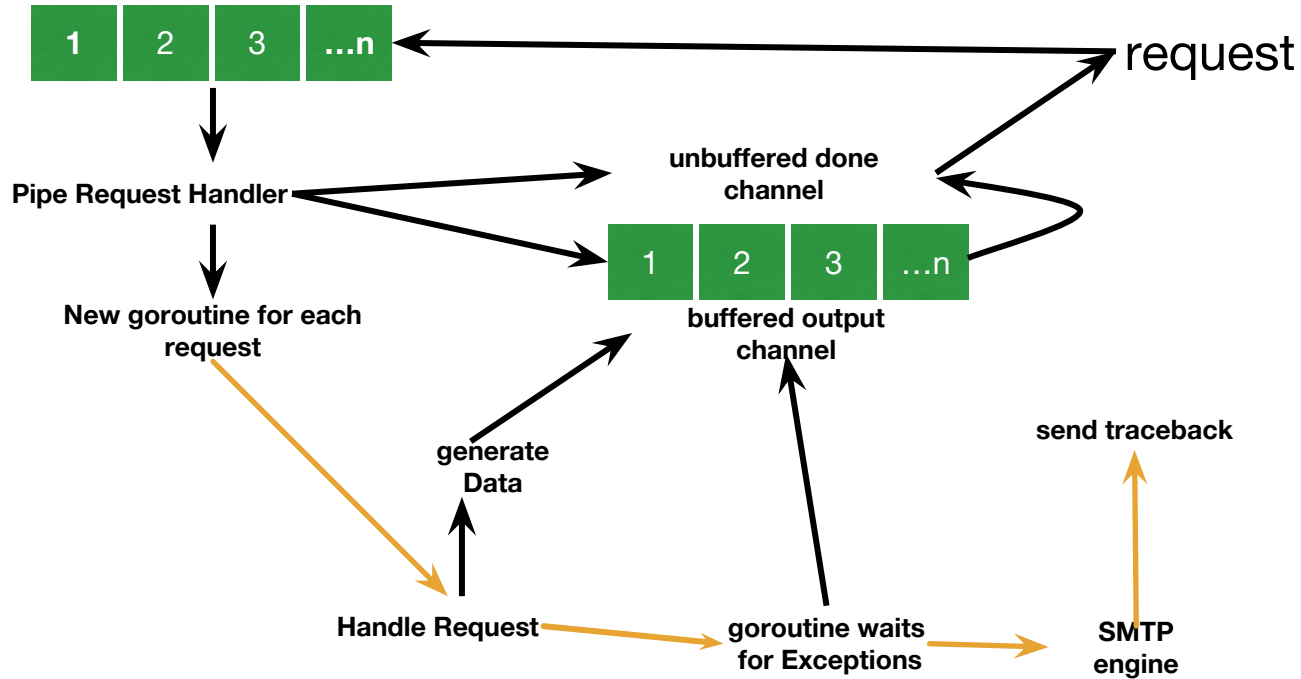
```
Object {  
  redirect: "/\w{6,10}/?$",  
  shorten: "/shorten/?$"  
}
```

```
$.ajax({  
  url: "http://staging.smnr.me/shorten/",  
  type: "POST",  
  data: JSON.stringify({  
    "url": "http://google.com"  
  }),  
}).done(function(ret) {  
  console.log(ret.data)  
})
```

```
third_party.js?server=siminars.com:13 XHR finished  
loading: POST "http://staging.smnr.me/shorten/".
```

```
Object {  
  url: "http://google.com",  
  short_url: "http://staging.smnr.me/njyy5o"  
}
```


PIPE(s) & Architecture



Sequential Batch

```
piyush:shortener [master] $ curl -X POST -H
"Content-Type: application/json" -d '{"stack": [
{"url": "/shorten", "data": {"url": "http://google.
com"}, "method": "POST"},
{"url": "/shorten", "data": {"url": "http://duckduckgo.
com"}, "method": "POST"},
{"url": "/shorten", "data": {"url": "http://microsoft.
com"}, "method": "POST"},
{"url": "/shorten", "data": {"url": "http://yahoo.
com"}, "method": "POST"}
]}' http://staging.smnr.me/pipe
```

```
piyush:shortener [master] $
{"data": [
  {"data": {
    {"url": "http://google.com",
      "short_url": "http://staging.smnr.me/nj3b6f"},
    "index": 0,
    "message": "",
    "status": 200},
  {"data": {
    {"url": "http://duckduckgo.com",
      "short_url": "http://staging.smnr.me/nj3bde"},
    "index": 1,
    "message": "",
    "status": 200},
  {"data": {
    {"url": "http://microsoft.com",
      "short_url": "http://staging.smnr.me/nj3bdf"},
    "index": 2,
    "message": "",
    "status": 200},
  {"data": {
    {"url": "http://yahoo.com",
      "short_url": "http://staging.smnr.me/nj3bdg"},
    "index": 3,
    "message": "",
    "status": 200}
  ],
  "message": "",
  "status": 200}
```

Async-Batch

```
piyush:shortener [master] $ curl -X POST -H
"Content-Type: application/json" -d '
{"stack": [
{"url": "/shorten", "data": {"url": "http://google.com"},
"method": "POST"},
{"url": "/shorten", "data": {"url": "http://duckduckgo.com"},
"method": "GET"},
{"url": "/shorten", "data": {"url": "http://microsoft.com"},
"method": "POST"},
{"url": "/shorten", "data": {"url": "http://yahoo.com"},
"method": "POST"}
]}
' http://staging.smnr.me/async-pipe
```

```
piyush:shortener [master] $
{"data": [
  {"data":
    {"url": "http://google.com",
      "short_url": "http://staging.smnr.me/nj3b6f"},
    "index": 0,
    "message": "",
    "status": 200},
  {"data": null,
    "index": 1,
    "message": "Method not Implemented",
    "status": 501},
  {"data":
    {"url": "http://microsoft.com",
      "short_url": "http://staging.smnr.me/nj3bdf"},
    "index": 2,
    "message": "",
    "status": 200},
  {"data":
    {"url": "http://yahoo.com",
      "short_url": "http://staging.smnr.me/nj3bdg"},
    "index": 3,
    "message": "",
    "status": 200}],
  "message": "",
  "status": 200}
```

Upcoming features

- Protocol Buffer/Avro support POST/PUT requests.
 - Implement XML as new response Content-Type.
 - Dependencies between operations in the PIPE request.
 - Server controlled background Workers. [Done]
-

Thank you.

<https://github.com/Simiversity/gottp>
piyush@piyushverma.net
