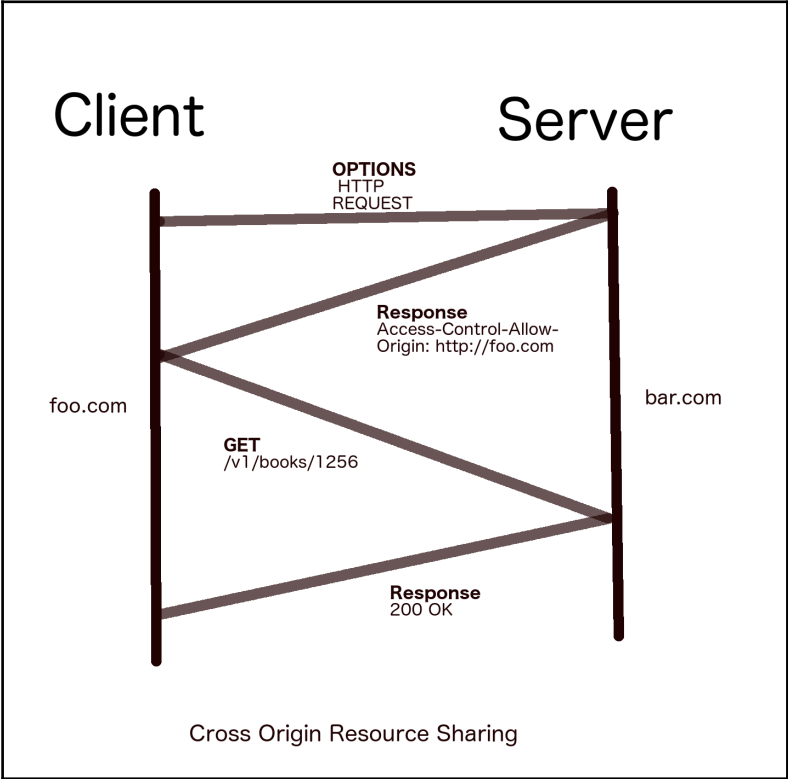
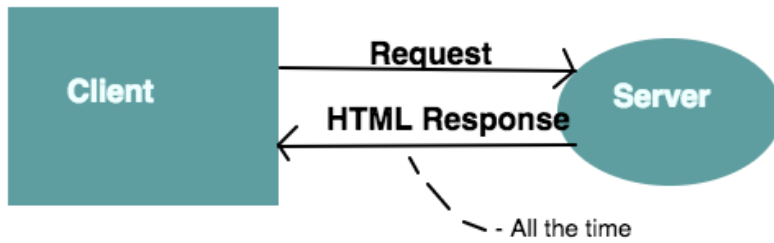


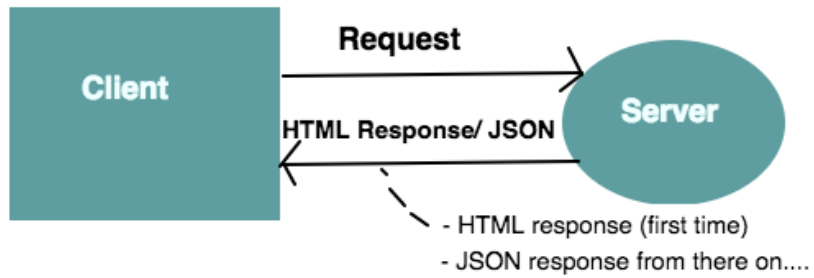
Chapter 1: Getting Started with REST API Development



Old Way



New Way



```
bin/
  hello          # command executable
pkg/
  linux_amd64/   # this will reflect your OS and architecture
    github.com/user/
      stringutil.a # package object
src/
  github.com/user/
    hello/
      hello.go    # command source
    stringutil/
      reverse.go  # package source
```

```
└─ github.com
   └─ narenaryan
      └─ romanNumerals
         └─ data.go
            └─ romanserver
               └─ main.go
```

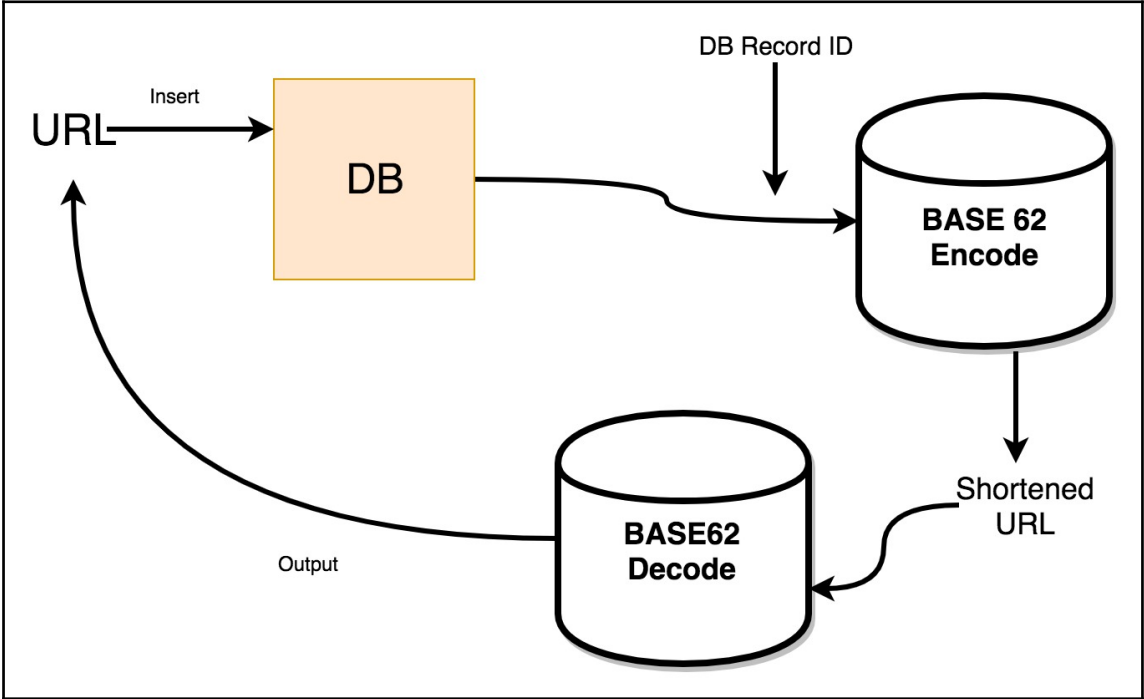
```
root@SiteB-Host1:~# supervisorctl
myserver                                RUNNING    pid 11405, uptime 0:00:01
supervisor> help

default commands (type help <topic>):
=====
add      exit      open  reload  restart  start  tail
avail   fg        pid   remove  shutdown status  update
clear   maintail  quit  reread  signal   stop   version

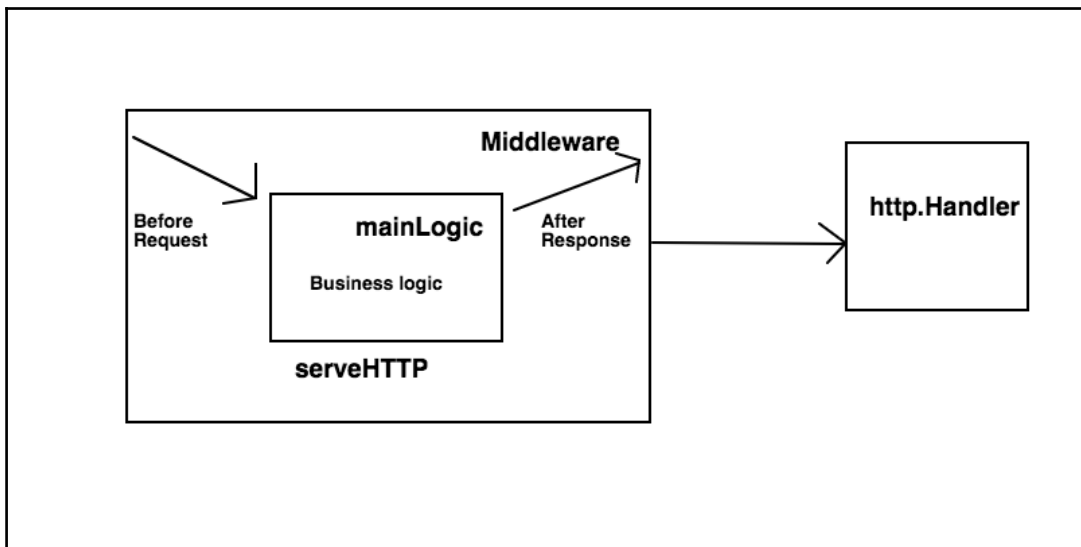
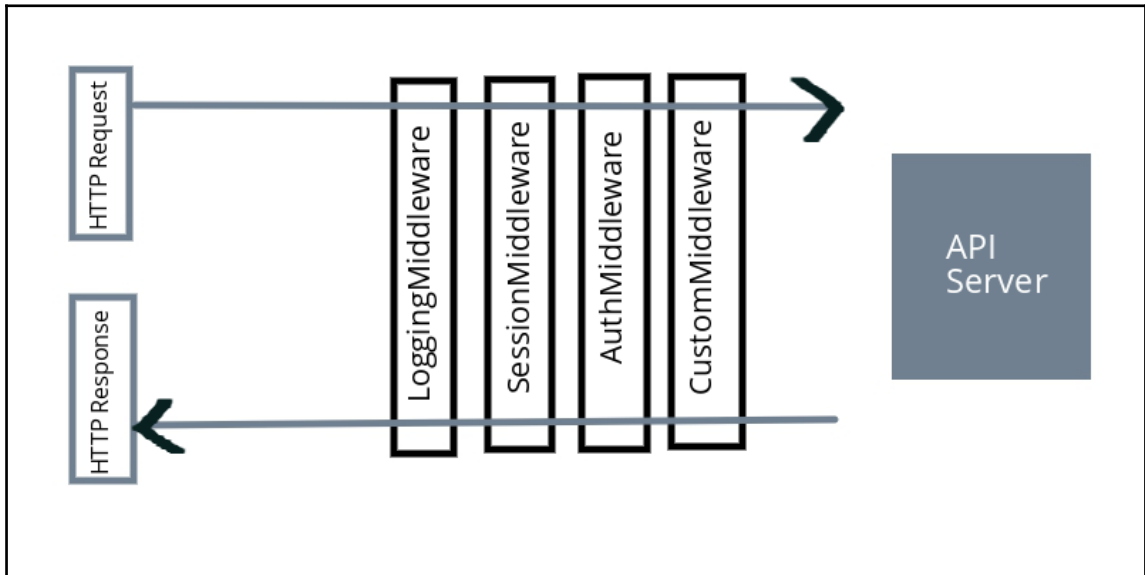
supervisor> █
```

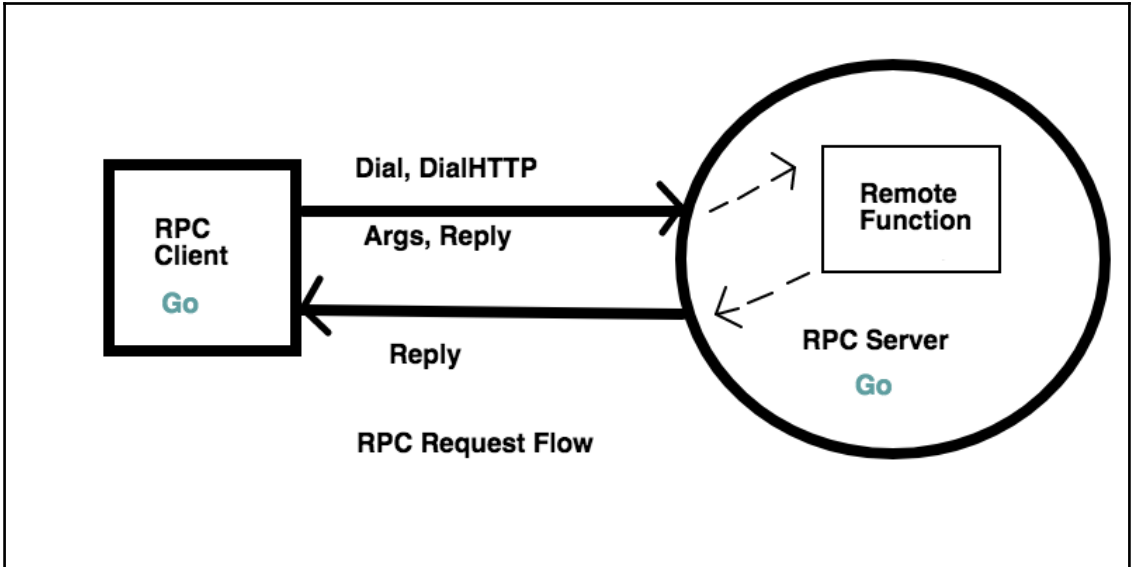
```
[20:50:38] Starting 'watch'...  
[20:50:38] Finished 'watch' after 45 ms  
[20:50:38] Starting 'default'...  
[20:50:38] Finished 'default' after 11 µs  
[20:50:46] Starting 'install-binary'...  
[20:50:47] Finished 'install-binary' after 1.07 s  
[20:50:47] Starting 'restart-supervisor'...  
myservice: stopped  
myservice: started  
[20:50:48] Finished 'restart-supervisor' after 1.14 s
```

Chapter 2: Handling Routing for Our REST Services



Chapter 3: Working with Middleware and RPC





Chapter 4: Simplifying RESTful Services with Popular Go Frameworks

```
/home/naren/workspace/src/github.com/narenaryan
.
├─ dbutils
│  └─ init-tables.go
│  └─ models.go
├─ railAPI
│  └─ main.go
└─ railapi.db

2 directories, 4 files
```

```
[9:29:34] naren:ginExamples git:(master*) $ go run ginBasic.go
[GIN-debug] [WARNING] Running in "debug" mode. Switch to "release" mode in production.
- using env:   export GIN_MODE=release
- using code: gin.SetMode(gin.ReleaseMode)

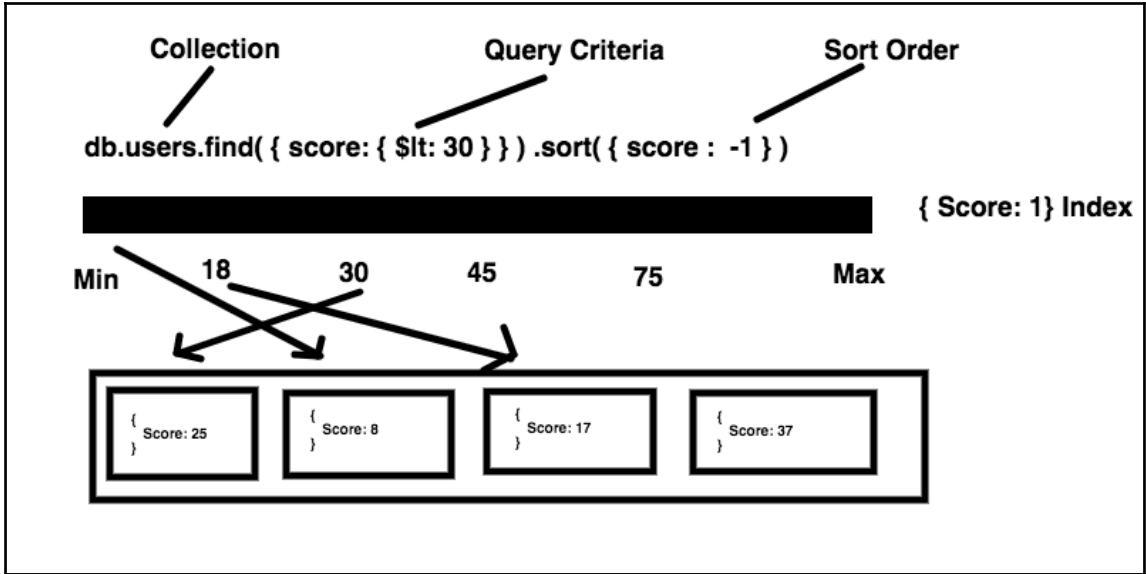
[GIN-debug] GET    /pingTime          --> main.main.func1 (3 handlers)
[GIN-debug] Listening and serving HTTP on :8000
[GIN] 2017/06/11 - 09:29:44 | 200 |      184.984µs | ::1 | GET    /pingTime
```

```
[GIN-debug] GET    /v1/stations/:station_id --> main.GetStation (3 handlers)
[GIN-debug] POST   /v1/stations       --> main.CreateStation (3 handlers)
[GIN-debug] DELETE /v1/stations/:station_id --> main.RemoveStation (3 handlers)
[GIN-debug] Listening and serving HTTP on :8000
[GIN] 2017/06/11 - 16:07:35 | 200 |    384.756µs | 10.101.254.223 | GET    /v1/stations/1
[GIN] 2017/06/11 - 16:09:11 | 200 |    222.511µs | 10.101.254.223 | GET    /v1/stations/2
[GIN] 2017/06/11 - 16:09:59 | 404 |     1.162µs | 10.101.254.223 | GET    /v1/station/1
[GIN] 2017/06/11 - 16:10:03 | 200 |    232.36µs | 10.101.254.223 | GET    /v1/stations/1
[GIN] 2017/06/11 - 16:55:15 | 200 |    290.281µs | 10.101.254.223 | GET    /v1/stations/1
[GIN] 2017/06/11 - 16:56:10 | 200 |    282.856µs | 10.101.254.223 | DELETE /v1/stations/1
```

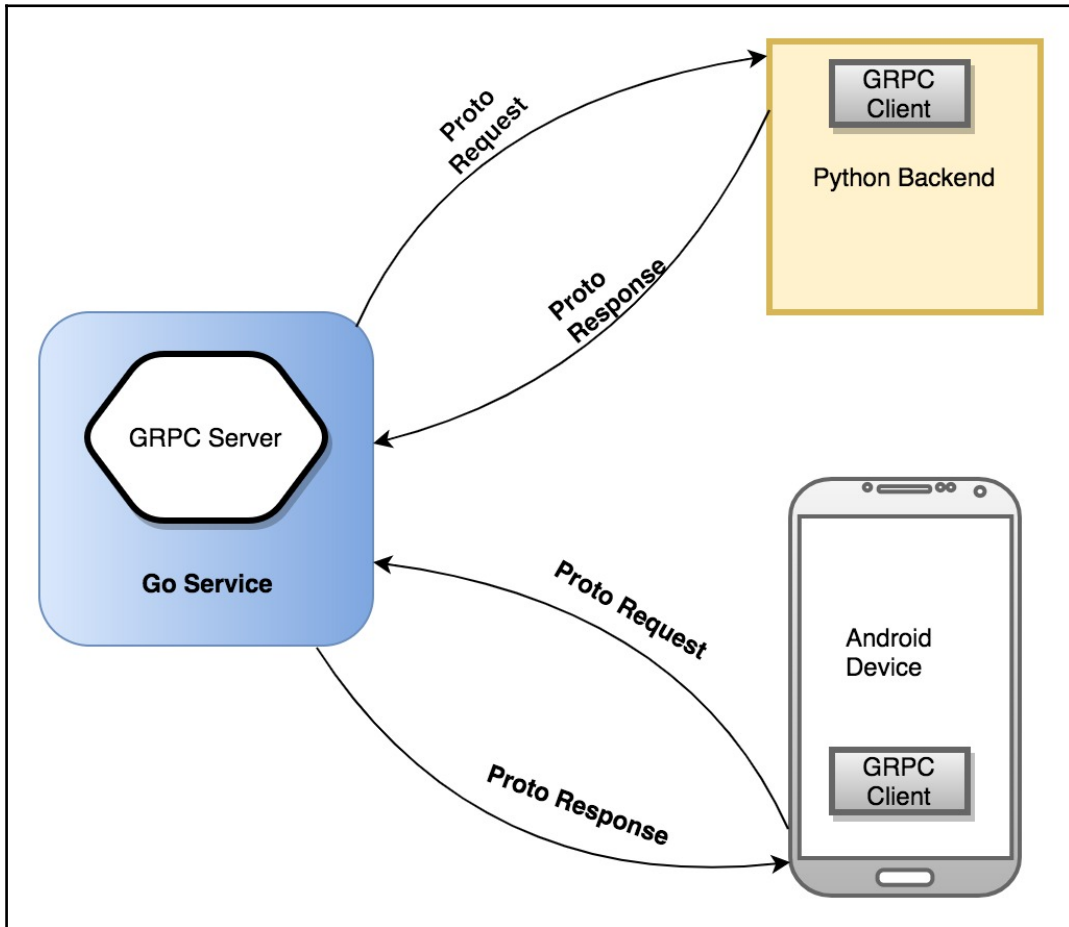
Chapter 5: Working with MongoDB and Go to Create REST APIs

```
[10:03:52] naren:~ $ mongod
2017-07-01T10:03:58.131+0530 I CONTROL [initandlisten] MongoDB starting : pid=86550 port=27017 dbpath=/data/db 64-bit host=Narens-MacBook-Air.local
2017-07-01T10:03:58.131+0530 I CONTROL [initandlisten] db version v3.4.5
2017-07-01T10:03:58.131+0530 I CONTROL [initandlisten] git version: 520b8f3092c48d934f0cd78ab5f40fe594f96863
2017-07-01T10:03:58.131+0530 I CONTROL [initandlisten] OpenSSL version: OpenSSL 1.0.2l 25 May 2017
2017-07-01T10:03:58.131+0530 I CONTROL [initandlisten] allocator: system
2017-07-01T10:03:58.131+0530 I CONTROL [initandlisten] modules: none
2017-07-01T10:03:58.131+0530 I CONTROL [initandlisten] build environment:
2017-07-01T10:03:58.131+0530 I CONTROL [initandlisten] distarch: x86_64
2017-07-01T10:03:58.131+0530 I CONTROL [initandlisten] target_arch: x86_64
```

```
[10:04:05] naren:~ $ mongo
MongoDB shell version v3.4.5
connecting to: mongodb://127.0.0.1:27017
MongoDB server version: 3.4.5
Welcome to the MongoDB shell.
For interactive help, type "help".
For more comprehensive documentation, see
  http://docs.mongodb.org/
Questions? Try the support group
  http://groups.google.com/group/mongodb-user
Server has startup warnings:
2017-07-01T10:03:58.675+0530 I CONTROL [initandlisten]
2017-07-01T10:03:58.675+0530 I CONTROL [initandlisten] ** WARNING: Access control is not enabled for the database.
2017-07-01T10:03:58.675+0530 I CONTROL [initandlisten] **      Read and write access to data and configuration is unrestricted.
2017-07-01T10:03:58.675+0530 I CONTROL [initandlisten]
> █
```

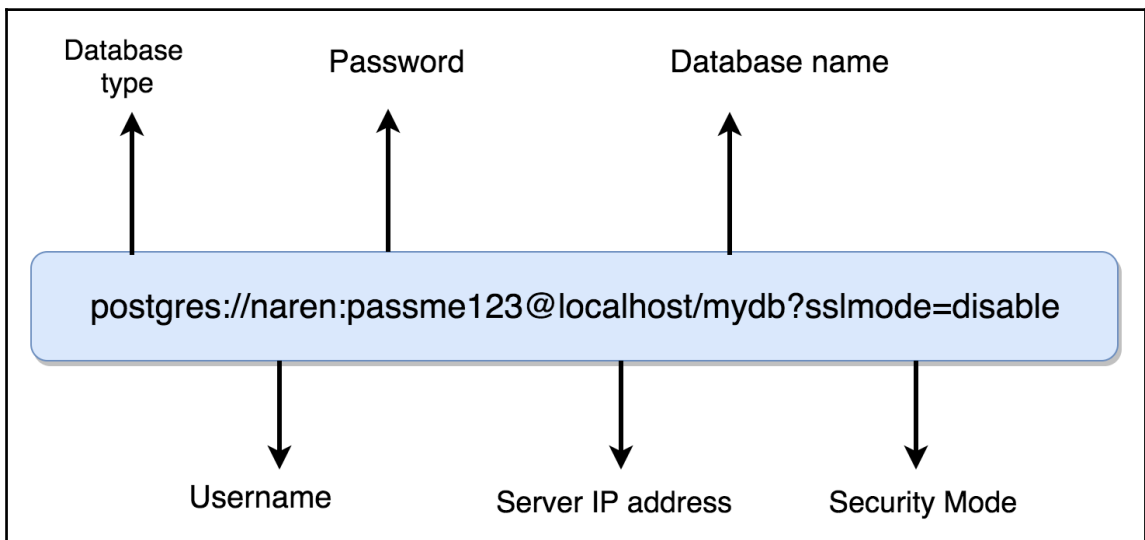


Chapter 6: Working with Protocol Buffers and GRPC

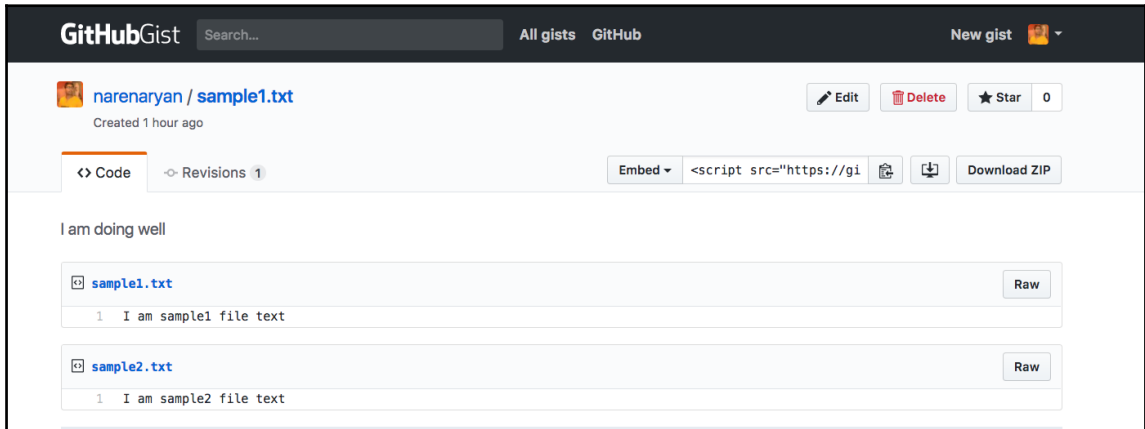
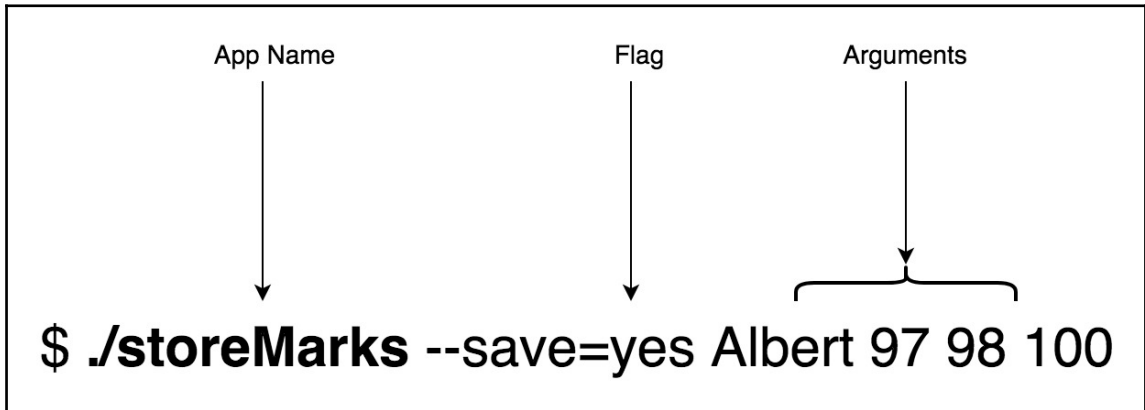


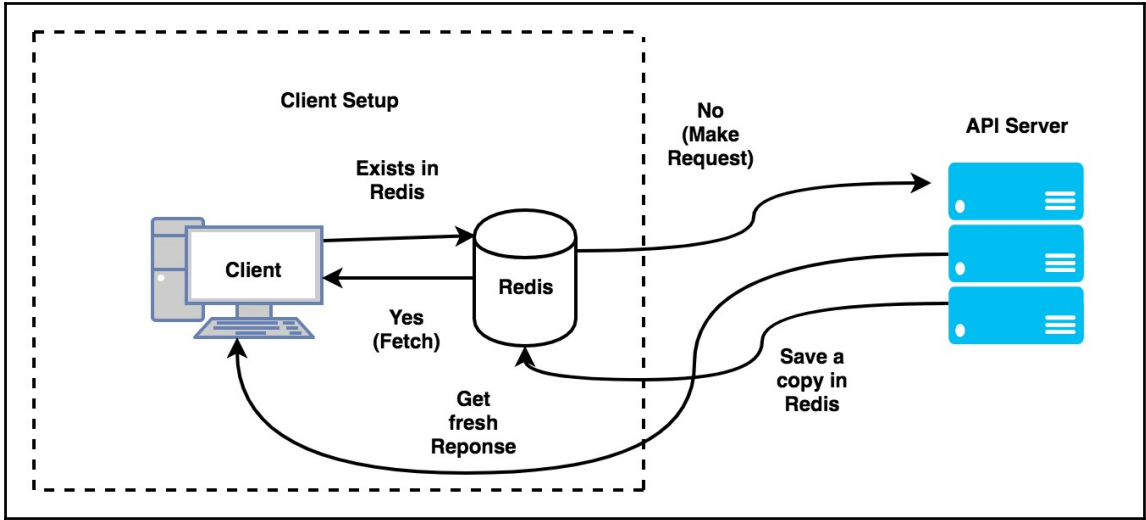
Chapter 7: Working with PostgreSQL, JSON, and Go

```
Informational
(options: S = show system objects, + = additional detail)
\d[S+]                list tables, views, and sequences
\d[S+] NAME          describe table, view, sequence, or index
\da[S] [PATTERN]    list aggregates
\dA[+] [PATTERN]    list access methods
\db[+] [PATTERN]    list tablespaces
\dc[S+] [PATTERN]   list conversions
\dC[+] [PATTERN]    list casts
\dd[S] [PATTERN]    show object descriptions not displayed elsewhere
\ddp [PATTERN]      list default privileges
\dD[S+] [PATTERN]   list domains
\det[+] [PATTERN]   list foreign tables
\des[+] [PATTERN]   list foreign servers
\deu[+] [PATTERN]   list user mappings
```

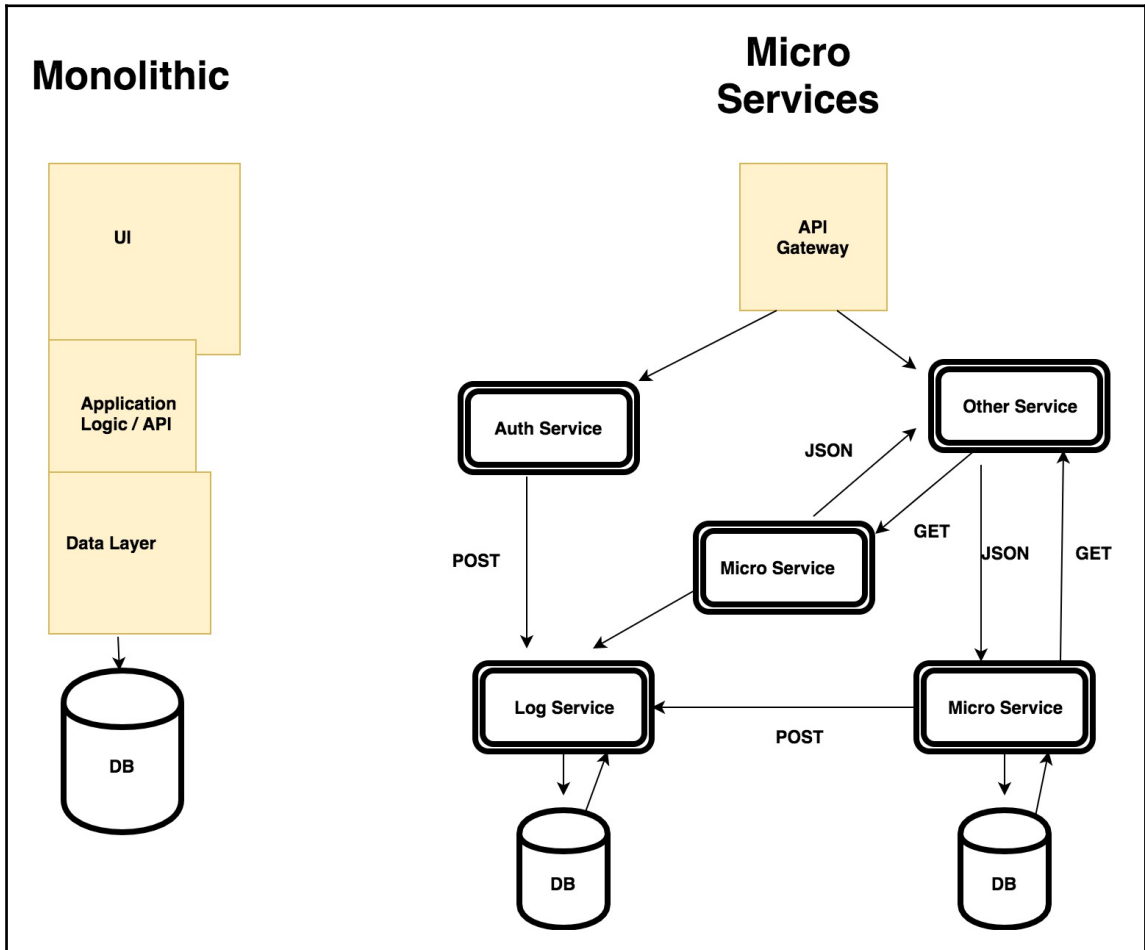


Chapter 8: Building a REST API Client in Go and Unit Testing





Chapter 9: Scaling Our REST API Using Microservices



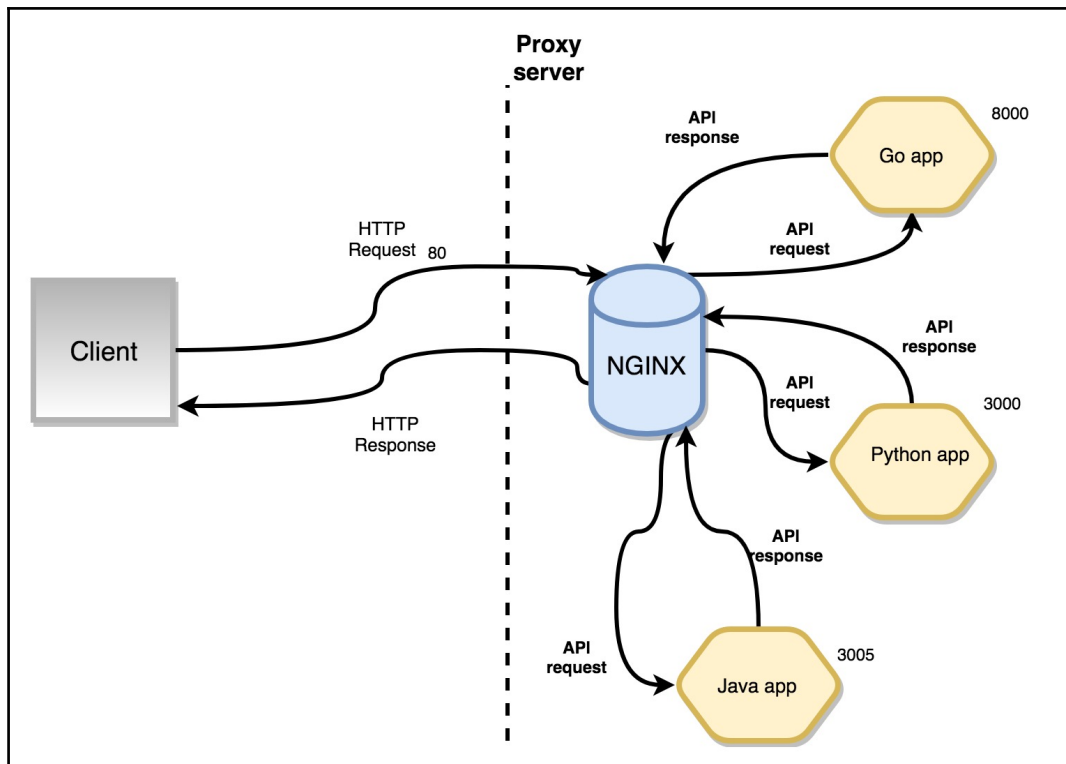
Chapter 10: Deploying Our REST services

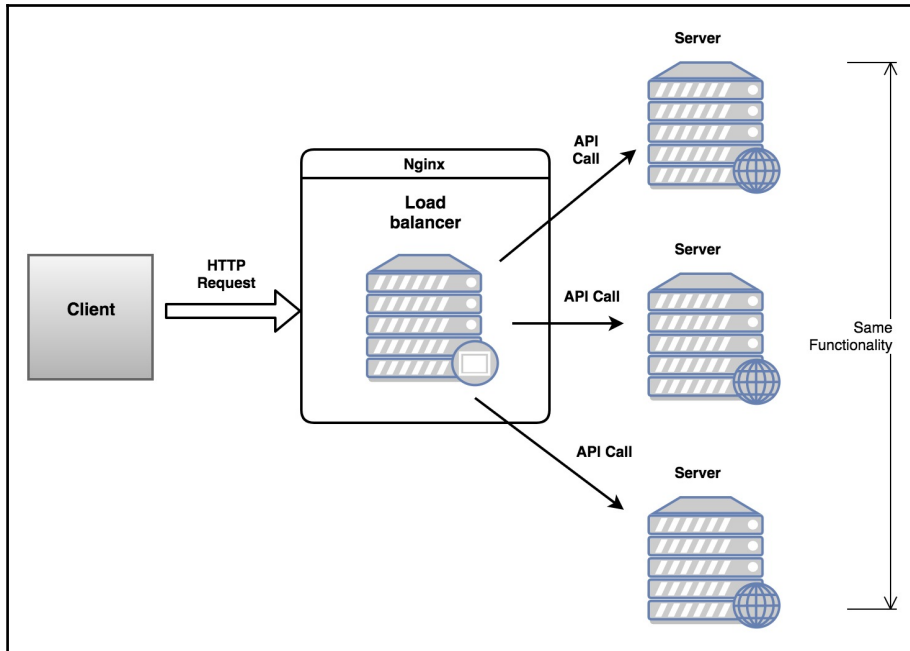
Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.





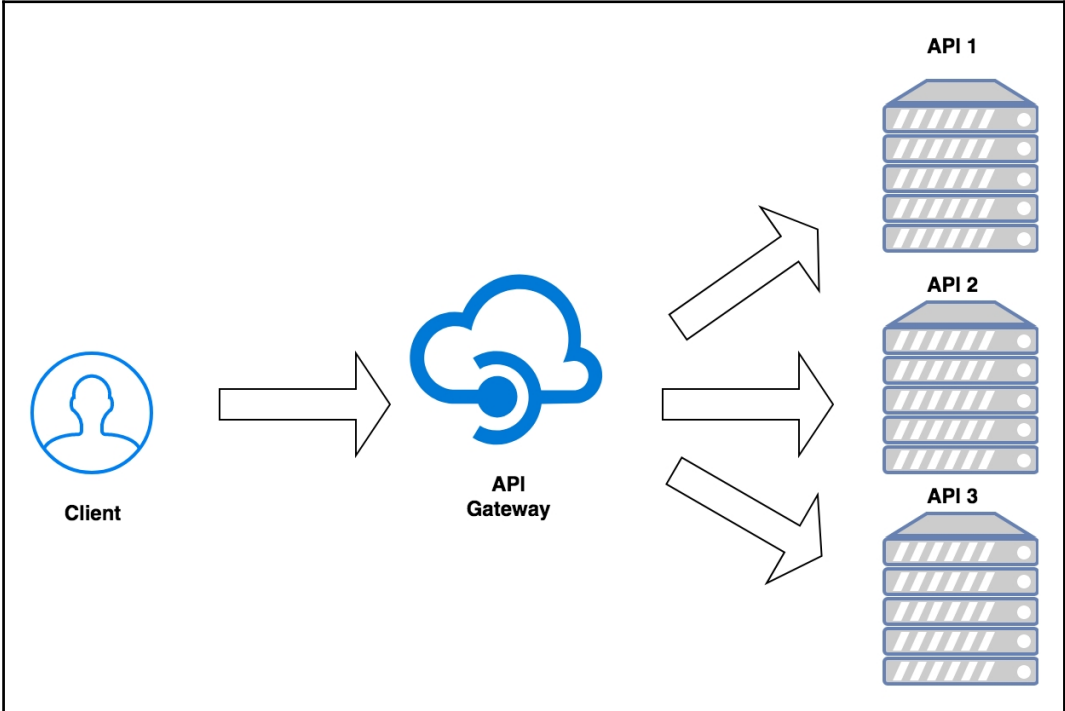
```
root@ubuntu:~# supervisorctl
myserver                RUNNING    pid 6886, uptime 0:00:47
supervisor> help

default commands (type help <topic>):
=====
add      clear  fg      open   quit   remove restart start  stop  update
avail   exit   maintail pid    reload reread shutdown status tail  version
supervisor> █
```

```
root@ubuntu:~# kill 6886
root@ubuntu:~# supervisorctl
myserver                                RUNNING    pid 6903, uptime 0:00:10
supervisor> help

default commands (type help <topic>):
```

Chapter 11: Using an API Gateway to Monitor and Metricize REST API



Kong Add API

POST localhost:8001/apis Params Send Save

Authorization Headers (1) Body Pre-request Script Tests Cookies Code

form-data x-www-form-urlencoded raw binary JSON (application/json)

```
1 {
2   "name": "myapi",
3   "hosts": "server1",
4   "upstream_url": "http://go-server:3000",
5   "uris":["/api/v1"],
6   "strip_uri": true,
7   "preserve_host": false
8 }
```

Get healthcheck x + No Environment

GET localhost:8000/api/v1/healthcheck Params Send Save

Pretty Raw Preview Text Save Response

```
1 2017-10-28 13:18:59.458568988 +0000 UTC m=+4263.316715654
```

Enable file logging plugin

POST localhost:8001/apis/myapi/plugins Params Send Save

Authorization Headers (1) Body Pre-request Script Tests Cookies Code

form-data x-www-form-urlencoded raw binary JSON (application/json)

```
1 {
2   "name": "file-log",
3   "config.path": "/tmp/file.log"
4 }
5 }
```

▶ Kong enable key auth

POST http://localhost:8001/apis/myapi/plugins Params Send Save

Authorization Headers (1) Body Pre-request Script Tests Cookies Code

form-data x-www-form-urlencoded raw binary JSON (application/json)

```
1 - {
2   "name" : "key-auth",
3   "config.hide_credentials" : true
4 }
```

▶ Kong create consumer

POST http://localhost:8001/consumers/ Params Send Save

Authorization Headers (1) Body Pre-request Script Tests Cookies Code

form-data x-www-form-urlencoded raw binary JSON (application/json)

```
1 - {
2   "username" : "johnd"
3 }
```

▶ Kong Create API Key for Consumer

POST http://localhost:8001/consumers/johnd/key-auth Params Send Save

Authorization Headers (1) Body Pre-request Script Tests Cookies Code

form-data x-www-form-urlencoded raw binary Text

```
1
```

Get healthcheck

GET localhost:8000/api/v1/healthcheck Params Send Save

Authorization Headers (2) Body Pre-request Script Tests Cookies Code

Key	Value	Bulk Edit	Presets
<input checked="" type="checkbox"/> Host	server1		
<input checked="" type="checkbox"/> apikey	89MH58EXzc4xHBO8WZB9axZ4uhZ1vW9d		
New key	value		

Body Cookies (1) Headers (7) Tests Status: 200 OK Time: 65 ms Size: 271 B

Pretty Raw Preview Text Save Response

```
1 2017-10-28 18:24:49.357110452 +0000 UTC m=+21460.589271643|
```

Kong rate limit consumer

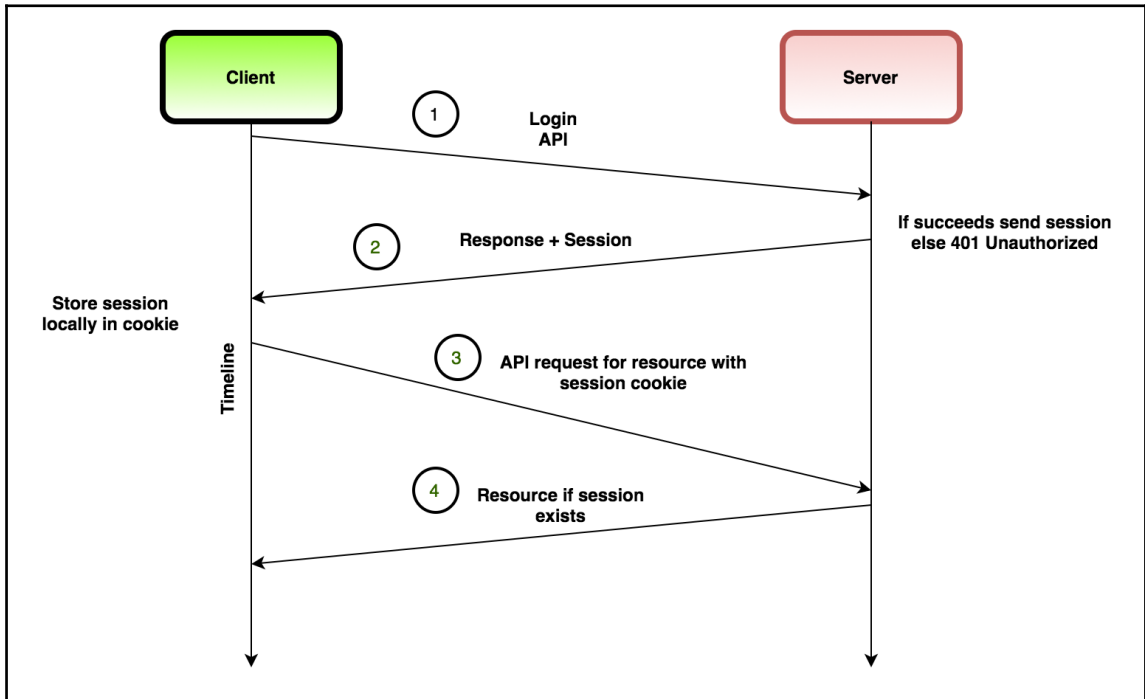
POST http://localhost:8001/apis/myapi/plugins Params Send Save

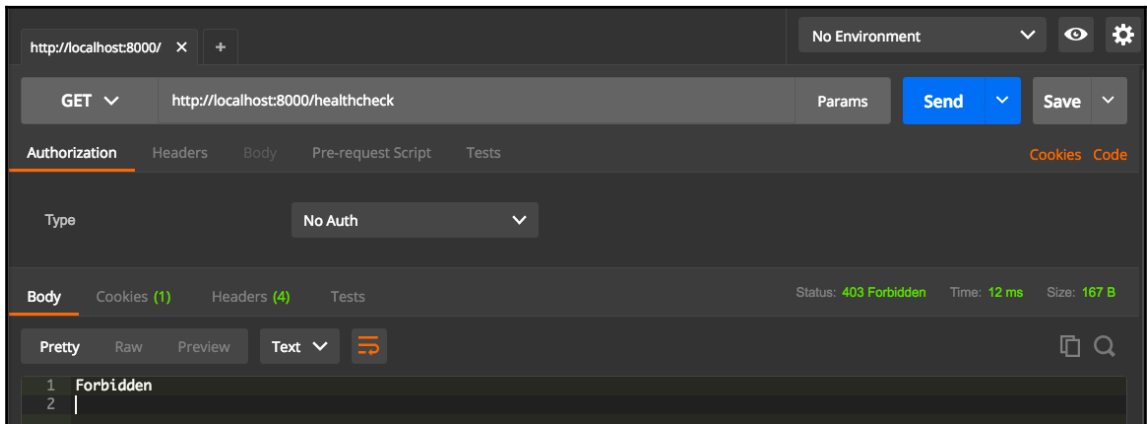
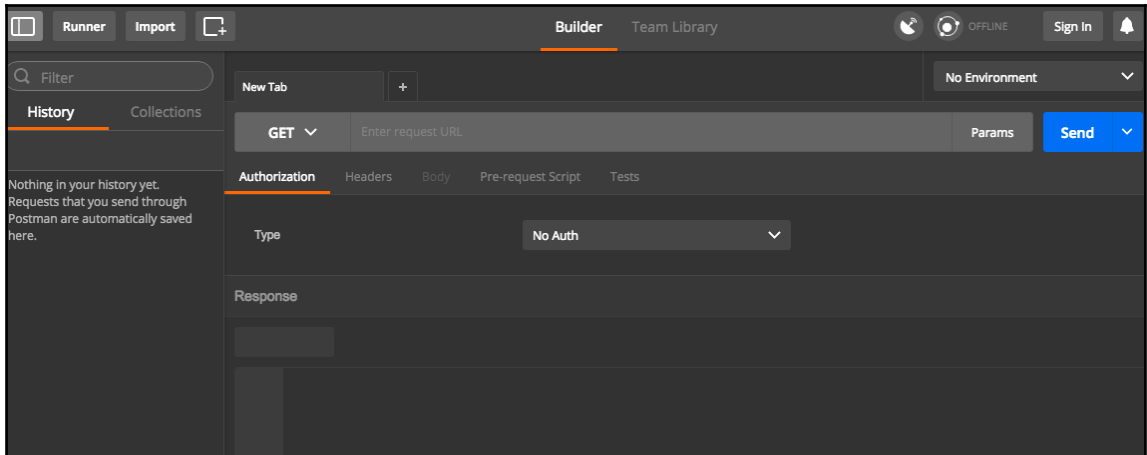
Authorization Headers (1) Body Pre-request Script Tests Cookies Code

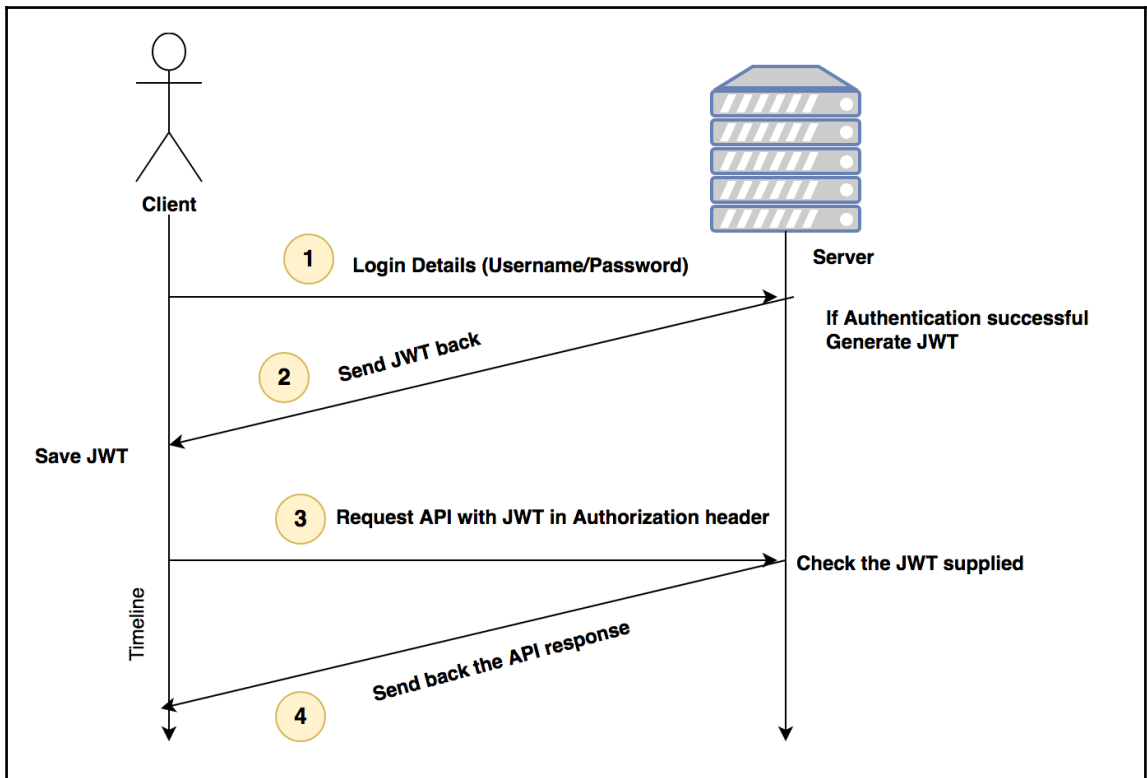
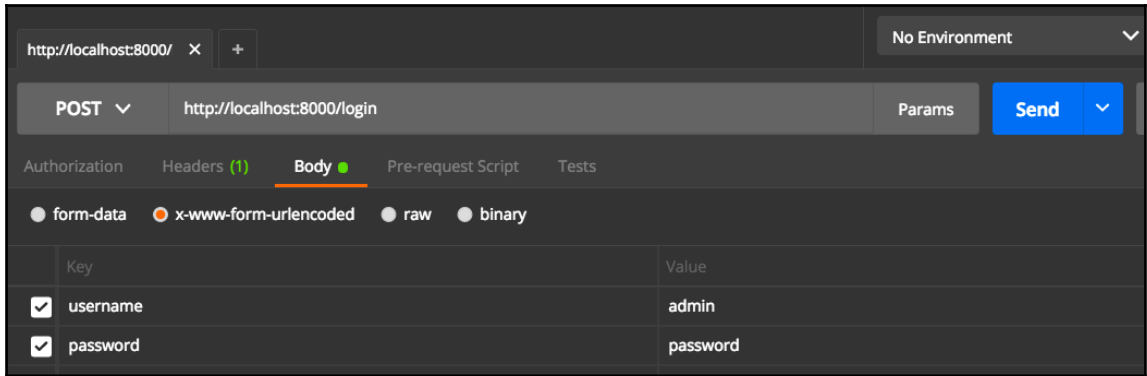
form-data x-www-form-urlencoded raw binary JSON (application/json)

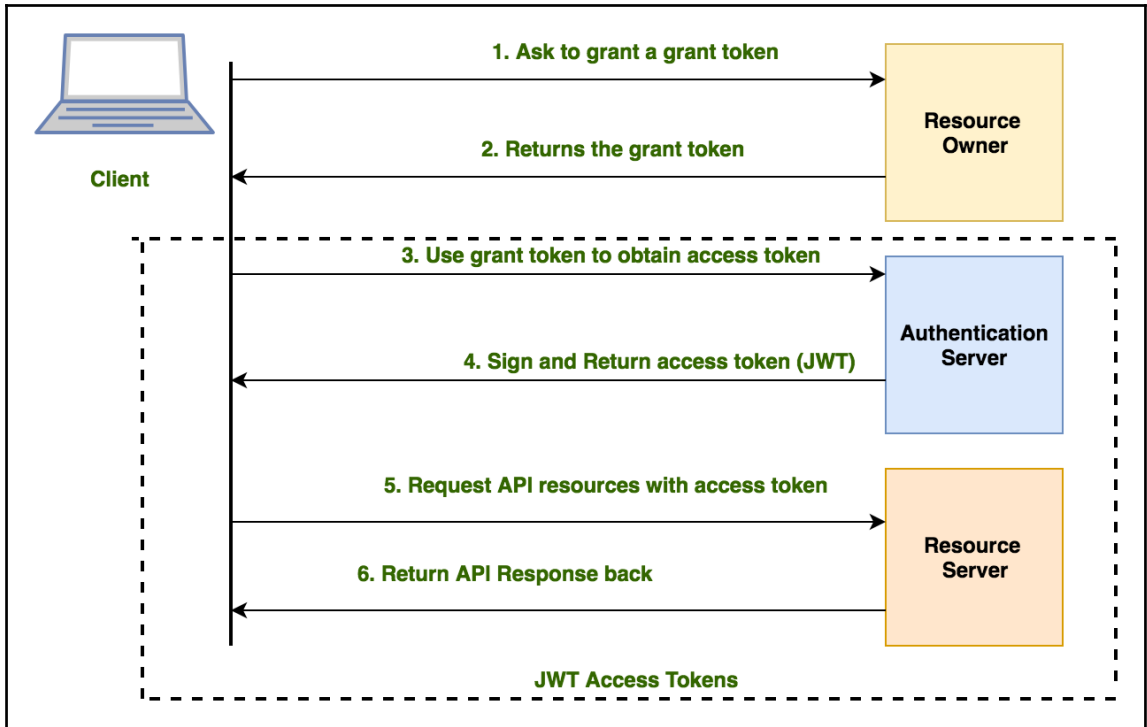
```
1 - {
2   "name" : "rate-limiting",
3   "config.hour" : "5000",
4   "consumer_id" : "df024acb-5cbd-4e4d-b3ed-751287eafd36"
5 }
```

Chapter 12: Handling Authentication for Our REST Services









http://localhost:8000/ x + No Environment

POST http://localhost:8000/getToken Params Send Save

Authorization Headers (1) Body Pre-request Script Tests Cookies Code

form-data x-www-form-urlencoded raw binary

Key	Value	Bulk Edit
<input checked="" type="checkbox"/> username	naren	
<input checked="" type="checkbox"/> password	passme	
New key	value	

Body Cookies (1) Headers (3) Tests Status: 200 OK Time: 29 ms Size: 310 B

Pretty Raw Preview Text

```
1 [{"token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJFeHBpcmVzQXQiOiJlMDAwLkZjc3N1ZWRBdCI6MTUwODc1OTM4NywiOiJ1YXJ1biJ9.uCGXyWCV02EShjie2NzNpV8LqvvkP541YZR5wofnT1E", "status": "success"}]
```

http://localhost:8000/ x + No Environment

GET http://localhost:8000/healthcheck Params Send Save

Authorization Headers (1) Body Pre-request Script Tests Cookies Code

Key	Value	Bulk Edit	Presets
<input checked="" type="checkbox"/> access_token	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJFeHBpcmVzQXQiOiJlMDAwLkZjc3N1ZWRBdCI6MTUwODc1OTM4NywiOiJ1YXJ1biJ9.uCGXyWCV02EShjie2NzNpV8LqvvkP541YZR5wofnT1E		
New key	value		

Body Cookies (1) Headers (3) Tests Status: 200 OK Time: 25 ms Size: 182 B

Pretty Raw Preview Text

```
1 [{"time": "2017-10-23 17:21:50.724864162 +0530 IST", "user": "naren"}]
```