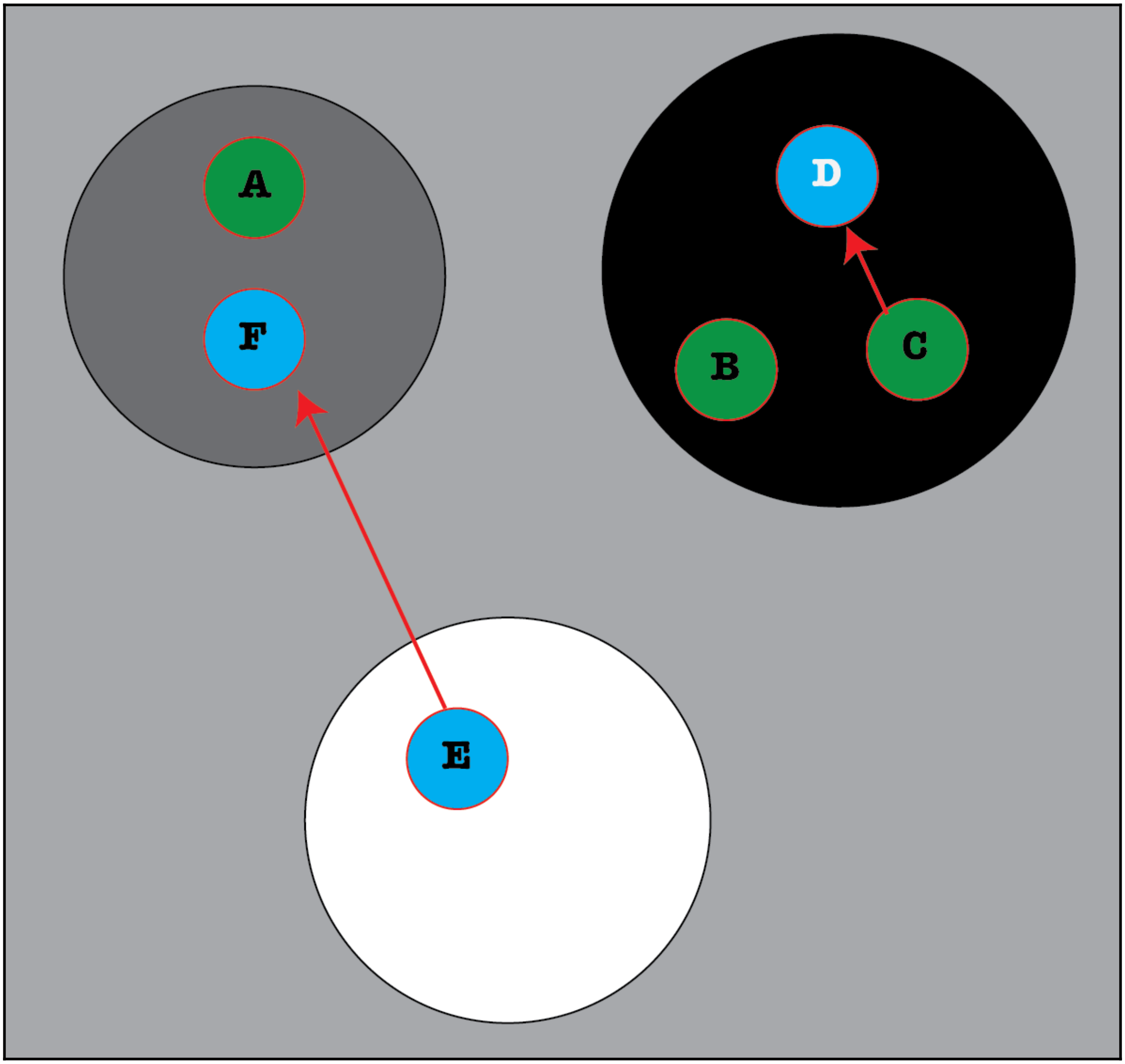
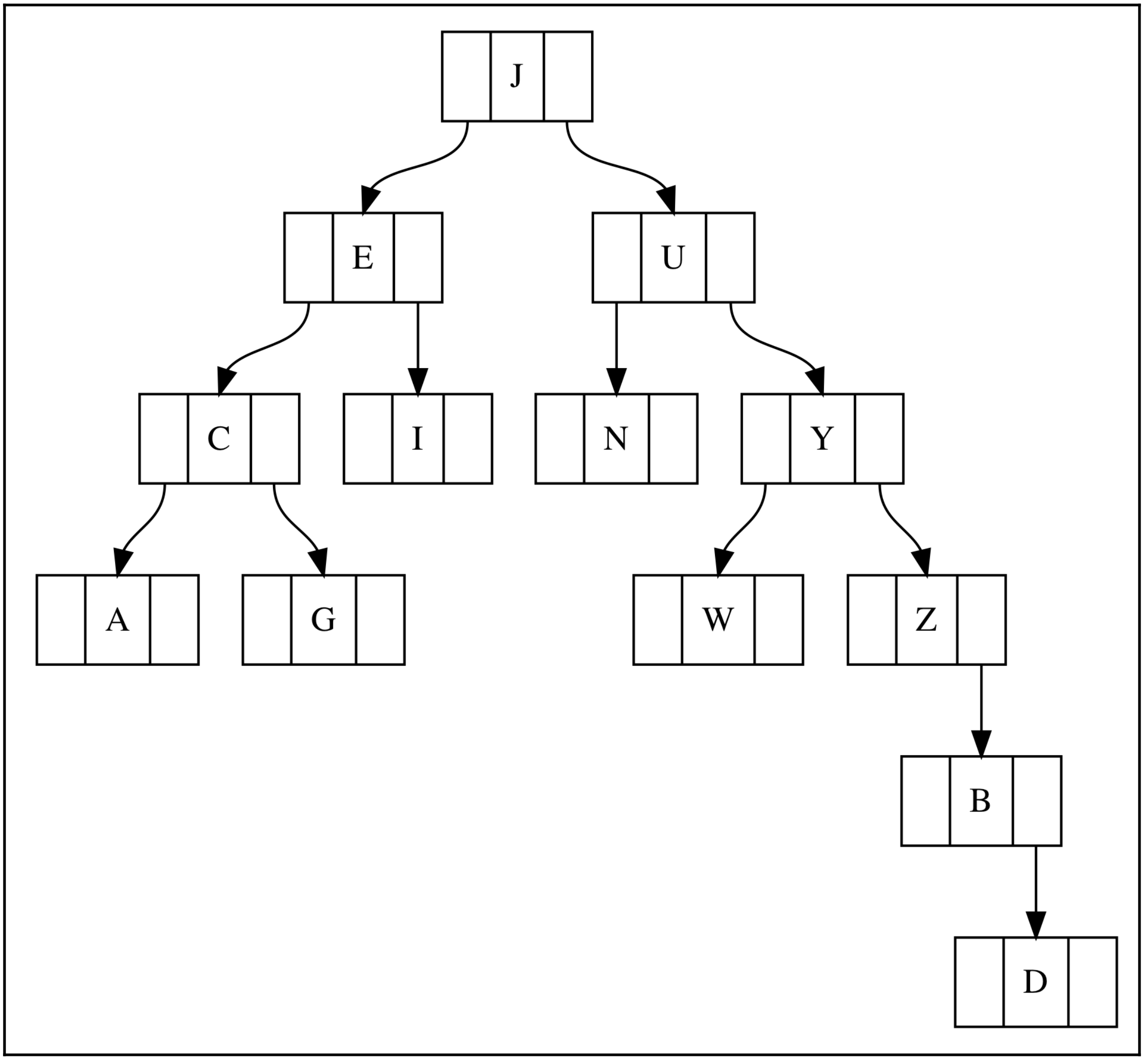
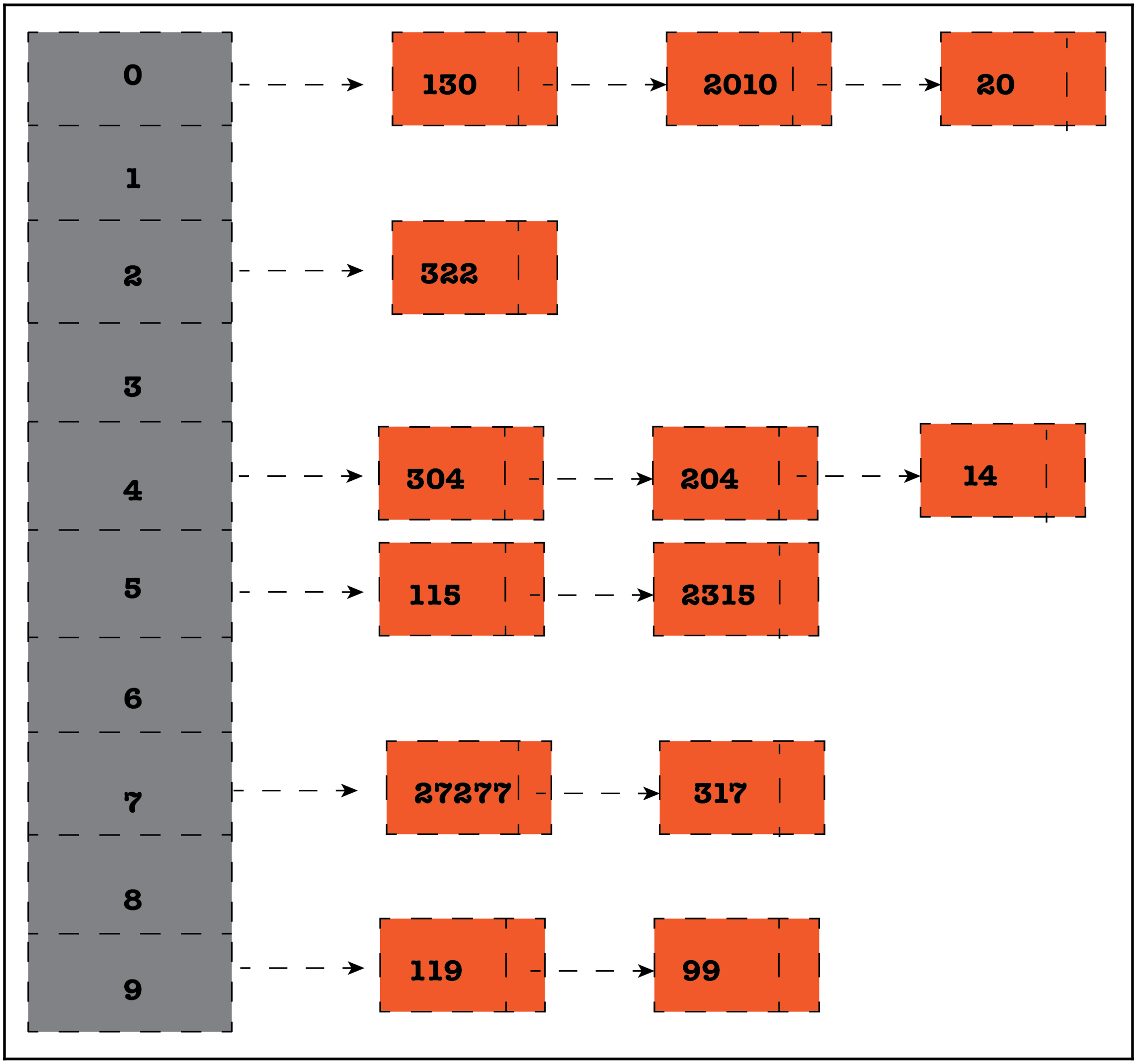
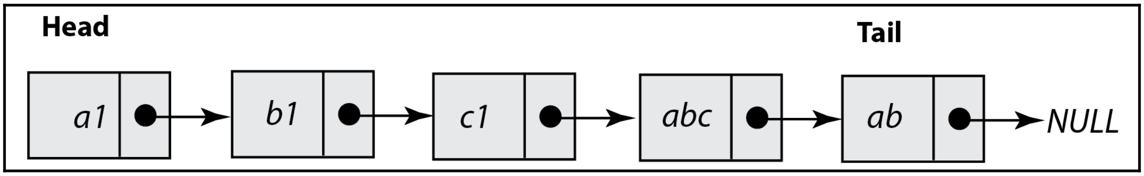
Chapter 2: Understanding Go Internals

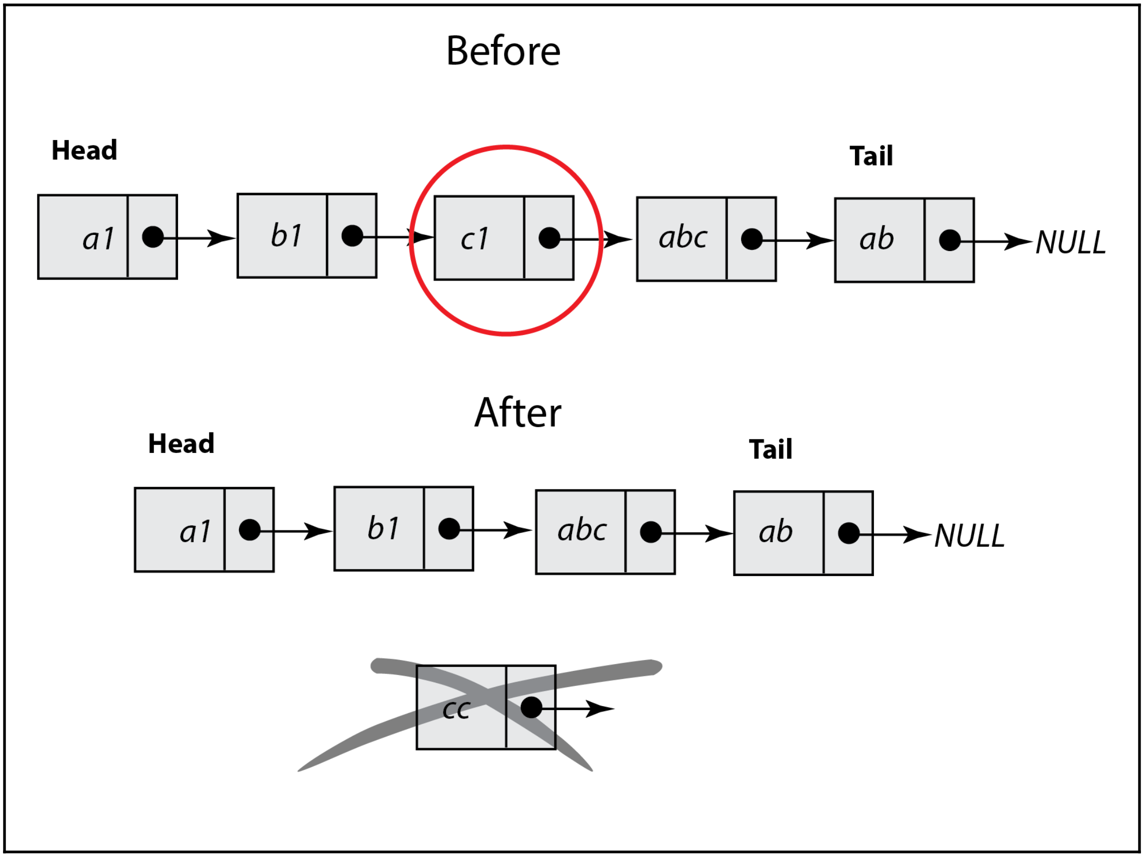


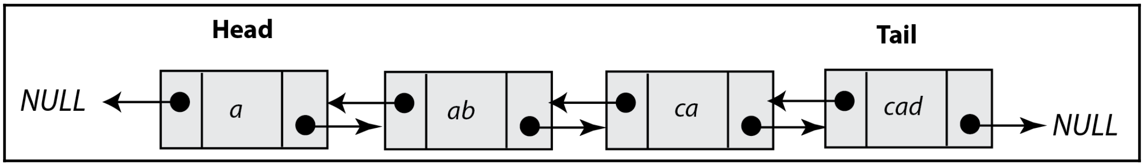
Chapter 5: How to Enhance Go Code with Data Structures

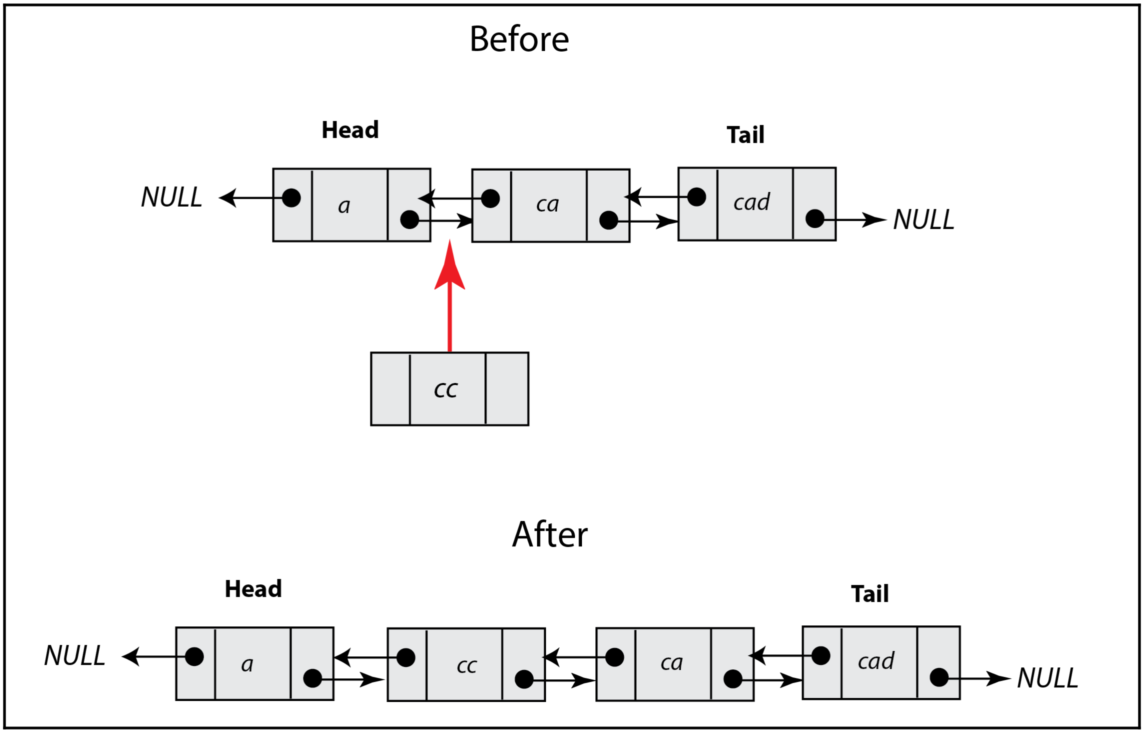



0
1
2
3
4
5
6
7
8
9
130
2010
20
322
304
204
14
115
2315
27277
317
119
99

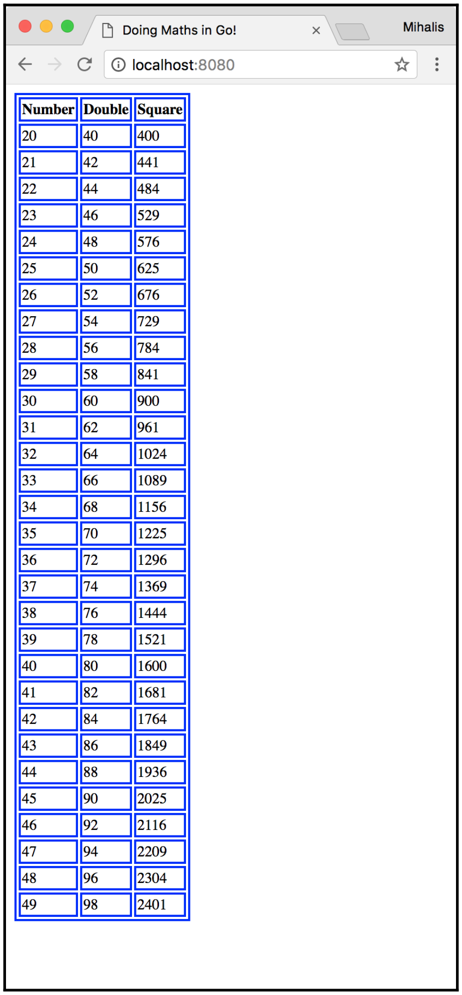

Head
Tail
a1
b1
c1
abc
ab
NULL


Before
Head
Tail
a1
b1
c1
abc
ab
NULL
After
Head
Tail
a1
b1
abc
ab
NULL
cc


Head
Tail
NULL
a
ab
ca
cad
NULL

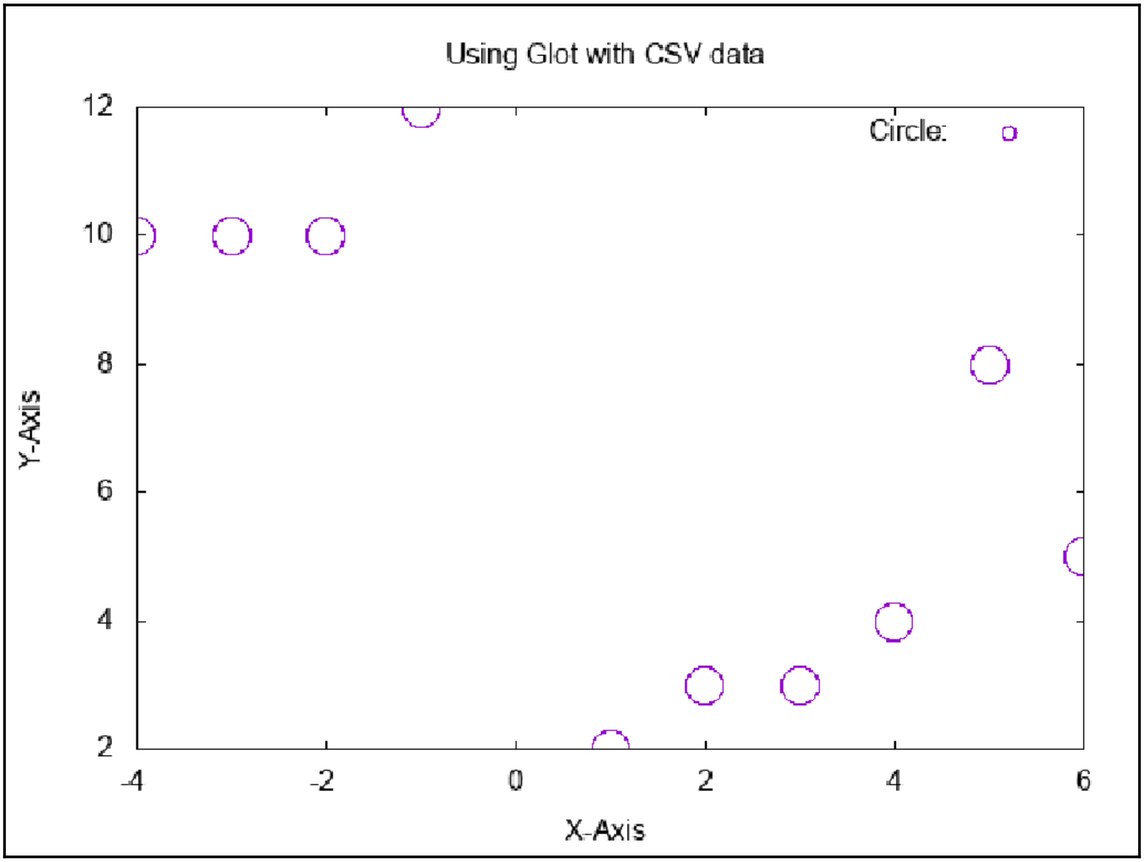

Before
Head
Tail
NULL
a
ca
cad
NULL
cc
After
Head
Tail
NULL
a
cc
ca
cad
NULL

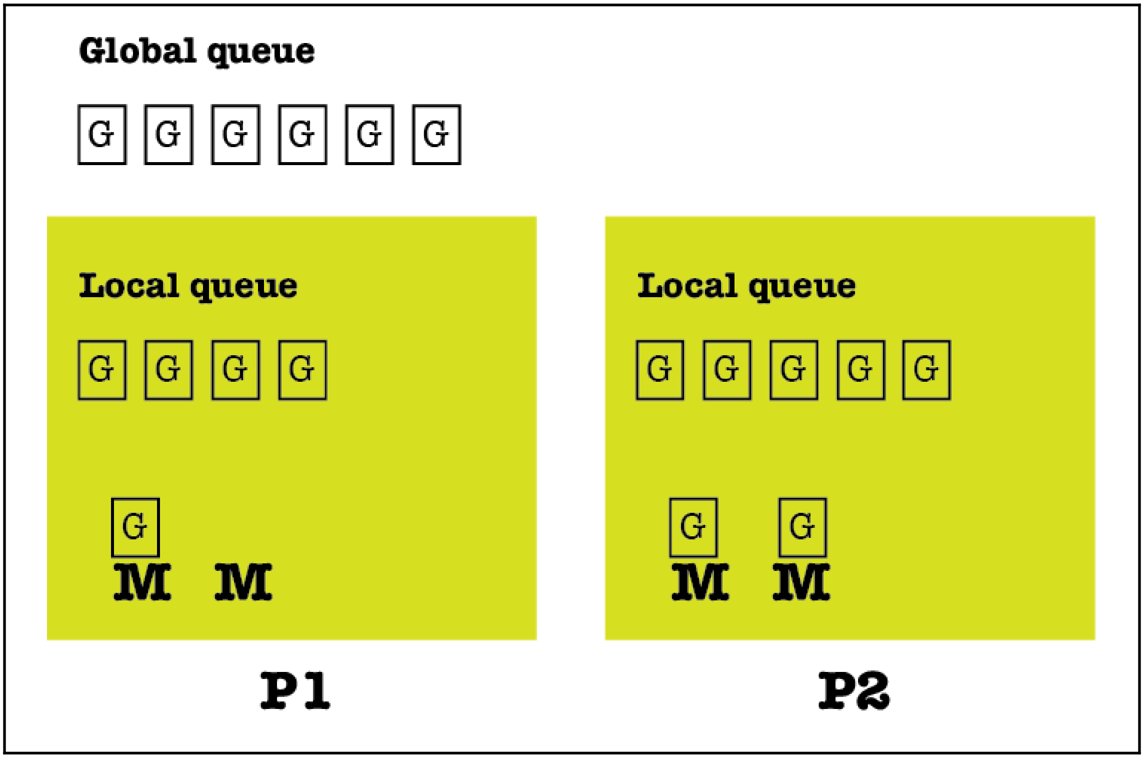
Chapter 6: What You Might Not Know About Go Packages



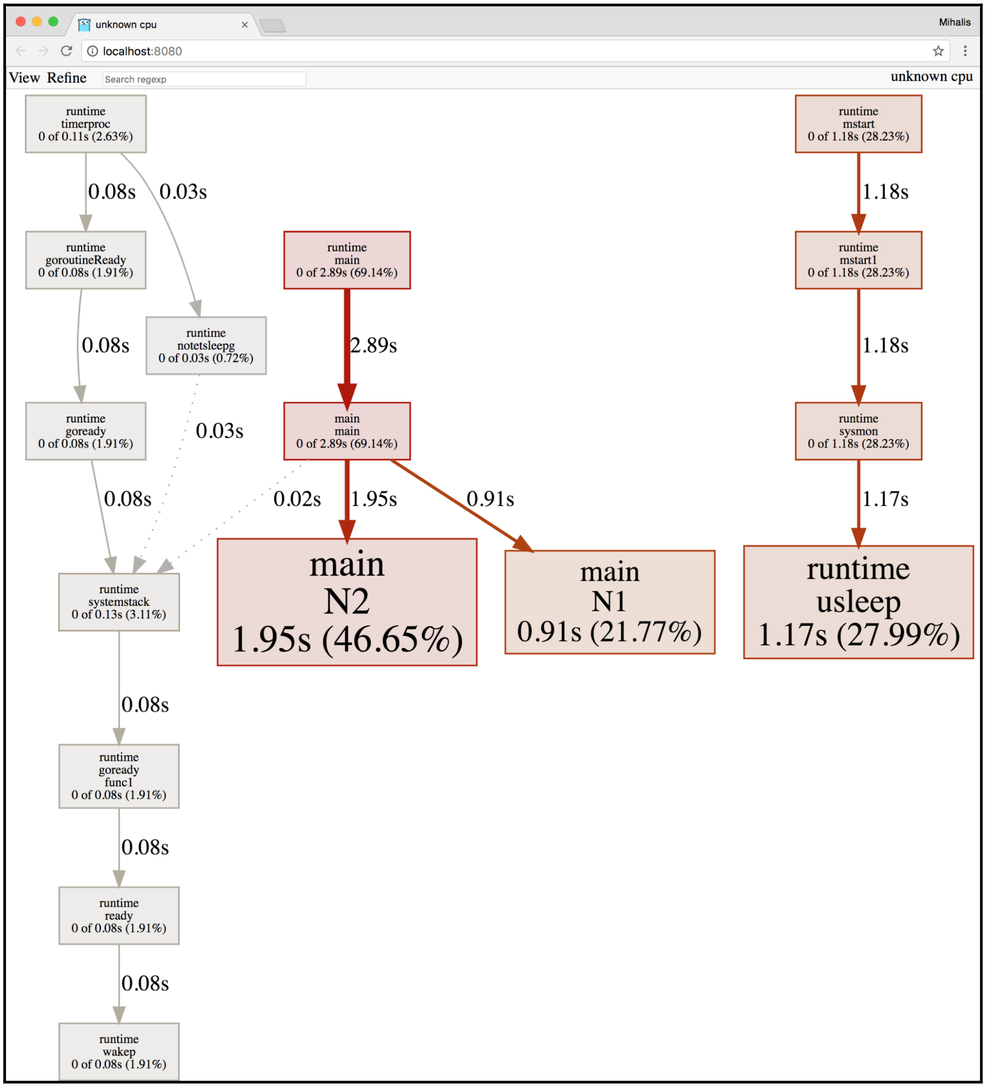
Number	Double	Square
20	40	400
21	42	441
22	44	484
23	46	529
24	48	576
25	50	625
26	52	676
27	54	729
28	56	784
29	58	841
30	60	900
31	62	961
32	64	1024
33	66	1089
34	68	1156
35	70	1225
36	72	1296
37	74	1369
38	76	1444
39	78	1521
40	80	1600
41	82	1681
42	84	1764
43	86	1849
44	88	1936
45	90	2025
46	92	2116
47	94	2209
48	96	2304
49	98	2401

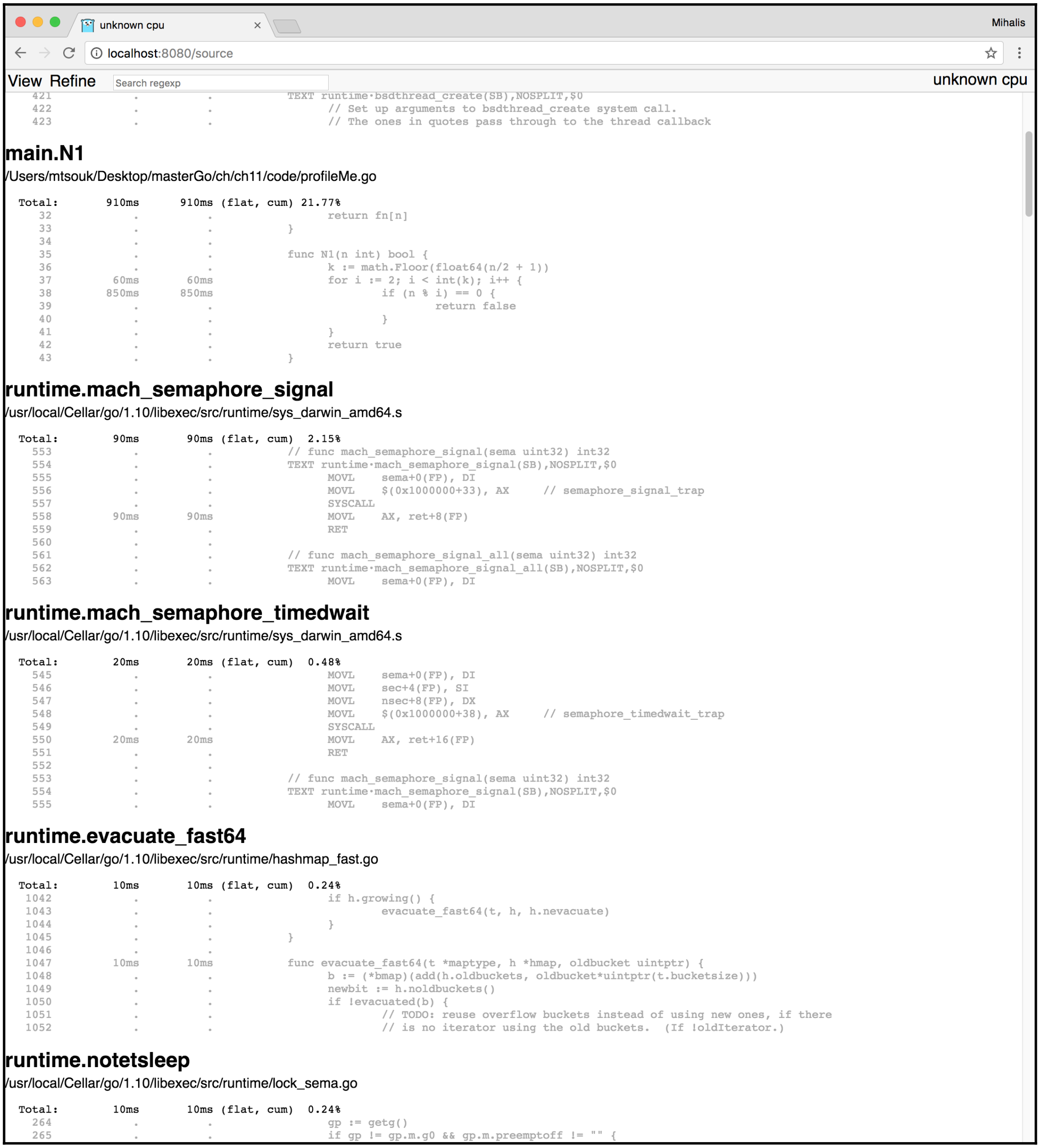
Chapter 8: Telling a Unix System What to Do




Global queue
G
G
G
G
G
G
Local queue
G
G
G
G
G
M
M
P1
Local queue
G
G
G
G
G
G
G
M
M
P2

Chapter 11: Code Testing, Optimization, and Profiling


unknown cpu
Mihalis
localhost:8080
View Refine
Search regexp
unknown cpu
runtime timerproc 0 of 0.11s (2.63%)
0.08s
0.03s
runtime goroutineReady 0 of 0.08s (1.91%)
0.08s
runtime notetsleepg 0 of 0.03s (0.72%)
0.03s
runtime goready 0 of 0.08s (1.91%)
0.08s
runtime systemstack 0 of 0.13s (3.11%)
0.08s
runtime goready func1 0 of 0.08s (1.91%)
0.08s
runtime ready 0 of 0.08s (1.91%)
0.08s
runtime wakep 0 of 0.08s (1.91%)
runtime main 0 of 2.89s (69.14%)
2.89s
main main 0 of 2.89s (69.14%)
0.02s
1.95s
0.91s
main N2 1.95s (46.65%)
main N1 0.91s (21.77%)
runtime mstart 0 of 1.18s (28.23%)
1.18s
runtime mstart1 0 of 1.18s (28.23%)
1.18s
runtime sysmon 0 of 1.18s (28.23%)
1.17s
runtime usleep 1.17s (27.99%)



```
  421            .          .           TEXT runtime·bsdthread_create(SB),NOSPLIT,$0
  422            .          .                 // Set up arguments to bsdthread_create system call.
  423            .          .                 // The ones in quotes pass through to the thread callback
```

main.N1

/Users/mtsouk/Desktop/masterGo/ch/ch11/code/profileMe.go

```
Total:       910ms      910ms (flat, cum) 21.77%
   32            .          .                 return fn[n]
   33            .          .           }
   34            .          .
   35            .          .           func N1(n int) bool {
   36            .          .                 k := math.Floor(float64(n/2 + 1))
   37         60ms       60ms                 for i := 2; i < int(k); i++ {
   38        850ms      850ms                         if (n % i) == 0 {
   39            .          .                                 return false
   40            .          .                         }
   41            .          .                 }
   42            .          .                 return true
   43            .          .           }
```

runtime.mach_semaphore_signal

/usr/local/Cellar/go/1.10/libexec/src/runtime/sys_darwin_amd64.s

```
Total:        90ms       90ms (flat, cum)  2.15%
  553            .          .           // func mach_semaphore_signal(sema uint32) int32
  554            .          .           TEXT runtime·mach_semaphore_signal(SB),NOSPLIT,$0
  555            .          .                 MOVL    sema+0(FP), DI
  556            .          .                 MOVL    $(0x1000000+33), AX     // semaphore_signal_trap
  557            .          .                 SYSCALL
  558         90ms       90ms                 MOVL    AX, ret+8(FP)
  559            .          .                 RET
  560            .          .
  561            .          .           // func mach_semaphore_signal_all(sema uint32) int32
  562            .          .           TEXT runtime·mach_semaphore_signal_all(SB),NOSPLIT,$0
  563            .          .                 MOVL    sema+0(FP), DI
```

runtime.mach_semaphore_timedwait

/usr/local/Cellar/go/1.10/libexec/src/runtime/sys_darwin_amd64.s

```
Total:        20ms       20ms (flat, cum)  0.48%
  545            .          .                 MOVL    sema+0(FP), DI
  546            .          .                 MOVL    sec+4(FP), SI
  547            .          .                 MOVL    nsec+8(FP), DX
  548            .          .                 MOVL    $(0x1000000+38), AX     // semaphore_timedwait_trap
  549            .          .                 SYSCALL
  550         20ms       20ms                 MOVL    AX, ret+16(FP)
  551            .          .                 RET
  552            .          .
  553            .          .           // func mach_semaphore_signal(sema uint32) int32
  554            .          .           TEXT runtime·mach_semaphore_signal(SB),NOSPLIT,$0
  555            .          .                 MOVL    sema+0(FP), DI
```

runtime.evacuate_fast64

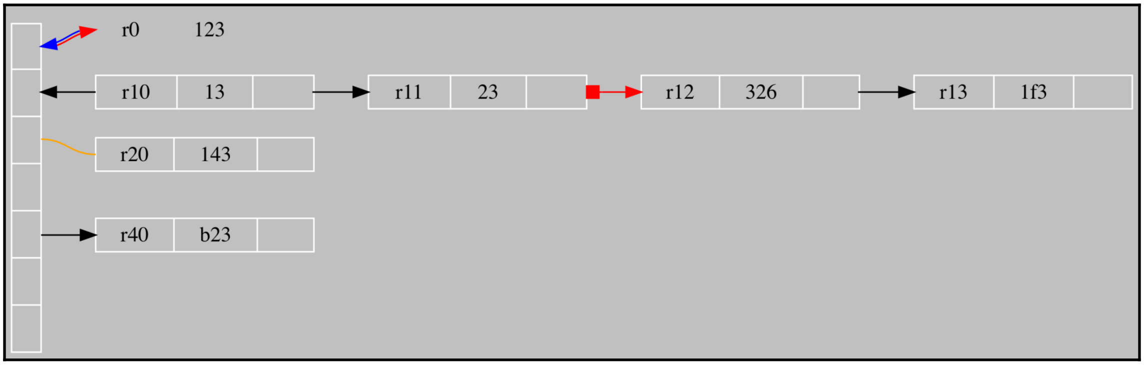
/usr/local/Cellar/go/1.10/libexec/src/runtime/hashmap_fast.go

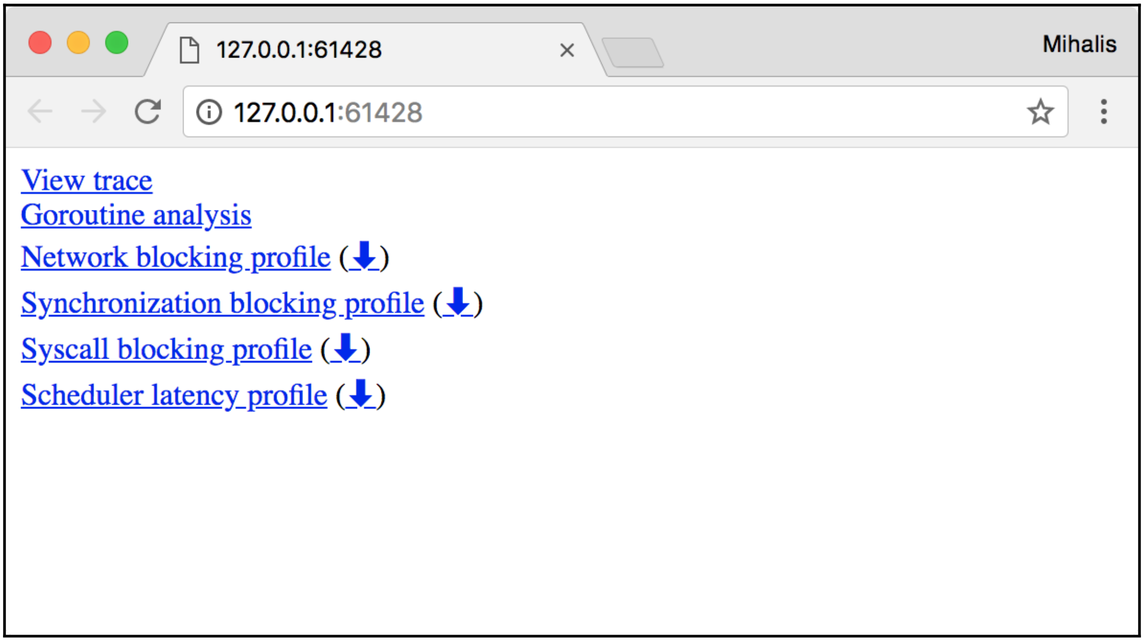
```
Total:        10ms       10ms (flat, cum)  0.24%
 1042            .          .                 if h.growing() {
 1043            .          .                         evacuate_fast64(t, h, h.nevacuate)
 1044            .          .                 }
 1045            .          .           }
 1046            .          .
 1047         10ms       10ms           func evacuate_fast64(t *maptype, h *hmap, oldbucket uintptr) {
 1048            .          .                 b := (*bmap)(add(h.oldbuckets, oldbucket*uintptr(t.bucketsize)))
 1049            .          .                 newbit := h.noldbuckets()
 1050            .          .                 if !evacuated(b) {
 1051            .          .                         // TODO: reuse overflow buckets instead of using new ones, if there
 1052            .          .                         // is no iterator using the old buckets.  (If !oldIterator.)
```

runtime.notetsleep

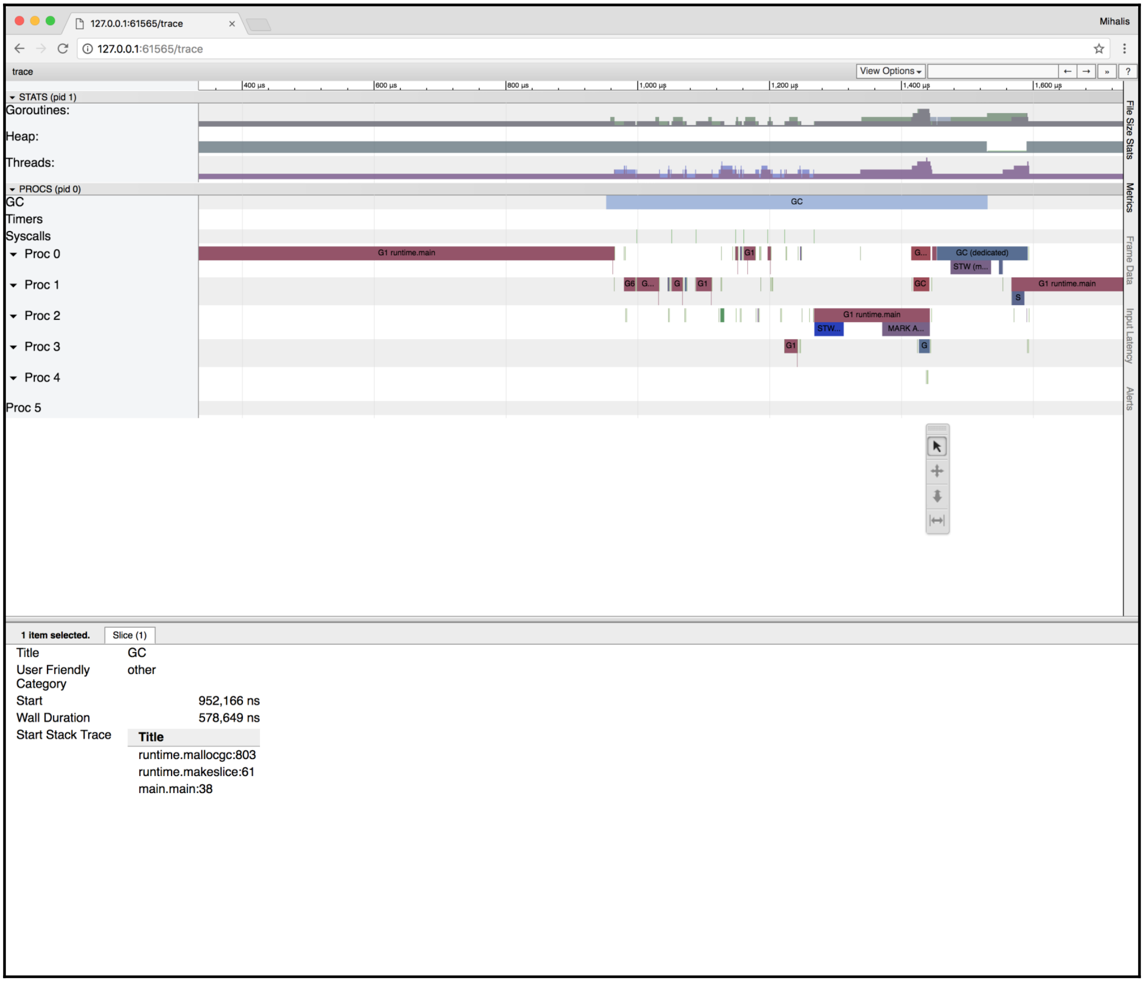
/usr/local/Cellar/go/1.10/libexec/src/runtime/lock_sema.go

```
Total:        10ms       10ms (flat, cum)  0.24%
  264            .          .                 gp := getg()
  265            .          .                 if gp != gp.m.g0 && gp.m.preemptoff != "" {
```



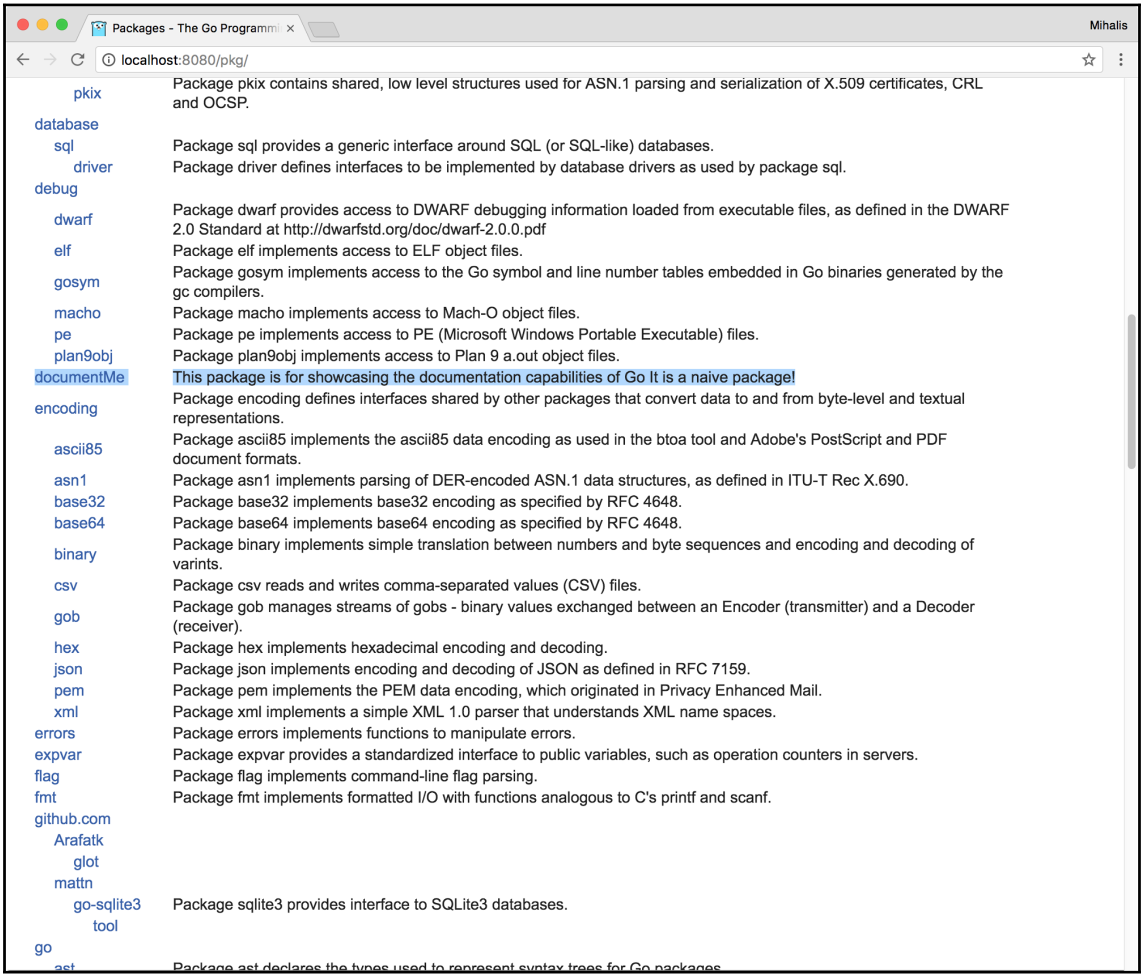


View trace
Goroutine analysis
Network blocking profile (⬇)
Synchronization blocking profile (⬇)
Syscall blocking profile (⬇)
Scheduler latency profile (⬇)

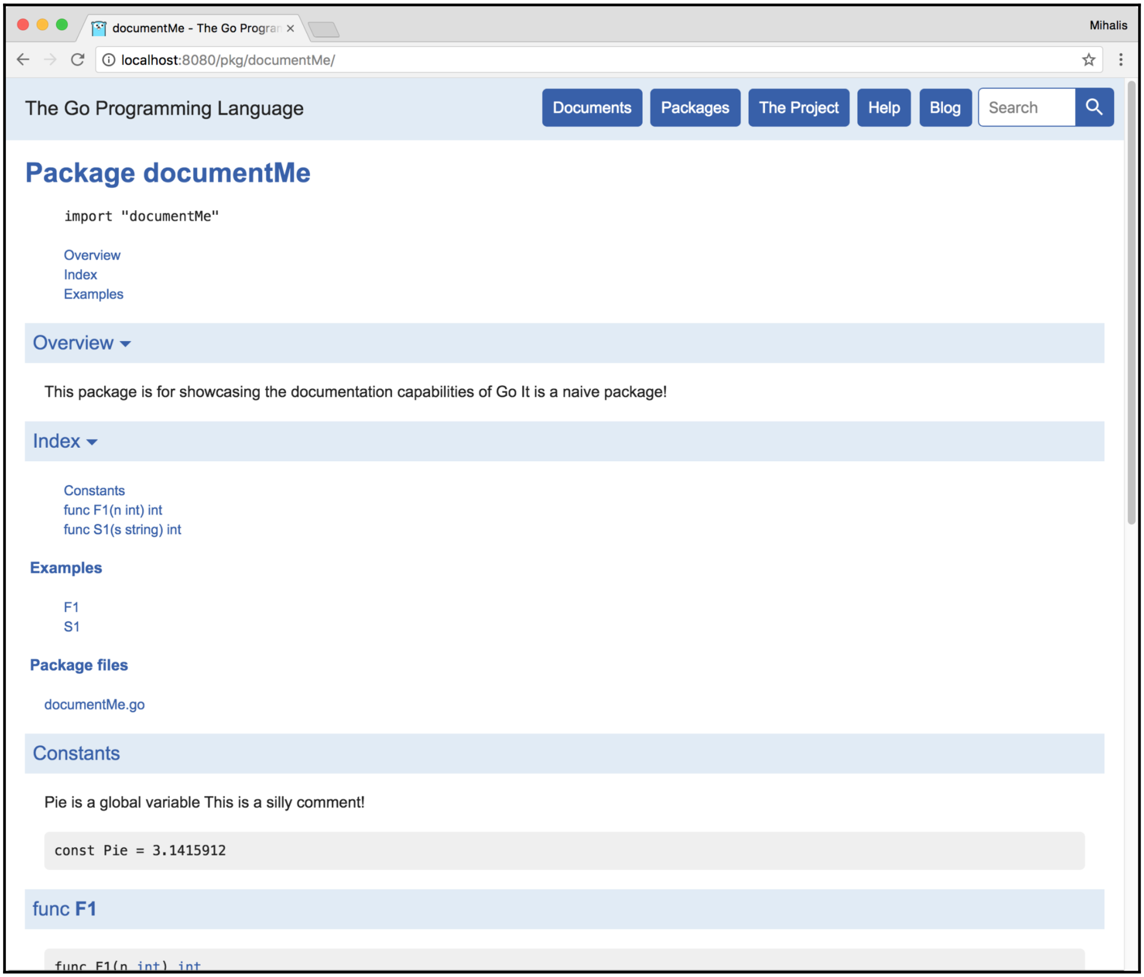


1 item selected. Slice (1)

Title	GC	
User Friendly Category	other	
Start		952,166 ns
Wall Duration		578,649 ns
Start Stack Trace	**Title** runtime.mallocgc:803 runtime.makeslice:61 main.main:38	



pkix	Package pkix contains shared, low level structures used for ASN.1 parsing and serialization of X.509 certificates, CRL and OCSP.
database	
sql	Package sql provides a generic interface around SQL (or SQL-like) databases.
driver	Package driver defines interfaces to be implemented by database drivers as used by package sql.
debug	
dwarf	Package dwarf provides access to DWARF debugging information loaded from executable files, as defined in the DWARF 2.0 Standard at http://dwarfstd.org/doc/dwarf-2.0.0.pdf
elf	Package elf implements access to ELF object files.
gosym	Package gosym implements access to the Go symbol and line number tables embedded in Go binaries generated by the gc compilers.
macho	Package macho implements access to Mach-O object files.
pe	Package pe implements access to PE (Microsoft Windows Portable Executable) files.
plan9obj	Package plan9obj implements access to Plan 9 a.out object files.
documentMe	This package is for showcasing the documentation capabilities of Go It is a naive package!
encoding	Package encoding defines interfaces shared by other packages that convert data to and from byte-level and textual representations.
ascii85	Package ascii85 implements the ascii85 data encoding as used in the btoa tool and Adobe's PostScript and PDF document formats.
asn1	Package asn1 implements parsing of DER-encoded ASN.1 data structures, as defined in ITU-T Rec X.690.
base32	Package base32 implements base32 encoding as specified by RFC 4648.
base64	Package base64 implements base64 encoding as specified by RFC 4648.
binary	Package binary implements simple translation between numbers and byte sequences and encoding and decoding of varints.
csv	Package csv reads and writes comma-separated values (CSV) files.
gob	Package gob manages streams of gobs - binary values exchanged between an Encoder (transmitter) and a Decoder (receiver).
hex	Package hex implements hexadecimal encoding and decoding.
json	Package json implements encoding and decoding of JSON as defined in RFC 7159.
pem	Package pem implements the PEM data encoding, which originated in Privacy Enhanced Mail.
xml	Package xml implements a simple XML 1.0 parser that understands XML name spaces.
errors	Package errors implements functions to manipulate errors.
expvar	Package expvar provides a standardized interface to public variables, such as operation counters in servers.
flag	Package flag implements command-line flag parsing.
fmt	Package fmt implements formatted I/O with functions analogous to C's printf and scanf.
github.com	
Arafatk	
glot	
mattn	
go-sqlite3	Package sqlite3 provides interface to SQLite3 databases.
tool	
go	
ast	Package ast declares the types used to represent syntax trees for Go packages.



documentMe - The Go Prograr ×
Mihalis
localhost:8080/pkg/documentMe/
The Go Programming Language
Documents
Packages
The Project
Help
Blog
Search

Package documentMe

```
import "documentMe"
```

Overview
Index
Examples

Overview ▾

This package is for showcasing the documentation capabilities of Go It is a naive package!

Index ▾

Constants
func F1(n int) int
func S1(s string) int

Examples

F1
S1

Package files

documentMe.go

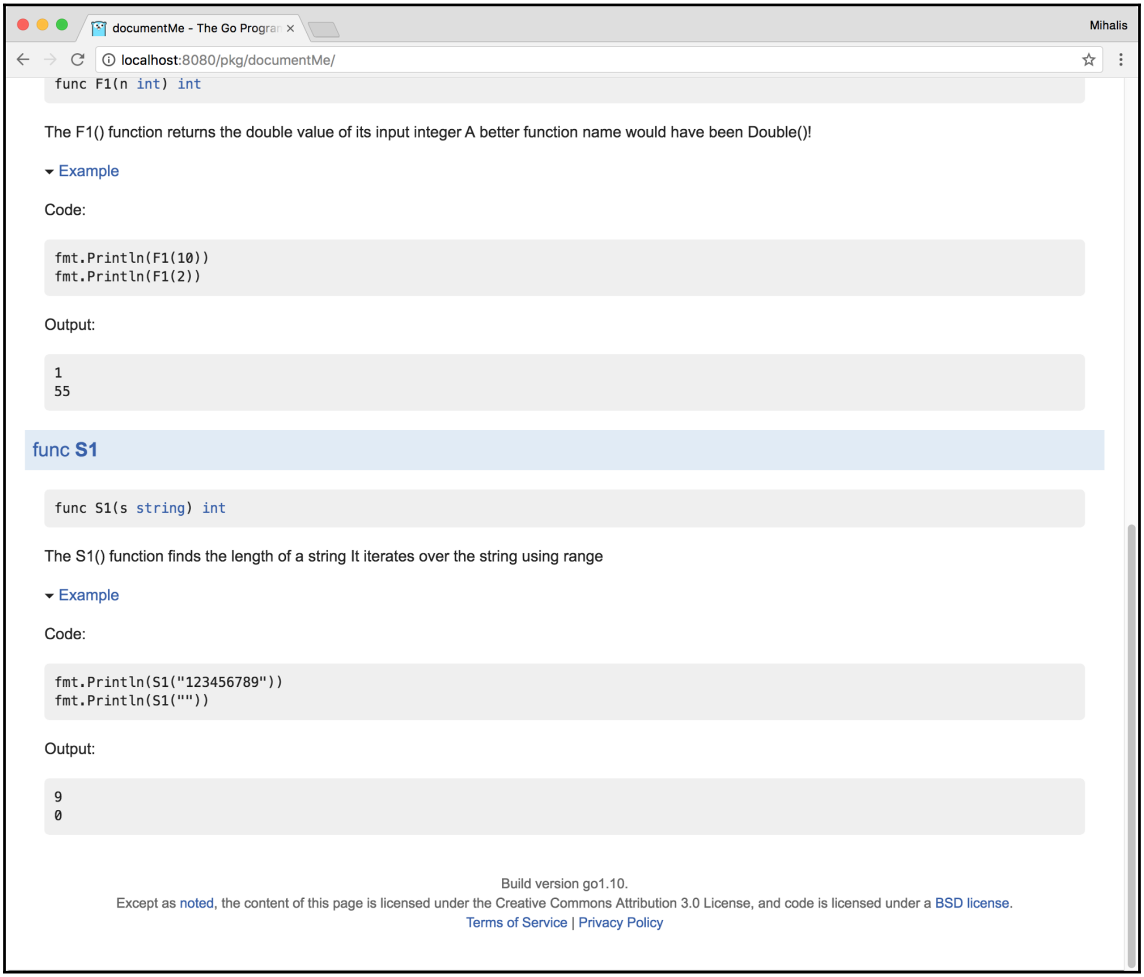
Constants

Pie is a global variable This is a silly comment!

```
const Pie = 3.1415912
```

func F1

```
func F1(n int) int
```



```
func F1(n int) int
```

The F1() function returns the double value of its input integer A better function name would have been Double()!

▾ Example

Code:

```
fmt.Println(F1(10))
fmt.Println(F1(2))
```

Output:

```
1
55
```

func **S1**

```
func S1(s string) int
```

The S1() function finds the length of a string It iterates over the string using range

▾ Example

Code:

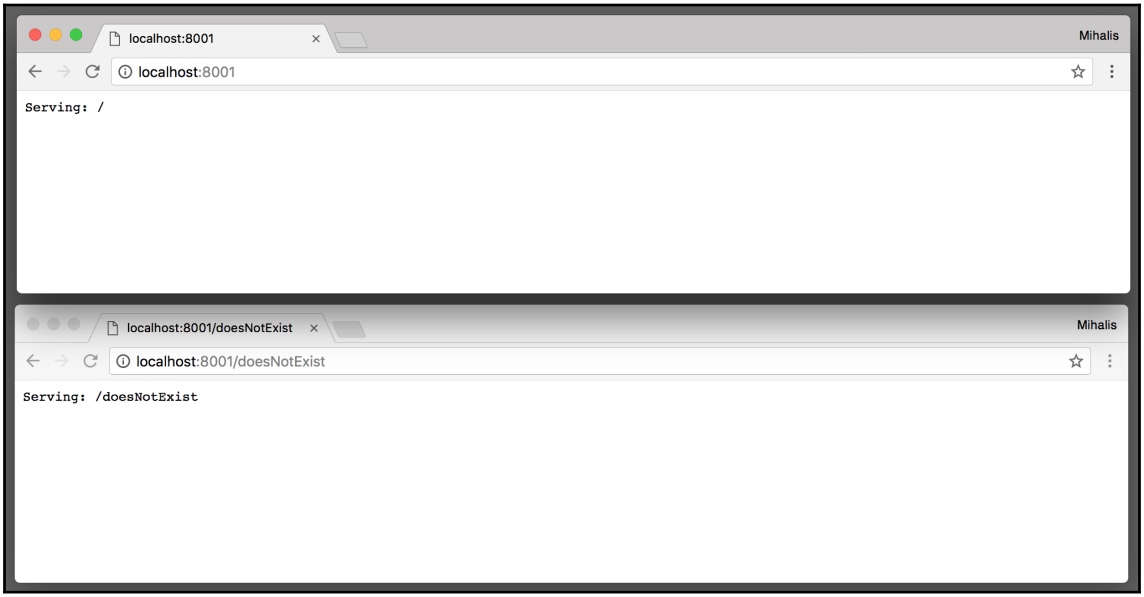
```
fmt.Println(S1("123456789"))
fmt.Println(S1(""))
```

Output:

```
9
0
```

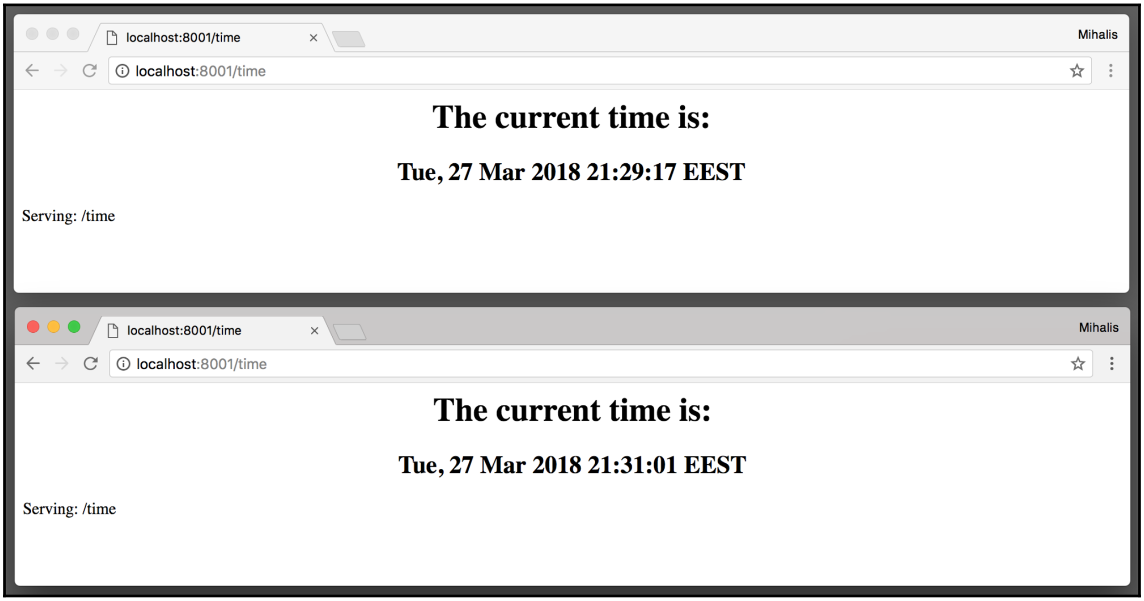
Build version go1.10.
Except as noted, the content of this page is licensed under the Creative Commons Attribution 3.0 License, and code is licensed under a BSD license.
Terms of Service | Privacy Policy

Chapter 12: The Foundations of Network Programming in Go



```
Serving: /
```

```
Serving: /doesNotExist
```



The current time is:

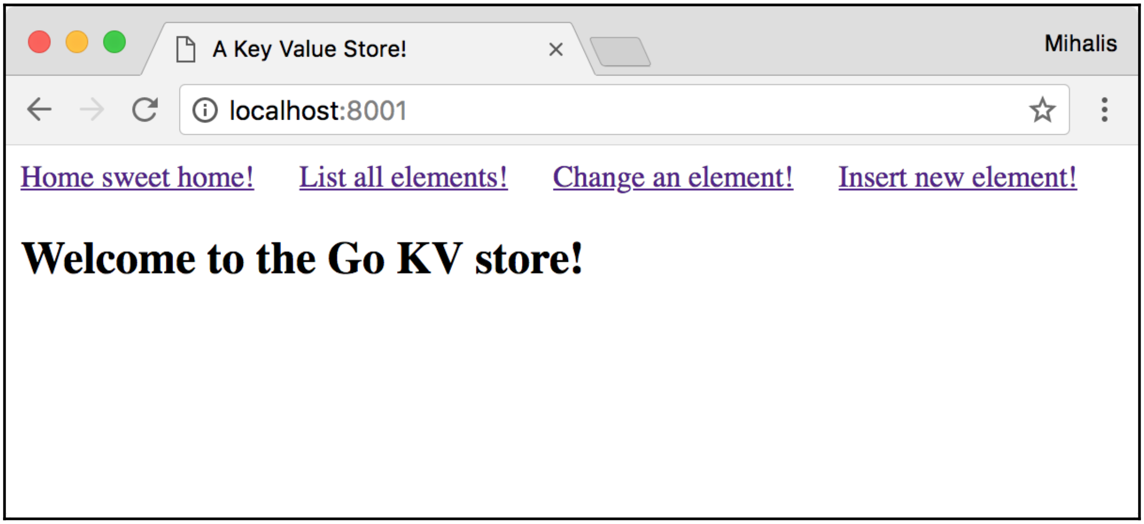
Tue, 27 Mar 2018 21:29:17 EEST

Serving: /time

The current time is:

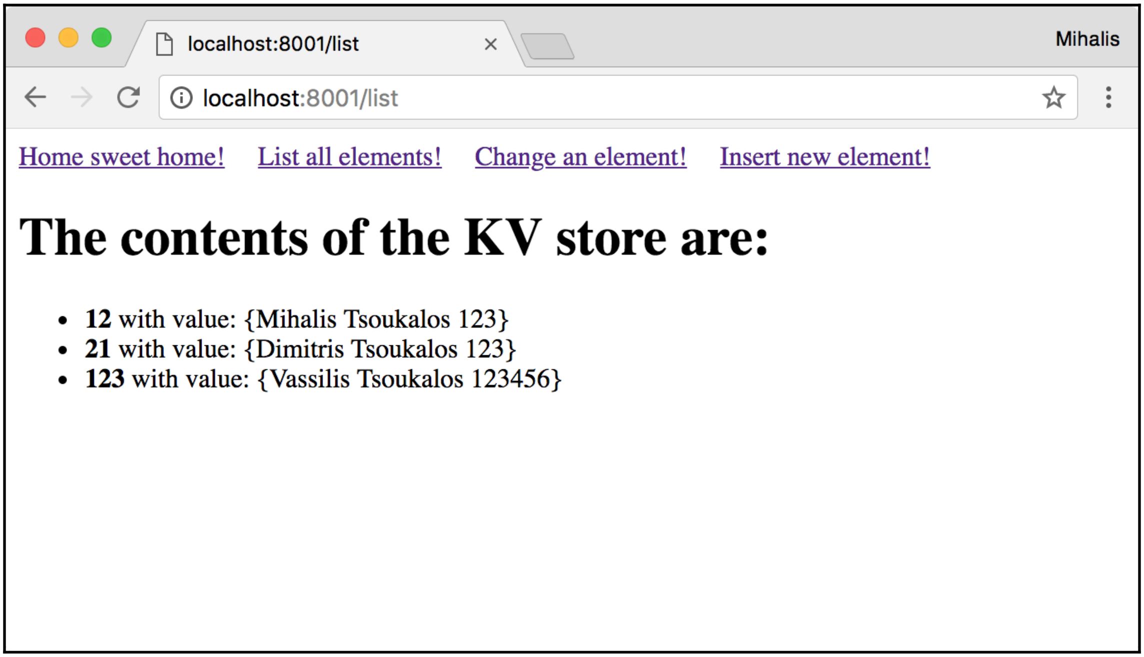
Tue, 27 Mar 2018 21:31:01 EEST

Serving: /time



Home sweet home! List all elements! Change an element! Insert new element!

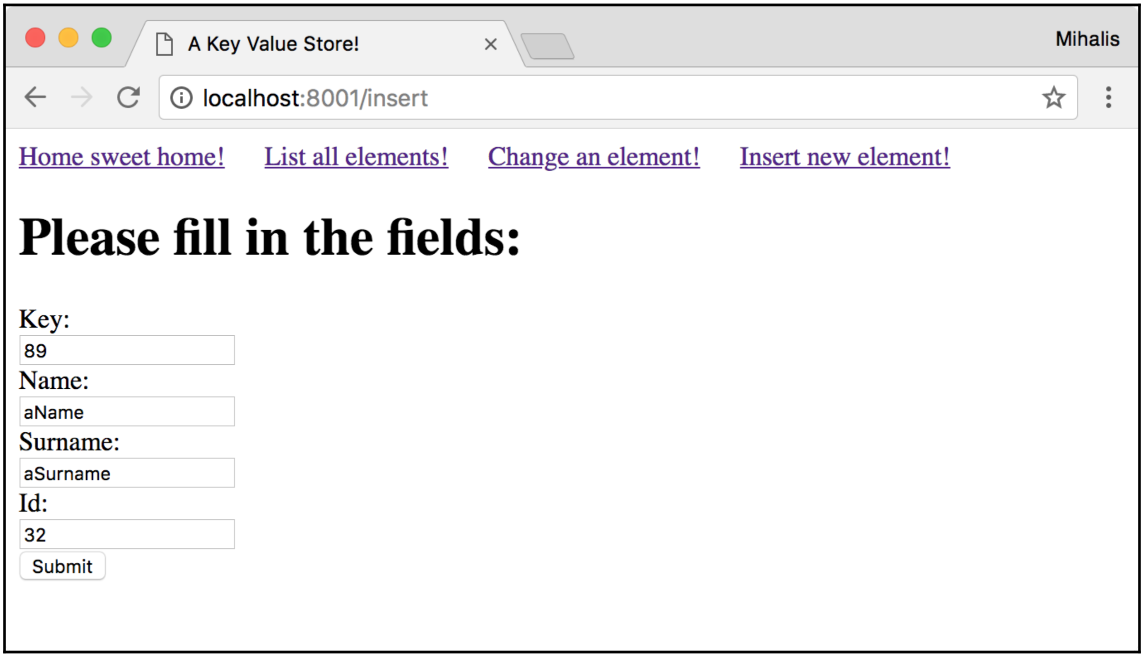
Welcome to the Go KV store!



Home sweet home! List all elements! Change an element! Insert new element!

The contents of the KV store are:

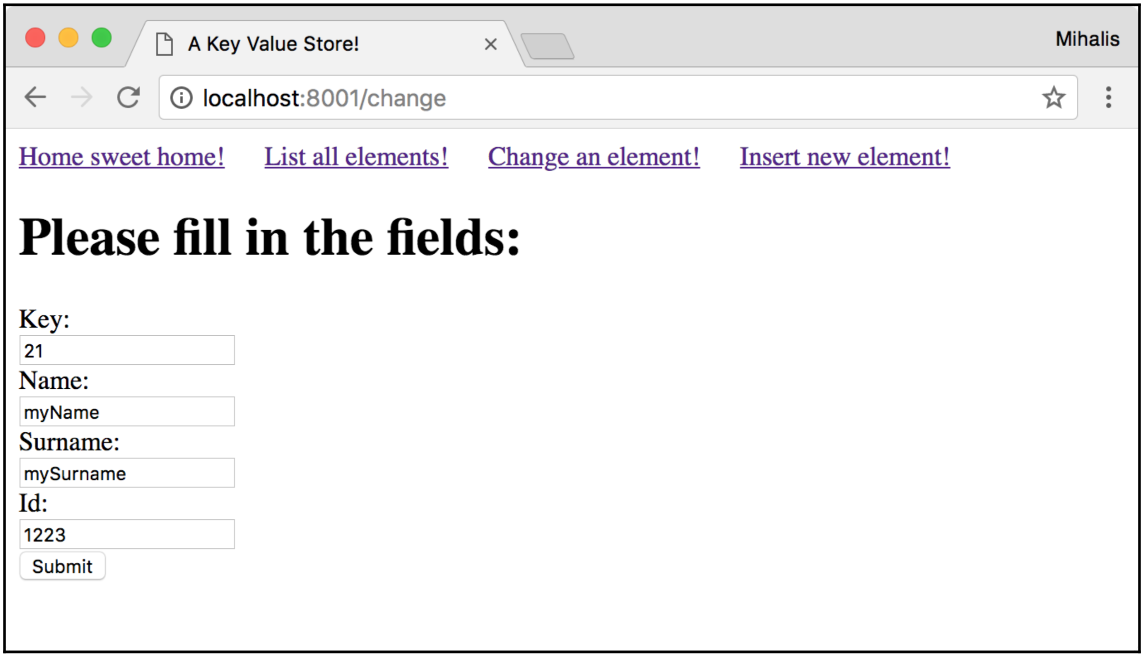
- **12** with value: {Mihalis Tsoukalos 123}
- **21** with value: {Dimitris Tsoukalos 123}
- **123** with value: {Vassilis Tsoukalos 123456}



Home sweet home! List all elements! Change an element! Insert new element!

Please fill in the fields:

Key:
89
Name:
aName
Surname:
aSurname
Id:
32
Submit



Home sweet home! List all elements! Change an element! Insert new element!

Please fill in the fields:

Key:
21
Name:
myName
Surname:
mySurname
Id:
1223
Submit