Go
Everything You Wanted to Know About Go
and
Didn't Know to Ask

Raymond E. Marcil

<marcilr@gmail.com>

Revision 52 (February 4, 2013)

Abstract

Working at GCI Network Services, OSS I needed a location for Go documentation. Hence this document.

Contents

List of Figures

List of Tables

List of Definitions and Abbreviations

- **MOA** - Municipality of Anchorage

Introduction

Introduction to the LaTeX Template with Examples.

Packages

Types

Type such as arrays, structs, slices, and maps.[1]

---

[1] "More types: structs, slices, and maps. Learn how to define types based on existing ones: this lesson covers structs, arrays, slices, and maps."
https://tour.golang.org/list

Functions

Variables

GOPATH

"The `GOPATH` environment variable is used to specify directories outside of `$GOROOT` that contain the source for Go projects and their binaries."[2]

Flow Control

Flow control of code using conditionals like `for`, `if`, `else`, `switch`, and defer.[3]

Iteration

While, for, do until, and such.
[FIXME: Need data here]

---

[2]`GOPATH`, `golang/go`, https://github.com/golang/go/wiki/GOPATH

[3]"Flow control statements: for, if, else, switch and defer. Learn how to control the flow of your code with conditionals, loops, switches and defers."
https://tour.golang.org/list

DRAFT -- DRAFT -- DRAFT -- DRAFT -- DRAFT --

Database Access

"Two fast and reliable drivers are available for MySQL :

- MyMySQL
  http://github.com/ziutek/mymysql

- Go-MySQL-Driver
  https://github.com/Go-SQL-Driver/MySQL

I've used both of them in production, programs are running for months with connection numbers in the millions without failure."[4]

MyMySQL

[FIXME: Need data here]

Go-MySQL-Driver

"A MySQL-Driver for Go's database/sql package"[5]

Installation

Simple install the package to your `$GOPATH` with the go tool from shell:[6]

```
$ go get github.com/go-sql-driver/mysql
```

Make sure Git is installed on your machine and in your system's PATH."

Usage

Go MySQL Driver is an implementation of Go's `database/sql/driver` interface. You only need to import the driver and can use the full database/sql API then.

Use `mysql` as driverName and a valid DSN as dataSourcename:

```
import "database/sql"
import _ "github.com/go-sql-driver/mysql"

db, err := sql.Open("mysql", "user:password@/dbname")
```

---

[4]What's the recommended way to connect to MySQL from Go?
http://stackoverflow.com/questions/11353679/whats-the-recommended-way-to-connect-to-mysql-from-go
[5]Go-MySQL-Driver, https://github.com/Go-SQL-Driver/MySQL/
[6]Ibid.

DSN (Data Source Name)

"The Data Source Name has a common format,... (optional parts marked by squared brackets):[7]

$$[username[: password]@][protocol[(address)]]/dbname[?param1 = value1\&...\&paramN = valueN]$$

A DSN in its fullest form:
$$username : password@protocol(address)/dbname?param = value$$

Except for the databasename, all values are optional. So the minimal DSN is:
$$/dbname$$

If you do not want to preselect a database, leave dbname empty:
$$/$$

This has the same effect as an empty DSN string:

Alternatively, `Config.FormatDSN` can be used to create a DSN string by filling a struct."

---

[7]DSN (Data Source Name),
https://github.com/Go-SQL-Driver/MySQL/#dsn-data-source-name

DRAFT -- DRAFT -- DRAFT -- DRAFT -- DRAFT --

Comments

"Comments serve as program documentation. There are two forms:

1. Line comments start with the character sequence `//` and stop at the end of the line.

2. General comments start with the character sequence `/*` and stop with the first subsequent character sequence `*/`.

A comment cannot start inside a rune or string literal, or inside a comment. A general comment containing no newlines acts like a space. Any other comment acts like a newline."[8]

Examples

---

[8]Comments, Lexical elements, The Go Programming Language Specification, Version of January 5, 2016, https://golang.org/ref/spec#Comments

Appendix

Links

About the go command
The Go Programming Language
https://golang.org/doc/articles/go_command.html

A Tour of Go
The tour covers the most important features of the language, mainly: Basics (Packages, variables, functions, flow control statements: for, if, else, switch, defer, more types: structs, slices, and maps), Methods and interfaces, and Concurrency.
https://tour.golang.org/welcome/1

Go example projects
This repository contains a collection of Go programs and libraries that demonstrate the language, standard libraries, and tools.
https://github.com/golang/example

Go-MySQL-Driver
Go MySQL Driver is a lightweight and fast MySQL driver.
https://github.com/Go-SQL-Driver/MySQL

How to Write Go Code
https://golang.org/doc/code.html

List of integrated circuit packaging types
From Wikipedia, the free encyclopedia
https://en.wikipedia.org/wiki/List_of_integrated_circuit_packaging_types

MyMySQL
MySQL driver
http://github.com/ziutek/mymysql

Package Testing
`"import testing"`
The Go Programming Language
https://golang.org/pkg/testing/

Package types
`import "go/types"`
Package types declares the data types and implements the algorithms for type-checking of Go packages. Use `Config.Check` to invoke the type checker for a package. Alternatively, create a new type checked with `NewChecker` and invoke it incrementally by calling `Checker.Files`.

DRAFT -- DRAFT -- DRAFT -- DRAFT -- DRAFT --

https://golang.org/pkg/go/types/

The Go Programming Language
https://golang.org/pkg/go/types/

The Go Programming Language Specification
Version of January 5, 2016
The Go Programming Language
https://golang.org/ref/spec#Selectors

zieckey/goini
A simple, flexible, efficient and powerful INI parser for Golang.
import github.com/zieckey/goini
https://github.com/zieckey/goini