
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

Contents	2
List of Figures	3
List of Tables	3
List of Definitions and Abbreviations	4
Introduction	5
Packages	6
Types	7
Functions	8
Variables	9
GOPATH	9
Flow Control	9
Iteration	9
Database Access	10
MyMySQL	10
Go-MySQL-Driver	11
Installation	11
Usage	11
DSN (Data Source Name)	12
Comments	13
Error Handling	14
Examples	14
Appendix	15
Links	15

List of Figures

List of Tables

List of Definitions and Abbreviations

- **MOA** - Municipality of Anchorage

Introduction

Introduction to the L^AT_EX 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>

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]

⁴What’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>

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`.”⁷

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

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

⁵Go-MySQL-Driver, <https://github.com/Go-SQL-Driver/MySQL/>

⁶Ibid.

⁷Usage, Go-MySQL-Driver,
<https://github.com/Go-SQL-Driver/MySQL/#usage>

DSN (Data Source Name)

“The Data Source Name has a common format,... (optional parts marked by squared brackets):⁸

[username[: password]@][protocol[(address)]]/dbname[?param1 = value1&...¶mN = 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.”

⁸DSN (Data Source Name),
<https://github.com/Go-SQL-Driver/MySQL/#dsn-data-source-name>

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.”⁹

⁹Comments, Lexical elements, The Go Programming Language Specification, Version of January 5, 2016, <https://golang.org/ref/spec#Comments>

Error Handling

[FIXME: Need data here]

Examples

[FIXME: Need data here]

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`.

<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>

ziecekey/goini

A simple, flexible, efficient and powerful INI parser for Golang.

import github.com/ziecekey/goini

<https://github.com/ziecekey/goini>