

ORM-

[ORM](#)SQLDoQueryDoExec DoFilter

ORMSQL

SQLmonitorSQLDriverORMMySQLdriversmysqlDriver

Content Menu

-
-
-

```

package driver

import (
    "context"

    "github.com/gogf/gf/contrib/drivers/mysql/v2"
    "github.com/gogf/gf/v2/database/gdb"
    "github.com/gogf/gf/v2/os/gtime"
)

// MyDriver is a custom database driver, which is used for testing only.
// For simplifying the unit testing case purpose, MyDriver struct inherits
the mysql driver
// gdb.Driver and overwrites its functions DoQuery and DoExec.
// So if there's any sql execution, it goes through MyDriver.DoQuery
/MyDriver.DoExec firstly
// and then gdb.Driver.DoQuery/gdb.Driver.DoExec.
// You can call it sql "HOOK" or "HiJack" as your will.
type MyDriver struct {
    *mysql.Driver
}

var (
    // customDriverName is my driver name, which is used for
    registering.
    customDriverName = "MyDriver"
)

func init() {
    // It here registers my custom driver in package initialization
    function "init".
    // You can later use this type in the database configuration.
    if err := gdb.Register(customDriverName, &MyDriver{}); err != nil {
        panic(err)
    }
}

// New creates and returns a database object for mysql.
// It implements the interface of gdb.Driver for extra database driver
installation.
func (d *MyDriver) New(core *gdb.Core, node *gdb.ConfigNode) (gdb.DB,
error) {
    return &MyDriver{
        &mysql.Driver{
            Core: core,
        },
        nil
    }
}

// DoCommit commits current sql and arguments to underlying sql driver.
func (d *MyDriver) DoCommit(ctx context.Context, in gdb.DoCommitInput)
(out gdb.DoCommitOutput, err error) {
    tsMilliStart := gtime.TimestampMilli()
    out, err = d.Core.DoCommit(ctx, in)
    tsMilliFinished := gtime.TimestampMilli()
    _, _ = in.Link.ExecContext(ctx,
        "INSERT INTO `monitor`(`sql`,`cost`,`time`,`error`) VALUES
(?,?,?,?)",
        gdb.FormatSqlWithArgs(in.Sql, in.Args),
        tsMilliFinished-tsMilliStart,
        gtime.Now(),
        err,
    )
    return
}

```

```

initgdb.Register("MyDriver", &MyDriver{})gdb.Register("mysql", &MyDriver{})mys
ql

```



mysqlDriverMysql

MyDrivergdbtype

```
database:
  default:
    - link: "MyDriver:root:12345678@tcp(127.0.0.1:3306)/user"
```

Linkg.DB