初始化 MySQL 数据库并建立连接

本节核心内容

- Go ORM 数量众多，本小册介绍一个笔者认为比较好的 ORM 包，并给出原因
- 介绍如何初始化数据库
- 介绍如何连接数据库

> 本小节源码下载路径：[demo04 (https://github.com/lexkong/apiserver_demos/tree/mas](https://github.com/lexkong/apiserver_demos/tree/mas)
>
> 可先下载源码到本地，结合源码理解后续内容，边学边练。
>
> 本小节的代码是基于 [demo03 (https://github.com/lexkong/apiserver_demos/tree/mas](https://github.com/lexkong/apiserver_demos/tree/mas) 来开发的。
>
> apiserver 用的 ORM 是 GitHub 上 star 数最多的 [gorm (https://github.com/jinzhu/gorm)](https://github.com/jinzhu/gorm)，相较于其他 ORM，它用起来更方便，更稳定，社区也更活跃。

初始化数据库

在 `model/init.go` 中添加数据初始化代码

因为一个 API 服务器可能需要同时访问多个数据库，为了对多个数据库进行初始化和连接管理，这里定义了一个叫 Database 的 struct：

```
type Database struct {
    Self   *gorm.DB
    Docker *gorm.DB
}
```

Database 结构体有个 Init() 方法用来初始化连接：

```
func (db *Database) Init() {
    DB = &Database {
        Self:   GetSelfDB(),
        Docker: GetDockerDB(),
    }
}
```

Init() 函数会调用 GetSelfDB() 和 GetDockerDB() 方法来同时创建两个 Database 的数据库对象。这两个 Get 方法最终都会调用 func openDB(username, password, addr, name string) *gorm.DB 方法来建立数据库连接，不同数据库实例传入不同的 username、password、addr 和名字信息，从而建立不同的数据库连接。openDB 函数为：

```
func openDB(username, password, addr, name
string) *gorm.DB {
    config := fmt.Sprintf("%s:%s@tcp(%s)/%s?
charset=utf8&parseTime=%t&loc=%s",
        username,
        password,
        addr,
        name,
        true,
        //"Asia/Shanghai"),
        "Local")

    db, err := gorm.Open("mysql", config)
    if err != nil {
        log.Errorf(err, "Database connection
failed. Database name: %s", name)
    }

    // set for db connection
    setupDB(db)

    return db
}
```

可以看到，openDB() 最终调用 gorm.Open() 来建立一个数据库连接。

完整的 model/init.go 源码文件请参考 [demo04/model/init.go (https://github.com/lexkong/apiserver_demos/tree/master/c](https://github.com/lexkong/apiserver_demos/tree/master/c)

主函数中增加数据库初始化入口

```
package main

import (
    ...
    "apiserver/model"

    ...
)

...

func main() {
    ...

    // init db
    model.DB.Init()
    defer model.DB.Close()

    ...
}
```

通过 model.DB.Init() 来建立数据库连接，通过 defer model.DB.Close() 来关闭数据库连接。

编译并运行

1. 下载 apiserver_demos 源码包（如前面已经下载过，请忽略此步骤）

```
$ git clone
https://github.com/lexkong/apiserver_demos
```

2. 将apiserver_demos/demo04复制

为$GOPATH/src/apiserver

```
$ cp -a apiserver_demos/demo04/
$GOPATH/src/apiserver
```

3. 在 apiserver 目录下编译源码

```
$ cd $GOPATH/src/apiserver
$ gofmt -w .
$ go tool vet .
$ go build -v .
```

小结

本小结介绍了如何用 gorm 建立数据库连接，为之后的业务逻辑处理作准备。至于具体怎么使用 gorm 来进行增删改查等操作，请参考 GORM 指南 (http://gorm.io/zh_CN/docs/index.html)。

> 本小节只是介绍了如何初始化数据库，至于怎么对数据库做 CURD 操作，请参考第 12 节：用户业务逻辑处理。