“Stock Pro Server” 只是一个开发代号， 不是产品名称。

[Server 开发平台与架构](#h.omvzjj9sxdwh)

[开发环境建议](#h.jq1b61bdzes4)

[配置文件](#h.dilw15r1b58h)

[部署流程](#h.bl0286q8aex4)

[设计与实现](#h.omv8yiotsmdc)

[数据表](#h.j5wh40d0dc8m)

[数据结构](#h.jananc3qw34v)

[实现笔记与扩展建议](#h.8lngw1cja7vn)

[源码结构](#h.507zlqhrh4c3)

[接口实现说明](#h.9v3re2gre1b7)

[csv 数据录入程序及数据同步方案建议](#h.swicyhboswx)

[录入程序](#h.xuil06jvlf1a)

[同步方案](#h.vfdlufqesx6w)

# Server 开发平台与架构

开发语言： Go 语言

框架： Revel

数据库： MariaDB / MySQL

## 开发环境建议

1 ide

eclipse + Goclipse 插件

或 vim + go-vim 插件

2 依赖安装

1 revel <http://revel.github.io/tutorial/gettingstarted.html>

2 go get github.com/go-sql-driver/mysql

## 配置文件

conf/app.conf 文件中需要配置数据库连接字符串 db.datasource

## 部署流程

revel run nlfinance/stockpro

或配置 nginx 的 proxy\_pass 字段部署到现有环境

# 设计与实现

## 数据表

/\*

1 SP for Stock Pro

2 id -gen> salt -hash>passwd

3 email considering the longest possible email address

4 status

0: inactive

1: normal

2: suspended for violating rules.

5 note for op: reason of the status

\*/

create table T\_SP\_USER(

id integer not null auto\_increment primary key,

username varchar(16) not null,

email varchar(254) not null,

passwd varchar(500) not null,

userstatus integer not null default 1,

note varchar(511) null,

ctime timestamp not null default current\_timestamp,

mtime timestamp not null default current\_timestamp on update current\_timestamp

);

create unique index IDX\_U\_SP\_USER\_NAME on T\_SP\_USER(username);

create unique index IDX\_U\_SP\_USER\_EMAIL on T\_SP\_USER(email);

create table T\_SP\_STOCK\_INFO(

id integer not null auto\_increment primary key,

stockname varchar(200),

listid varchar(10),

mktcode integer,

abbr varchar(20),

corpname varchar(200),

ctime timestamp not null default current\_timestamp,

mtime timestamp not null default current\_timestamp on update current\_timestamp

);

create unique index IDX\_U\_SP\_STOCK\_INFO on T\_SP\_STOCK\_INFO(mktcode, listid);

create index IDX\_SP\_STOCK\_INFO on T\_SP\_STOCK\_INFO(stockname, abbr, corpname);

create table T\_SP\_CLOSE\_RETURN(

id integer not null auto\_increment primary key,

StockCode varchar(10) not null,

BeginDate date not null,

EndDate date not null,

Window integer ,

MatchedStockCode varchar(10),

M\_BeginDate date,

M\_EndDate date,

Dist double,

E\_PreA double,

E\_PreB double,

E\_PreC double,

E\_PreA\_F double,

E\_PreB\_F double,

E\_PreC\_F double,

PreA double,

PreB double,

PreC double,

PreA\_F double,

PreB\_F double,

PreC\_F double,

ctime timestamp default current\_timestamp,

mtime timestamp default current\_timestamp on update current\_timestamp

);

create unique index IDX\_U\_SP\_CLOSE\_RETURN on T\_SP\_CLOSE\_RETURN(

StockCode, EndDate, Window, MatchedStockCode, M\_EndDate

);

create index IDX\_SP\_CLOSE\_RETURN\_StockCode on T\_SP\_CLOSE\_RETURN(

StockCode, EndDate, Window, Dist

);

## 数据结构

type StockShape struct {

// Period PERIOD

StockId string

StartTime time.Time

EndTime time.Time

Window int

}

type RespBase struct {

Status int

Msg string

}

// get /stockpro/{stockid}/recentKGraph

type RespRecentKGraph struct {

RespBase

GraphUri string

}

// post /stockpro/similar\_shape

// stock id

// window

// start time

// end time

type RespShapeMatches struct {

RespBase

MatchedShapes []StockShapeMatch

}

type RespSimilarShapes struct {

RespBase

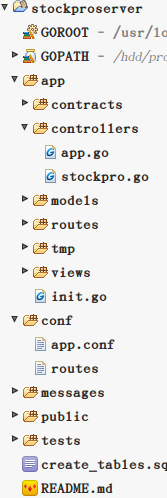
Shapes []StockShape

}

## 实现笔记与扩展建议

原代码请见： nlfinance.tar.gz

### 源码结构



其中

app/contracts 是数据结构的定义

app/controllers 是 endpoints 的定义

app/models 是数据相关操作及抽象

app/routes 及 app/tmp 是 revel 生成代码

app/views 是 html template

conf/app.conf 中需要配置数据库相关参数

routes 中可配置 url 与 endpoints 的映射关系

### 接口实现说明

1 GET /stockpro/{stockid}/recentKGraph 通过 stock id 拼接新浪 k 线图地址

2 POST /stockpro/similar\_shape 通过查表， 找到形态最接近的 n 个记录

3 GET /stockpro/SearchStocks/?keyword=xxx&limit=yyy 通过查表， 找到与 keyword 相似的股票代码

详细笔记，请见《实现笔记 》

# csv 数据录入程序及数据同步方案建议

源码： csv2db.tar.gz

### 录入程序

解压后只有一个文件 importer/main.go

以下参数需要手工修改代码实现：

数据库连接参数 line 19

扫描根目录 line 36

运行程序后， 会从根目录开始递归扫描文件， 发现 csv 文件后即尝试解析并导入到数据库中．

### 同步方案

1 服务器间文件同步（比如与数据源之间的同步）可以使用 btsync 点对点同步软件实现

<https://www.btsync.com/en/>

按文档安装部署即可， 可以在多个服务器间同步。

2 本地数据同步有以下几个选择：

a. 修改录入程序， 记录已导入的数据文件， 按指定频率通过 crontab 或计划任务运行程序

b. 通过 shell 脚本， 每次执行完程序后，把旧文件转移到其他目录， 仍是按指定频率通过 crontab 或计划任务运行程序

c. 改写录入程序为常驻内存， 定时扫描目录文件， 对比 md5 hash 数值， 发现有不同或有新文件时就导入对应文件。