

Tecnologías para desarrollos en internet

Manual CRUD: Beego

Kihui-DEV

Fecha: 18/09/16
Facultad de Ciencias UNAM

Índice

1. Introducción	2
2. Instalación de Go 1.7	2
2.1. Descarga	2
2.2. Configuración	2
2.3. Prueba	3
3. Instalación de Beego	4
4. Creación de proyecto	4
4.1. Estructura del proyecto - MVC	5
4.2. Prueba del servidor local	5
5. Elaboración del CRUD bajo MVC	7
5.1. Preparación	7
5.2. Creación	8
5.3. Lectura	8
5.4. Actualización	8
5.5. Eliminación	8
5.6. Resultados	8
6. Referencias	8

1. Introducción

2. Instalación de Go 1.7

Paquetes de instalación para *Apple OS X*, *Microsoft Windows* y *Linux* son provistos en la página oficial de descargas de Go. También viene incluido entre las opciones el código fuente del compilador del lenguaje junto con instrucciones para llegar a una instalación tan completa como las demás.

A continuación, presentamos la instalación para *Linux* y *OS X*.¹

2.1. Descarga

Linux

Para obtener el paquete de Golang²
Ejecutar:

```
$ wget https://golang.org/doc/install?download=go1.7.1.linux-amd64.tar.gz
```

O bien descargarlo directamente desde este enlace:

Go 1.7 para Linux y OS X.

Mac OS X

Alternativa a las siguientes instrucciones, existe la opción de descargar el fichero *.pkg* instalable de *Go* que automatiza la configuración para este sistema operativo.³

Para obtener una configuración inicial personalizada *Mac OS X* utilice el paquete descargable para *Linux* disponible en la sección anterior.

El resto de la configuración se sigue de la misma manera para ambas plataformas.

2.2. Configuración

Primero extraemos los archivos del paquete comprimido sobre algún directorio.
Para descomprimir el paquete sobre */usr/local* como es usual:

```
# tar -C /usr/local -xzf go1.7.1.linux-amd64.tar.gz
```

¹Si se desea revisar la configuración para *Windows*, hacer uso del siguiente enlace: MSI Installer

²Actualmente la versión 1.7.1

³Instalador para Mac OS X

Al finalizar la extracción, procedemos a establecer una ruta a los binarios de las herramientas de *Go*, y así tenerlas disponibles en cada sesión.

Agregar esta línea en el script de inicio (típicamente sobre */user/local/profile* si se desea hacer una instalación general en el sistema operativo o en particular para el usuario en curso, usar en cambio *~/.profile*):

```
export PATH=$PATH:/usr/local/go/bin
```

2.3. Prueba

Crear un directorio que haga de workspace para la prueba.
Por ejemplo:

```
$ mkdir ~/go
```

Asignar la variable **GOPATH**⁴ para que apunte a tal dirección:

```
$ export GOPATH=$HOME/go
```

Bien podemos hacer persistente este cambio agregando la misma línea al script de inicio que editamos en la sección anterior (ir a Descarga 2.1 y configuración 2.2).

A continuación, creamos dentro de ese directorio *src/hola*. Y dentro de *hola/* un fichero nuevo de nombre *hola.go*:

```
package main

import "fmt"

func main() {
    fmt.Printf("hola, mundo\n")
}
```

Luego, desde cualquier ubicación podemos ejecutar esto:

```
$ go install hola
```

Esto producirá un ejecutable *hola* dentro de el directorio *go/bin/*, que podemos ejecutar utilizando lo siguiente:

```
$ $GOPATH/bin/hola
```

o directamente sobre el directorio donde se encuentre el ejecutable:

```
$ ./hola
```

Si produce la salida “hola, mundo”, quiere decir que nuestra instalación fue exitosa.

⁴La misma línea puede agregarse al script de inicio *profile* manejado en la sección de configuración para evitar ejecutarla y mantener los proyectos y aplicaciones de *Go* ubicados en un sólo directorio.

3. Instalación de Beego

Para instalar la última versión de Beego⁵ utilizamos el siguiente comando:

```
$ go get github.com/astaxie/beego
```

Para compilar y correr nuestros proyectos necesitaremos instalar Bee⁶ también :

```
$ go get github.com/beego/bee
$ go install github.com/beego/bee
```

Para poder utilizar *bee* sin necesidad de ir a la carpeta de binarios de *Go*, podemos crear un enlace simbólico que apunte precisamente al ejecutable.

```
# ln -s $GOPATH/bin/bee /usr/bin/bee
```

4. Creación de proyecto

Para crear un proyecto en Beego, necesitamos ir al directorio de nuestro \$GOPATH, donde escribimos el siguiente comando:

```
$ bee new beego-crud
```

Podremos ver que se han creado las siguientes carpetas y archivos necesarios para nuestra aplicación:

```
beego-crud/
|-- conf/
|   |-- app.conf
|-- controllers/
|   |-- default.go
|-- main.go
|-- models/
|-- routers/
|   |-- router.go
|-- static/
|   |-- css/
|   |-- img/
|   |-- js/
|-- tests/
|   |-- default_test.go
|-- views/
|   |-- index.tpl
```

⁵A la fecha de elaboración de este manual: 1.7.1

⁶A la fecha de elaboración de este manual: 1.5.2

4.1. Estructura del proyecto - MVC

Modelo

- 1) *conf/*
- 2) *models/*

Vista

- 3) *controllers/*
- 4) *routers/*

Controlador

- 5) *views/*

Extra

- 6) *static/*
- 7) *tests/*

4.2. Prueba del servidor local

Finalmente para correr el nuevo proyecto que hemos creado, hacemos lo siguiente:

```
$ cd $GOPATH/src/beego-crud  
$ bee run
```

Ingresamos *localhost:8080* como dirección en el navegador para encontrarnos con la siguiente página de inicio:



Welcome to Beego

Beego is a simple & powerful Go web framework which is inspired by tornado and sinatra.

Official website: beego.me / Contact me: astaxie@gmail.com

5. Elaboración del CRUD bajo MVC

En esta sección revisaremos la implementación de un CRUD (*Create-Read-Update-Delete*) sobre la aplicación que creamos en la sección anterior (s. 4). Comenzaremos explicando los documentos que tenemos que modificar para tener una configuración exitosa y la forma de escribir el código de acuerdo al MVC (*Modelo Vista Controlador*), siguiendo la estructura que nos proporciona *Beego* para éste.

5.1. Preparación

Para implementar nuestro primer CRUD en Beego es necesario tener un sistema de datos persistente sobre el que realicemos nuestras operaciones. La mejor solución que en casi todos los casos se señala para lograr esto en el desarrollo web por supuesto no son los archivos, sino una base de datos, por lo general relacional.

Todo proyecto de Beego, como ya vimos, está organizado según el MVC. Es precisamente sobre el archivo de **models.go** que escribiremos nuestras relaciones expresadas con atributos asignados de acuerdo a los tipos de dato que maneja Go para después verlos reflejados en tablas en una base de datos ya explorable. Bien podría ser en sentido inverso, desde un esquema *SQL* generar los modelos para nuestro proyecto de Beego, pero en nuestro caso decidimos implementarlo en esta dirección porque nuestra intención es enfocarnos en Beego, no en SQL.

A continuación presentamos la configuración de la base de datos y un mapeo de una sola relación con la que trabajaremos todas las operaciones.

Base de datos

Beego-ORM es la herramienta de Mapeo Objeto-Relacional de *Beego* escrita en *Go*. Según la página oficial está inspirada en *Django ORM* y *SQLAlchemy*. Se advierte que al estar en desarrollo, no se garantiza un 100 % de compatibilidad y es posible encontrar algunos “bugs”, que el equipo de desarrollo de Beego promete solucionar apenas se reporten.

Lo primero que hay que considerar es la conexión con la base de datos que necesitamos para nuestro modelo. Beego provee soporte para tres sistemas manejadores de bases de datos con sus respectivos “drivers”:

- MySQL
- Sqlite
- Postgres

Por su sencilla configuración y mayoría de ejemplos de implementación junto con Beego, escogimos MySQL por lo que los siguientes puntos se ejemplificarán con el supuesto de que trabajamos

con este SMBD⁷ y que ya se cuenta con una instalación funcional⁸.

Lo primero que haremos será crear nuestra base de datos:

```
$ mysql -u root -p
MariaDB [(none)]> create database beego;
MariaDB [(none)]> exit
```

Configuración

Para realizar la configuración de conf/app.conf ??

5.2. Creación

5.3. Lectura

5.4. Actualización

5.5. Eliminación

5.6. Resultados

6. Referencias

⁷Para revisar más ampliamente las opciones de bases de datos recomendamos visitar este apartado de la documentación: ORM Usage

⁸En el manual de MySQL v5.7 viene una amplia guía de instalación. Recomendamos los siguientes apartados para distintos sistemas operativos: Windows | OS X | Linux