

Tecnologías para desarrollos en internet

Manual CRUD: Beego

Kihui-DEV

Fecha: 18/09/16
Facultad de Ciencias UNAM

Índice

1. Introducción	1
2. Instalación de Go 1.7	1
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	5
4.2. Prueba del servidor local	5
5. Elaboración del CRUD bajo MVC	6
5.1. Modelo	6
5.2. Controlador	6
5.3. Vista	6
5.4. Resultados	6
6. Referencias	6

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 Golang 1.7

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.²

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
```

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

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

²Instalador para Mac OS X

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**³ 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.

³La 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 Beego utilizamos el siguiente comando:

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

Para compilar y correr nuestros proyectos necesitaremos instalar Bee 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
```

4.1. Estructura del proyecto

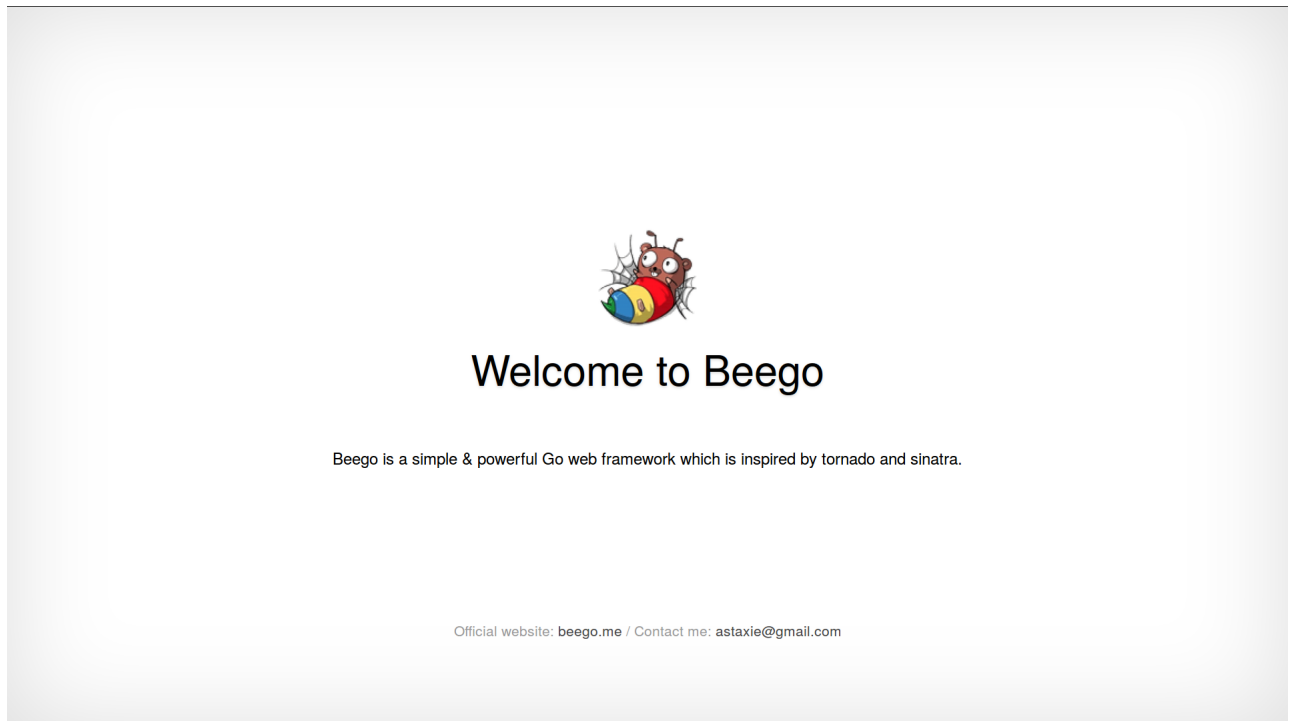
- 1) `conf/`
- 2) `models/`
- 3) `controllers/`
- 4) `routers/`
- 5) `views/`
- 6) `static/`
- 7) `tests/`

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:



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. Modelo

Base de datos

Configuración

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

Modelos

5.2. Controlador

Routers

Controladores

5.3. Vista

Plantillas

Estilo y agregados

5.4. Resultados

6. Referencias