

Lección 2 - Configuración del entorno

Tabla de contenido

- [Introducción](#)
- [Objetivos](#)
- [Desarrollo](#)
 - [Instalación de Gorilla y configuración de dependencias](#)
 - [Creación de la estructura del proyecto](#)
- [Conclusión](#)
- [Conoce más del autor](#)

Introducción

Configurar adecuadamente nuestro entorno de desarrollo es esencial para comenzar a trabajar con el framework Gorilla en Go. En esta lección, exploraremos los pasos necesarios para configurar nuestro entorno y empezar a desarrollar aplicaciones web eficientes y escalables utilizando Gorilla.

Objetivos

1. **Instalación y Configuración de Gorilla:** Aprender a instalar y configurar el framework Gorilla en nuestro entorno de desarrollo.
 2. **Explorar la Estructura Básica de un Proyecto Gorilla:** Familiarizarnos con la estructura básica de un proyecto Gorilla, incluyendo la organización de archivos y directorios para una mejor comprensión del flujo de trabajo.
 3. **Preparar el Entorno de Desarrollo:** Configurar nuestro entorno de desarrollo para trabajar eficientemente con Gorilla, incluyendo la instalación de dependencias y la creación de la estructura del proyecto.
-

Desarrollo

Instalación de Gorilla y configuración de dependencias

Creamos el directorio `Gorilla`

```
$ mkdir Gorilla
```

Primero, asegurémonos de tener Go instalado en nuestro sistema. Luego, abrimos la terminal y ejecutamos el siguiente comando:

```
$ go version
go version go1.21.0 windows/amd64

$ go get -u github.com/gorilla/mux
go: go.mod file not found in current directory or any parent directory.
'go get' is no longer supported outside a module.
To build and install a command, use 'go install' with a version,
like 'go install example.com/cmd@latest'
For more information, see https://golang.org/doc/go-get-install-
deprecation
or run 'go help get' or 'go help install'.
```

No te preocupes, aquí te explico cómo solucionar ese error: El error indica que estás intentando utilizar el comando `go get` fuera de un módulo Go. A partir de Go 1.16, se recomienda usar `go install` con la especificación de versión para instalar paquetes y comandos.

Para solucionarlo, sigue estos pasos:

- **Crea un módulo Go:**
 - Ubicate en el directorio donde desees crear tu proyecto Echo.
 - `cd Gorilla`
 - Ejecuta el comando `go mod init Gorilla` (reemplaza `nombre_proyecto` con el nombre que desees darle a tu módulo). Esto creará un archivo llamado `go.mod` que identifica tu proyecto como un módulo Go.

```
$ cd Gorilla
$ go mod init Gorilla
go: creating new go.mod: module Gorilla
```

- **Instala el comando Gorilla:**
 - Una vez que tienes el módulo Go creado, ejecuta el comando `go install github.com/gorilla/mux@latest` para instalar la última versión del comando Gorilla.

```
$ go install github.com/gorilla/mux@latest
```

```
...
```

```
package install...
```

```
$ go get -u github.com/gorilla/mux@latest
```

Esto descargará e instalará Gin y sus dependencias en tu proyecto.

Creación de la estructura del proyecto

Dentro de nuestro directorio de proyecto, crearemos la estructura básica de archivos y directorios.

Crear el fichero `main.go` y `routes/routes.go`.

```
$ pwd
/c/Users/Jerson/Documents/FrameworksWebEnGo/Gorilla

$ touch main.go
$ mkdir routes
$ touch routes/routes.go
```

El archivo `main.go` será responsable de crear el router y ejecutar el servidor. Importaremos el paquete `routes` y utilizaremos las funciones necesarias para registrar las rutas en el router.

El archivo `routes/routes.go` definirá las rutas de nuestra aplicación. Importará el paquete `mux` de Gorilla y utilizará sus funciones para definir las rutas. Además, definirá las funciones que manejan las solicitudes HTTP.

La estructura básica de nuestro proyecto se verá así:

```
Gorilla
├─ go.mod
├─ go.sum
├─ main.go
└─ routes
    └─ routes.go
```

Con la estructura de nuestro proyecto creada y Gorilla instalado, nuestro entorno de desarrollo está listo para empezar a trabajar en aplicaciones web con Gorilla. A partir de aquí, podemos

comenzar a desarrollar nuestras rutas, implementar lógica de negocios y crear una aplicación web totalmente funcional.

Conclusión

Configurar nuestro entorno de desarrollo es el primer paso crucial para trabajar con el framework Gorilla en Go. Al seguir los pasos descritos anteriormente, hemos establecido una base sólida para crear aplicaciones web eficientes y escalables utilizando Gorilla. Con Gorilla, podemos aprovechar al máximo las capacidades de Go y desarrollar aplicaciones web rápidas y seguras. ¡Estamos listos para comenzar a desarrollar con Gorilla y llevar nuestras ideas a la realidad!

Conoce más del autor

¡Encuéntrame en las siguientes redes sociales para estar al tanto de mis proyectos y actividades!

 Red Social	 Enlace
 Página web	jersonmartinez.com
 LinkedIn	Jerson Martínez - DevOps Engineer
 Canales de YouTube	DevOpsea Side Master
 GitHub	Perfil en GitHub
 Twitter (X)	@antoniomorenosm