

Lección 2 - Configuración del entorno

Tabla de contenido

- [Introducción](#)
- [Objetivos](#)
- [Desarrollo](#)
 - [Instalación de Echo y configuración de dependencias](#)
 - [Creando la estructura básica de un proyecto Echo](#)
- [Conclusión](#)
- [Conoce más del autor](#)

Introducción

En esta clase, nos adentraremos en la emocionante tarea de configurar nuestro entorno para comenzar a trabajar con el Framework Echo de Go. Configurar correctamente nuestro entorno de desarrollo es fundamental para aprovechar al máximo las capacidades de Echo y facilitar el proceso de creación de aplicaciones web y API RESTful.

Objetivos

1. **Instalar y Configurar Echo:** Explorar los pasos necesarios para instalar y configurar el framework Echo en nuestro entorno de desarrollo.
 2. **Explorar la Estructura Básica de un Proyecto Echo:** Aprender a familiarizarse con la estructura básica de un proyecto Echo, incluida 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 de manera eficiente con Echo, incluida la instalación de dependencias y la creación de la estructura del proyecto.
-

Desarrollo

Instalación de Echo y configuración de dependencias

Creamos el directorio `Echo`

```
$ mkdir Echo
```

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/labstack/echo/v4
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 deseas crear tu proyecto Echo.
- `cd Echo`
- Ejecuta el comando `go mod init Echo` (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 Echo
$ go mod init Echo
go: creating new go.mod: module Echo
```

- **Instala el comando Echo:**

- Una vez que tienes el módulo Go creado, ejecuta el comando `go install github.com/labstack/echo/v4@latest` para instalar la última versión del comando Echo.

```
$ go install github.com/labstack/echo/v4@latest
```

```
...
```

```
package install...
```

```
$ go get -u github.com/labstack/echo/v4
```

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

Creando la estructura básica de un proyecto Echo

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

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

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

El fichero `main.go`:

- Se encarga de crear el router.
- Se encarga de iniciar el servidor.
- Importará el paquete `routes` y usará las funciones. necesarias para registrar rutas en el router.

El fichero `routes/routes.go`:

- Se encargará de definir las rutas de la aplicación.
- Importa el paquete `Echo` y usa sus funciones para definir las rutas.
- Además, define las funciones que se encargan de procesar las solicitudes.

La estructura básica de nuestro proyecto se verá así:

```
Echo
├─ go.mod
├─ go.sum
└─ main.go
```

└─ routes
└─ routes.go



Con la estructura del proyecto creada y Echo instalado, nuestro entorno de desarrollo está listo para comenzar a trabajar en aplicaciones web con Echo. 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 Echo de Go. Al seguir los pasos descritos anteriormente, hemos establecido una base sólida para crear aplicaciones web y API RESTful utilizando Echo. Con Echo, podemos aprovechar al máximo las capacidades de Go y desarrollar aplicaciones web rápidas, seguras y escalables. ¡Estamos listos para embarcarnos en el emocionante viaje de desarrollo con Echo!

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