



GO

BOOTCAMP

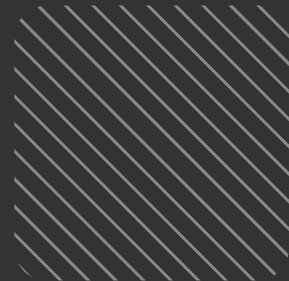


Clase en vivo

//Go Web

IT BOARDING

BOOTCAMP



Objetivos de la clase:

- Comprender qué son y cómo declarar variables de entorno.
- Implementar el uso del package estándar “os” para poder manipular variables de entorno en Go.

Índice



01 [Repaso](#)

02 [Variables de entorno en GO](#)

03 [Live Coding](#)

IT BOARDING

BOOTCAMP



1

Repaso

IT BOARDING

BOOTCAMP



Variables de entorno

IT BOARDING

BOOTCAMP



// ¿Qué son?

Son **valores dinámicos** que forman parte del sistema operativo (**os**), en **formato key - value**

Proporcionan **información y configuración**, que otros **procesos** pueden utilizar, afectando su **comportamiento**

// ¿Qué datos se suelen guardar?

- Ruta de directorios
- Ruta de archivos
- Otros valores de configuración

// Implementación

En Windows - Powershell

{ }

```
# Set environment variable  
$Env:KEY = "VALUE"  
# Get environment variable  
$Env:KEY
```

En Unix-Based System

{ }

```
# Set environment variables here.  
export KEY = "value"  
# Get the environment variable  
$KEY
```

// Variable de Entorno - Path

La variable **PATH** almacena rutas de directorios con **ejecutables**, que el **sistema operativo** utiliza para poder **ejecutarlos** directamente desde la **terminal**, sin necesidad de especificar su ruta.

En Windows - Powershell

```
{}
```

```
# Path (append with ";")  
$Env:PATH = $Env:PATH + ";" + "C:/Users/username/app/bin"
```

En Unix-Based System

```
{}
```

```
# Path (append with :)  
export PATH = $PATH:/path/to/bin
```

// Otras formas de Manejarlos

- **archivos .env:** Hoy en día muchas tecnologías soportan este tipo de ficheros: los propios IDEs, Docker y algunos frameworks web son solo unos pocos ejemplos de ello.
- **archivos toml, json, yaml, entre otros**

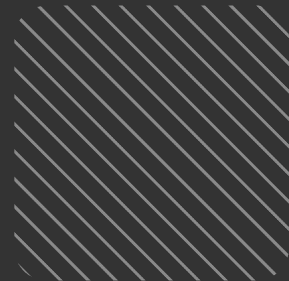


2

Variables de entorno en GO

IT BOARDING

BOOTCAMP





Package OS

El **package os** nos ofrece una serie de funcionalidades para trabajar con las **variables de entorno**.

- **os.Getenv:** nos permite **obtener** el **valor** de una **clave**, en formato **string**.
- **os.Setenv:** nos permite **guardar** un **valor** en una **clave**
- **os.Environ:** nos permite **obtener** todas las variables de entorno en un **slice** de **string** en formato **“key=value”**



Package OS

A continuación, un ejemplo de cómo **obtener** una **variable de entorno**, utilizando la función **os.Getenv()**

{}

```
package main

import "os"

func main() {
    // env
    filePath := os.Getenv("FILE_PATH")
    // open file
    f, err := os.Open(filePath)
    if err != nil {
        panic(err)
    }
    defer f.Close()
    // ...
}
```

Token en Variable de Entorno

IT BOARDING

BOOTCAMP





Token en Variable de Entorno

Cargar el **Token** en la sesión de la terminal

```
$ export Token = 123456
```

Utilizamos **os.Getenv** para obtener el Token

```
{}
```

```
func (p *Products) GetAll(w http.ResponseWriter, r *http.Request) {  
    // request  
    token := r.Header.Get("Authorization")  
    if token != os.Getenv("API_TOKEN") {  
        w.WriteHeader(http.StatusUnauthorized)  
        w.Header().Set("Content-Type", "application/json")  
        w.Write([]byte(`{"message": "unauthorized"}`))  
        return  
    }  
    // ...  
}
```


A codear...





3

Live Coding

IT BOARDING

BOOTCAMP



Live Coding





Conclusiones

En esta clase terminamos de comprender las variables de entorno, como guardarlas y obtenerlas.

Además aprendimos cómo utilizarlas en el entorno de Go



Actividad





Gracias.

IT BOARDING

BOOTCAMP

