

Synthèse - Quatrième semaine

Lors de ma quatrième semaine, j'ai pu découvrir l'utilisation des imprimantes 3D grâce à l'imprimante qu'on m'a prêté.

J'ai également continué mon apprentissage de Golang, ainsi que continuer la génération d'un fichier à partir d'un programme en Golang, en l'optimisant et en ajoutant les éléments demandés pour le projet.

Fichier "main.go":

```
package main

import (
    "fmt"
    "io/ioutil"
    "net/http"
)

func main() {
    // Définition de la route "/hello" avec la fonction de gestion
    http.HandleFunc("/hello", helloHandler)

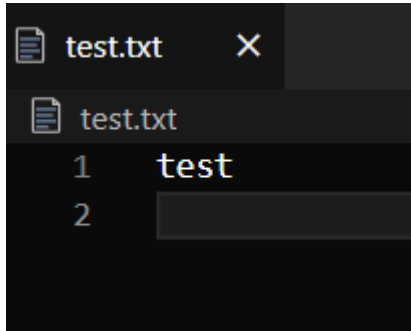
    // Démarrage du serveur sur le port 8080
    port := 8080
    fmt.Printf("Serveur écoutant sur le port %d...\n", port)
    err := http.ListenAndServe(fmt.Sprintf(":%d", port), nil)

    // Gestion des erreurs
    if err != nil {
        fmt.Println("Erreur:", err)
    }
}

// Fonction de gestion pour la route "/hello"
func helloHandler(w http.ResponseWriter, r *http.Request) {
    // Lecture du fichier
    data, err := ioutil.ReadFile("test.txt")
    if err != nil {
        // Envoi d'une réponse d'erreur au client
        http.Error(w, "Erreur de lecture du fichier",
            http.StatusInternalServerError)
        fmt.Println(err)
        return
    }

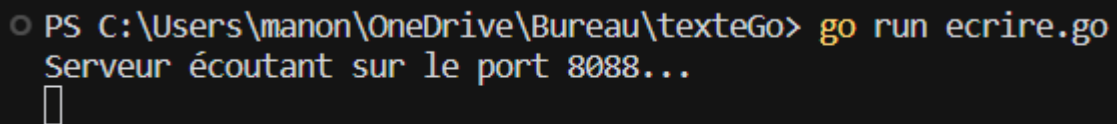
    // Envoi du contenu du fichier en tant que réponse
    w.Header().Set("Content-Type", "text/plain")
    w.Write(data)
}
```

Fichier "test.txt":



A screenshot of a text editor window titled 'test.txt'. The editor shows two lines of code: line 1 contains the word 'test' and line 2 is empty. The text is in a light blue font on a dark background.

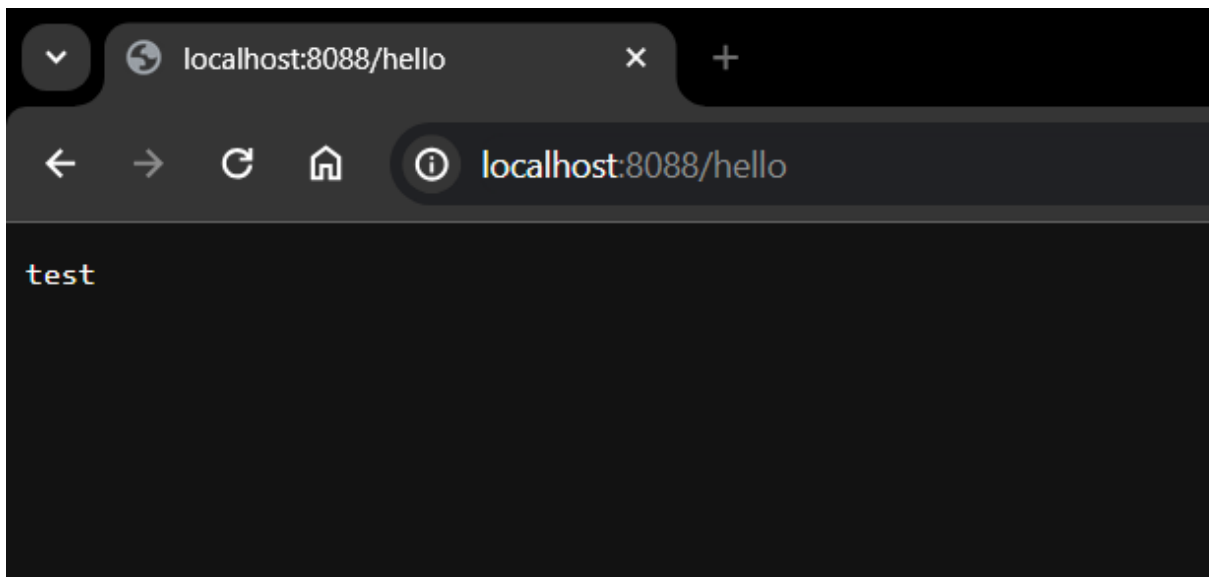
Commande Terminal :



```
PS C:\Users\manon\OneDrive\Bureau\texteGo> go run ecrire.go
Serveur écoutant sur le port 8088...

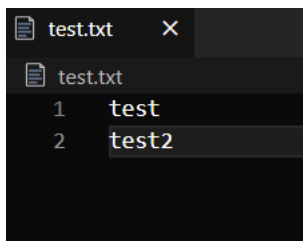
```

Résultat:



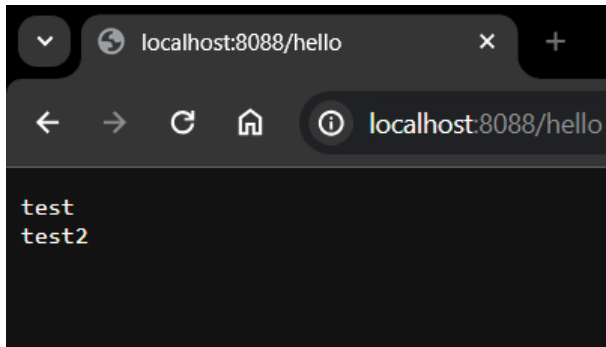
A screenshot of a web browser window. The address bar shows 'localhost:8088/hello'. The page content displays the word 'test' in a light blue font on a dark background.

Modification fichier test:



A screenshot of a text editor window titled 'test.txt'. The editor shows two lines of code: line 1 contains the word 'test' and line 2 contains the word 'test2'. The text is in a light blue font on a dark background.

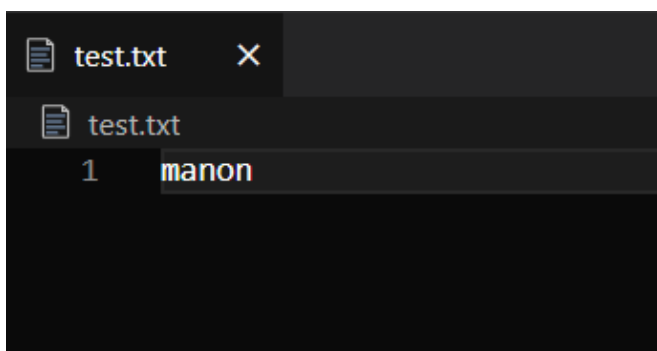
Résultat après sauvegarde et actualisation du navigateur:



Pour résumer les modifications, désormais, ce code en Go crée un serveur web très simple qui répond à une requête HTTP sur la route “/hello” en lisant le contenu d’un fichier texte appelé “test.txt” et en le renvoyant comme réponse.

Partie 2 : Création d’un input en Golang permettant de saisir un nom de projet à générer puis de l’enregistrer dans un fichier texte

```
PS C:\Users\manon\OneDrive\Bureau\texteGo> go run test.go
Saisir le nom du projet à générer: manon
Le nom du projet à générer a bien été enregistrée dans le fichier test.txt
test.txt
```



Fichier "main.go":

```
package main

import (
    "fmt"
    "os"
)

func main() {
    // Demander de saisir le nom du projet à générer)
    fmt.Print("Saisir le nom du projet à générer: ")

    // Déclarer une variable pour stocker le nom du projet
    var nomProjetInput string

    // Utiliser fmt.Scan pour lire le nom du projet et stocker la valeur dans
    la variable
    fmt.Scan(&nomProjetInput)

    // Nom du fichier dans lequel enregistrer le nom du projet à générer
    filename := "test.txt"

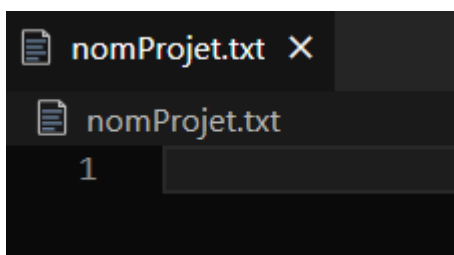
    // Ouvrir le fichier en mode écriture (créera le fichier s'il n'existe
    pas)
    file, err := os.Create(filename)
    if err != nil {
        fmt.Println("Erreur lors de l'ouverture du fichier :", err)
        return
    }
    defer file.Close() // Fermer le fichier à la fin de la fonction

    // Écrire le nom du projet dans le fichier
    _, err = file.WriteString(nomProjetInput)
    if err != nil {
        fmt.Println("Erreur lors de l'écriture dans le fichier :", err)
        return
    }

    fmt.Printf("Le nom du projet à générer a bien été enregistrée dans le
    fichier test.txt %s\n", filename)
}
```

Partie 3: Programme qui crée un répertoire avec le nom du projet saisi

Fichier "nomProjet.txt" avant saisie:



Fichier "main.go" :

```
package main

import (
    "fmt"
    "os"
)

func main() {
    // Demander de saisir le nom du projet à générer
    fmt.Print("Saisir le nom du projet à générer: ")

    // Déclarer une variable pour stocker le nom du projet
    var nomProjetInput string

    // Utiliser fmt.Scan pour lire le nom du projet et stocker la valeur dans
    la variable
    fmt.Scan(&nomProjetInput)

    // Créer un répertoire avec le nom du projet
    err := os.Mkdir(nomProjetInput, 0755)
    if err != nil {
        fmt.Println("Erreur lors de la création du répertoire du projet :",
err)
        return
    }

    // Nom du fichier dans lequel enregistrer le nom du projet à générer
    filename := "nomProjet.txt"

    // Ouvrir le fichier en mode écriture (créera le fichier s'il n'existe
    pas)
    file, err := os.Create(filename)
    if err != nil {
        fmt.Println("Erreur lors de l'ouverture du fichier :", err)
        return
    }
    defer file.Close() // Fermer le fichier à la fin de la fonction

    // Écrire le nom du projet dans le fichier
    _, err = file.WriteString(nomProjetInput)
    if err != nil {
        fmt.Println("Erreur lors de l'écriture dans le fichier :", err)
        return
    }

    fmt.Printf("Le nom du projet à générer a bien été enregistrée dans le
    fichier test.txt %s\n", filename)
}
```

Terminal:

```
PS C:\Users\manon\OneDrive\Bureau\texteGo> go run test.go
Saisir le nom du projet à générer: testNom
Le nom du projet à générer a bien été enregistrée dans le fichier test.txt
nomProjet.txt
PS C:\Users\manon\OneDrive\Bureau\texteGo> 
```

