Installation

> Dans le cadre de cette explication, j'utiliserais l'IDE Goland, donc il se peut que certaines choses soient simplifiés par l'IDE, et d'autres que je doive faire spécifiquement par rapport à cet IDE

Nouveau projet

Tout d'abord, nous allons créer un nouveau projet Go.

Untitled

> Dans l'environnement, j'ai spécifié `GOPROXY=direct`
> C'est très important, car cela va nous permettre d'inclure des librairies externes (Gin)

Vérifier dans vos paramètres Go / Go Modules que l'option **Enable Go Modules integration** est bien coché et ressemble à ça :

Untitled

Ajout des dépendances

Pour ajouter les dépendances nécessaires à Gin, nous devons ouvrir un terminal à la racine du projet (que ne dois contenir pour le moment que le fichier `go.mod`) et taper la commande suivante.

```
go get -u github.com/gin-gonic/gin
```

Cela va télécharger et indiquer dans notre fichier `go.mod` les dépendances nécessaires au fonctionnement de **Gin** :

Untitled

Création d'un serveur basique

Pour tester que tout va bien, nous allons créer le fichier `main.go` à la racine du projet et le remplir ainsi :

```
package main

import "github.com/gin-gonic/gin"

func main() {
	r := gin.Default()

	r.GET("/", func(c *gin.Context) {
		c.JSON(200, gin.H{
			"message": "hello world",
		})
	})

	r.Run(":8080")
}
```

Testons ce petit code rapidement avec la commande :

```
go run main.go
```

Untitled

Et voilà ! Vous avez votre premier serveur Gin qui est en marche !

Rendons-nous avec notre navigateur sur l'adresse localhost:8080 pour admirer la superbe réponse

Untitled

Untitled

Revision #3
Created 2022-05-09 19:23:34 UTC by Noé Larrieu-Lacoste
Updated 2022-05-09 21:42:12 UTC by Noé Larrieu-Lacoste