Chapitre 3
Installation et premiers pas

1. Installation de Go sur votre système (Windows, macOS, Linux)

Go est un langage multiplateforme et son installation est simple sur les principaux systèmes d'exploitation. Cette section détaille les étapes d'installation de Go sur Windows, macOS et Linux.

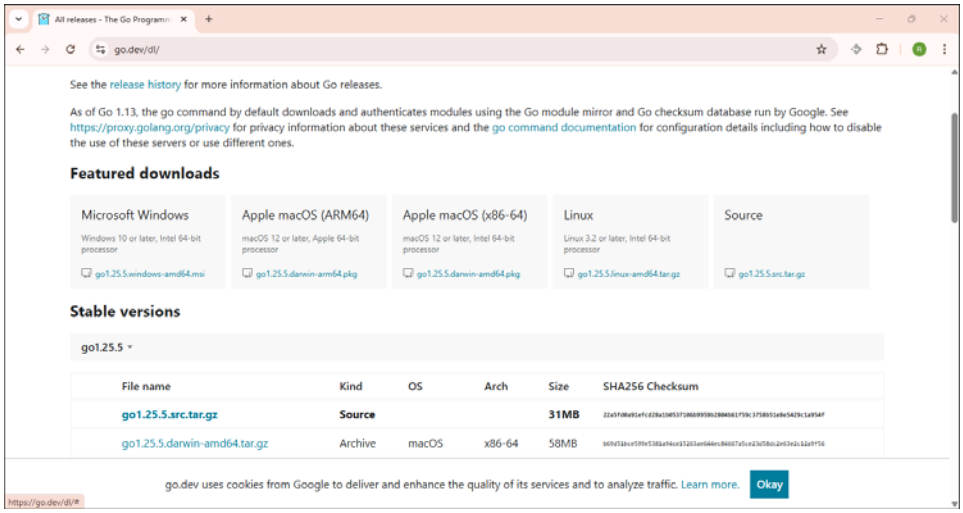
1.1 Installation sur Windows

L'installation de Go sur Windows suit un processus simple et accessible à tous les utilisateurs, qu'ils soient novices ou expérimentés. Cette section détaille chaque étape avec des explications approfondies pour garantir une installation correcte et fonctionnelle.

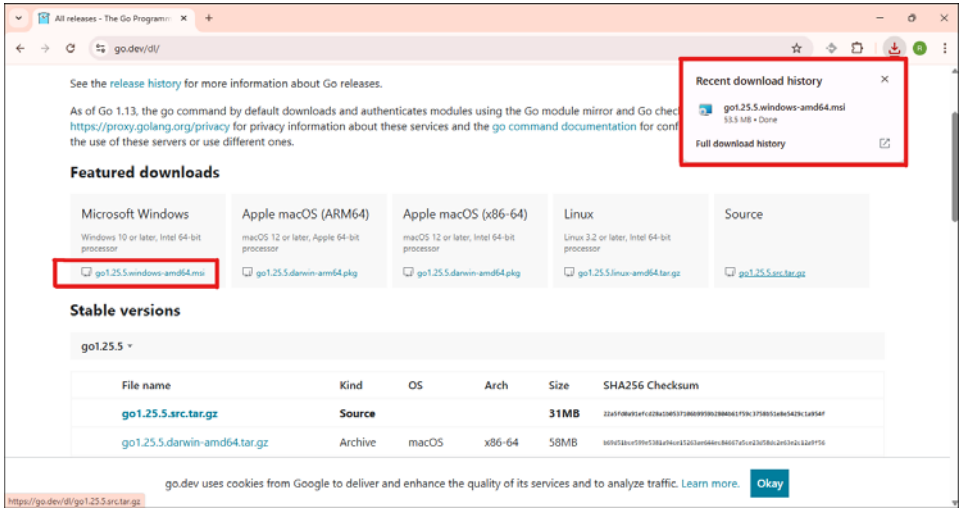
1.1.1 Télécharger Go

Le package d'installation de Go pour Windows est disponible sur le site officiel de Go. Il est conseillé de télécharger la version la plus récente afin de profiter des dernières fonctionnalités et corrections de bugs.

- Ouvrez un navigateur web et rendez-vous sur la page de téléchargement de Go (https://go.dev/dl/).



- Sélectionnez la version compatible avec votre architecture (généralement `Windows 64-bit` en `.msi`).
- Attendez la fin du téléchargement avant de passer à l'installation.



1.1.2 Installer Go

Une fois le fichier `.msi` téléchargé, l'installation peut commencer en suivant ces étapes :

- Double cliquez sur le fichier `.msi` pour lancer le programme d'installation.

Une interface graphique vous guidera tout au long de l'installation, en proposant les options suivantes :

– Accepter les termes du contrat de licence.
– Choisir le chemin d'installation (par défaut `C:\Go`).
– Valider les paramètres et lancer l'installation.

- Une fois le processus terminé, fermez l'assistant.

1.1.3 Vérifier l'installation

Après l'installation, il est essentiel de s'assurer que Go est correctement installé et accessible depuis le terminal.

- Ouvrez un terminal : appuyez sur [Win] R, tapez `cmd` ou `powershell`, puis validez.
- Vérifiez la version installée : exécutez la commande suivante :

```
go version
```

Cette commande doit afficher une sortie similaire à ceci :

```
go version go1.X.X windows/amd64
```

Si aucune version n'est affichée ou si une erreur est retournée, cela indique un problème avec l'installation.

Tester un programme minimal

- Pour confirmer que l'environnement Go fonctionne correctement, créez un fichier `test.go` avec le contenu suivant :

```
package main

import "fmt"
```

```
func main() {
    // Affiche un message pour confirmer que Go est installé
    fmt.Println("Go est bien installé sur votre système Windows !")
}
```

- Puis exécutez :

```
go run test.go
```

Si le message `Go est bien installé sur votre système Windows !` s'affiche, l'installation est fonctionnelle.

En cas de problème, vérifiez que le chemin d'accès à Go (`C:\Go\bin`) est bien ajouté à la variable d'environnement `PATH` de Windows.

1.2 Installation sur macOS

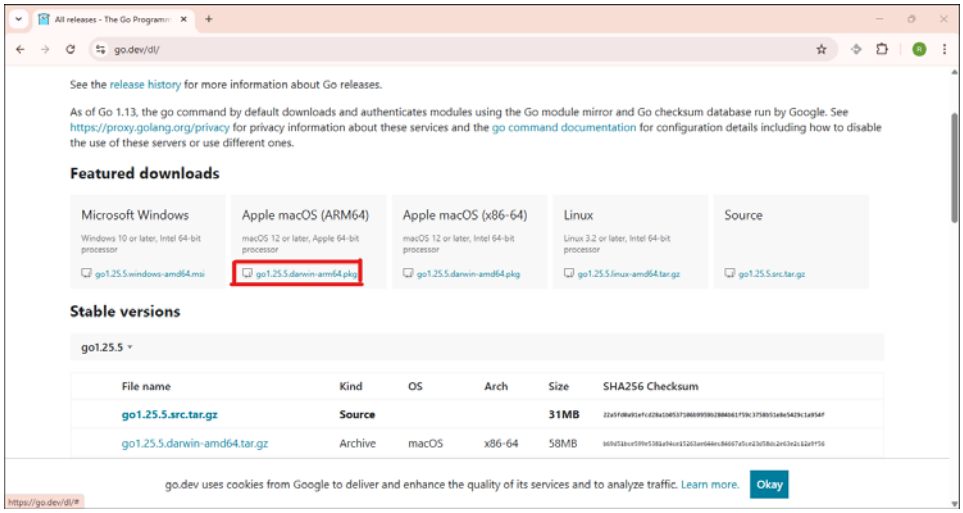
L'installation de Go sur macOS est un processus fluide, compatible avec les systèmes basés sur Intel et Apple Silicon (M1/M2). Cette section détaille plusieurs méthodes d'installation pour répondre aux besoins des différents utilisateurs.

1.2.1 Télécharger et installer Go

Il existe deux principales manières d'installer Go sur macOS : via le package officiel ou avec un gestionnaire de paquets comme Homebrew.

Installation via le package officiel

- Rendez-vous sur le site officiel de Go (https://go.dev/dl/).
- Téléchargez la version correspondante à macOS (`.pkg` pour architecture x86_64 ou arm64).



- Ouvrez le fichier téléchargé et suivez les instructions de l'assistant d'installation.
- Validez l'installation et fermez l'assistant une fois terminé.

Installation avec Homebrew

Homebrew est un gestionnaire de paquets populaire sur macOS, facilitant l'installation de logiciels en ligne de commande.

- Ouvrez le terminal ([Cmd] [Espace], tapez Terminal).
- Exécutez la commande suivante :

```
brew install go
```

- Attendez la fin de l'installation.
- Vérifiez que Go est installé correctement en consultant la version.

1.2.2 Vérifier l'installation

Une fois Go installé, il est nécessaire de confirmer que le système le reconnaît et que l'environnement est correctement configuré.

▶Ouvrez le terminal et tapez :

```
go version
```

Une sortie similaire doit apparaître :

```
go version go1.X.X darwin/arm64
```

Si cette commande renvoie une erreur, l'installation n'est peut-être pas correctement configurée.

▶Confirmez le bon positionnement du binaire avec :

```
which go
```

La réponse devrait indiquer un chemin comme :

```
/usr/local/go/bin/go
```

ou, en cas d'installation via Homebrew :

```
/opt/homebrew/bin/go
```

▶Testez un programme minimal en créant un fichier `test.go` :

```
package main

import "fmt"

func main() {
    // Affiche un message confirmant le bon fonctionnement de Go
    fmt.Println("Go est bien installé sur macOS
et fonctionne correctement !")
}
```

▶Exécutez le fichier avec :

```
go run test.go
```

Si le message `Go est bien installé sur macOS et fonctionne correctement !` s'affiche, l'installation est réussie.

En cas de problème, vérifiez que le bon chemin est ajouté à la variable d'environnement `PATH`. Une modification du fichier `~/.zshrc` ou `~/.bash_profile` peut être nécessaire en fonction du shell utilisé.

1.3 Installation sur Linux

L'installation de Go sur un système Linux peut varier en fonction de la distribution utilisée. Cette section décrit plusieurs méthodes pour installer Go proprement et configurer l'environnement de manière optimale.

1.3.1 Télécharger et extraire Go

L'installation manuelle est recommandée pour garantir que la dernière version stable de Go est utilisée.

- Ouvrez un terminal et naviguez vers un répertoire temporaire :

```
cd /tmp
```

- Téléchargez l'archive officielle de Go depuis le site officiel :

```
wget https://go.dev/dl/go1.X.X.linux-amd64.tar.gz
```

- Adaptez `go1.X.X.linux-amd64.tar.gz` à la dernière version disponible.
- Extrayez l'archive dans le répertoire `/usr/local` :

```
sudo tar -C /usr/local -xzf go1.X.X.linux-amd64.tar.gz
```

- Vérifiez l'installation :

```
ls /usr/local/go
```

Cette commande doit afficher le contenu du dossier Go.

1.3.2 Configurer le chemin d'accès

Pour utiliser Go depuis n'importe quel emplacement, il faut ajouter son chemin au `PATH`.

- Modifiez le fichier de configuration du shell.

Pour Bash :

```
echo 'export PATH=$PATH:/usr/local/go/bin' >> ~/.bashrc
source ~/.bashrc
```

Pour Zsh :

```
echo 'export PATH=$PATH:/usr/local/go/bin' >> ~/.zshrc
source ~/.zshrc
```

▶Vérifiez la configuration en affichant le chemin de Go :

```
which go
```

La sortie doit être `/usr/local/go/bin/go`.

1.3.3 Vérifier l'installation

▶Vérifiez la version installée :

```
go version
```

Une sortie similaire à `go version go1.X.X linux/amd64` doit apparaître.

▶Testez un programme simple. Pour cela, créez un fichier `hello.go` dans un répertoire de travail :

```
package main

import "fmt"

func main() {
    // Affiche un message de confirmation
    fmt.Println("Go est bien installé sur votre système Linux !")
}
```

▶Exécutez le programme :

```
go run hello.go
```

Si le message `Go est bien installé sur votre système Linux !` s'affiche, l'installation est réussie.

En cas de problème, vérifiez que `/usr/local/go/bin` est bien inclus dans le `PATH`. Un redémarrage du terminal ou du système peut être nécessaire pour appliquer les changements.

2. Configurer les variables d'environnement (GOPATH, GOROOT)

Après l'installation de Go, il est essentiel de configurer correctement les variables d'environnement pour garantir un fonctionnement optimal du langage. Les principales variables à configurer sont `GOPATH` et `GOROOT`.

2.1 Comprendre GOPATH et GOROOT

Lorsqu'on travaille avec Go, il est essentiel de comprendre les variables d'environnement `GOPATH` et `GOROOT`. Ces deux variables influencent la manière dont Go gère les fichiers sources, les dépendances et les binaires.

2.1.1 GOROOT : le répertoire d'installation de Go

La variable `GOROOT` désigne le chemin où Go est installé sur le système. Ce répertoire contient l'exécutable `go`, les bibliothèques standard et d'autres fichiers nécessaires à l'exécution de Go.

Vérifier le GOROOT

Pour vérifier le chemin actuel de `GOROOT`, on peut exécuter la commande suivante :

```
go env GOROOT
```

Une sortie typique pourrait ressembler à ceci :

```
/usr/local/go
```

Si `GOROOT` est incorrect ou non défini, il peut être nécessaire de le configurer manuellement en ajoutant la ligne suivante au fichier `~/.bashrc` ou `~/.zshrc` :

```
export GOROOT=/usr/local/go
export PATH=$GOROOT/bin:$PATH
```