

Lab1 - Go Systems Programming

1. Install Go

Go to GoLang Official Website (<https://go.dev/>) to install the language.

- Check installation successful

```
go version
```

- Where is Go installed?

```
which go
```

- Check Go environment variables

```
go env
```

One important variable is **GOROOT**, which is the path your go installed.

- Create HelloWorld.go

```
package main

import "fmt"

func main() {
    fmt.Println("HelloWorld")
}
```

- Build

```
go build HelloWorld.go
```

- Run

```
./HelloWorld
```

- Run without building

```
rm ./HelloWorld
go run HelloWorld.go
```

2. A Tour of Go

- Go through the **Basics**
<https://go.dev/tour/list>
-

3. GoLand

Go to GoLand Official Website (<https://www.jetbrains.com/go/>) to install the IDE.

- In your first open, there is a **A Tour of GoLand**. If you are not familiar with **JetBrains** product, go through that.
- In GoLand, your code are organised as a **Project**, which requires package management.

Go provides two ways to manage package: **GOPATH** and **Go Modules**.

- **GOPATH**

GOPATH is the original packages management.

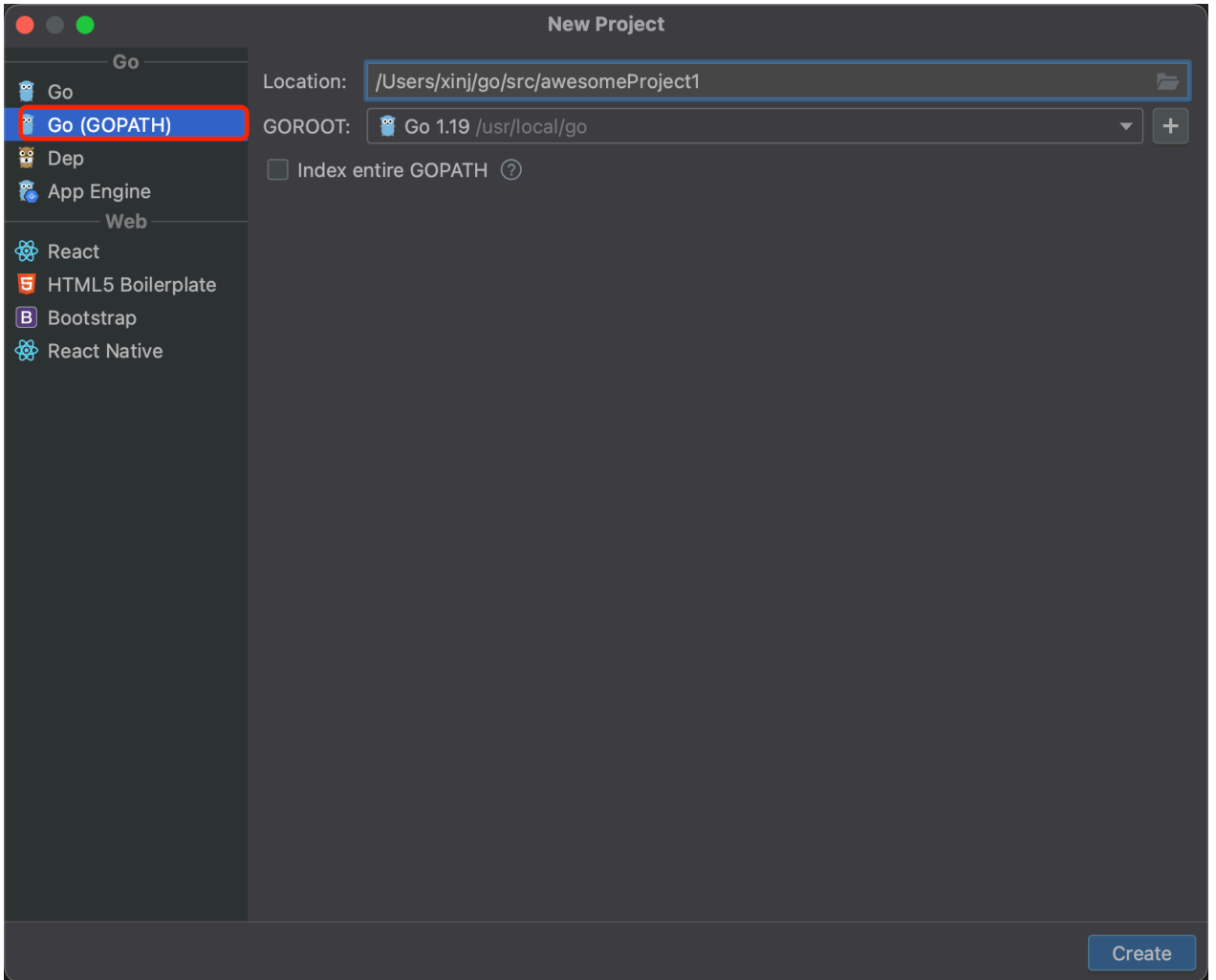
GOPATH is an **go environment variable**, by default set to `$HOME/go` which leads to the path of your project, you can check by:

```
go env GOPATH
```

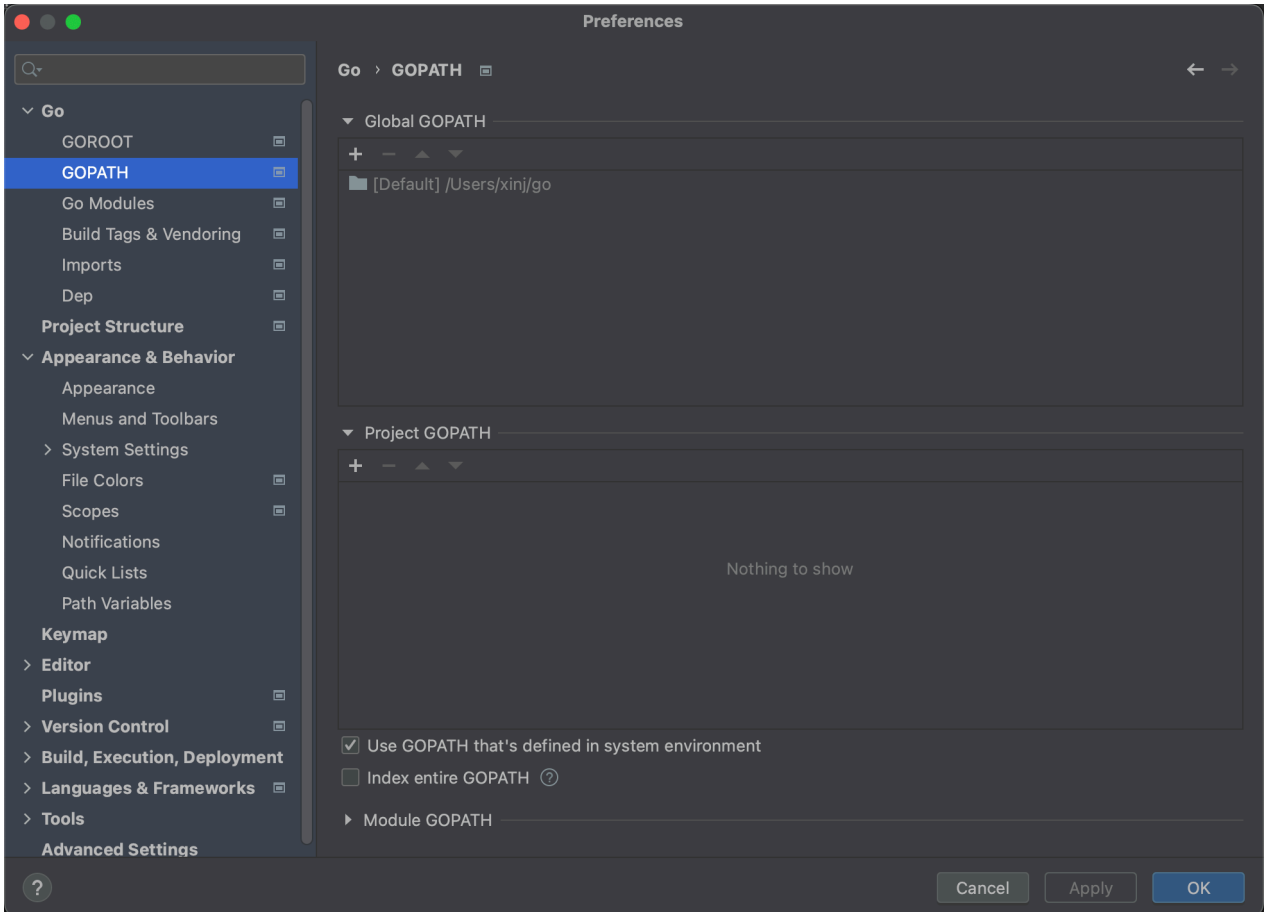
In this case, you should always work under your **GOPATH**, and the path should contains three folders:

- **bin**: location of compiled executable programs built by Go.
- **pkg**: location of compiled package code (for example, **.a**).
- **src**: location of Go source code (for example, **.go**).

In GoLand, you can create a GOPATH project:



If your project's is not under your default **GOPATH**, you should change the **Project GOPATH**:



Todo: create and run HelloWorld.go

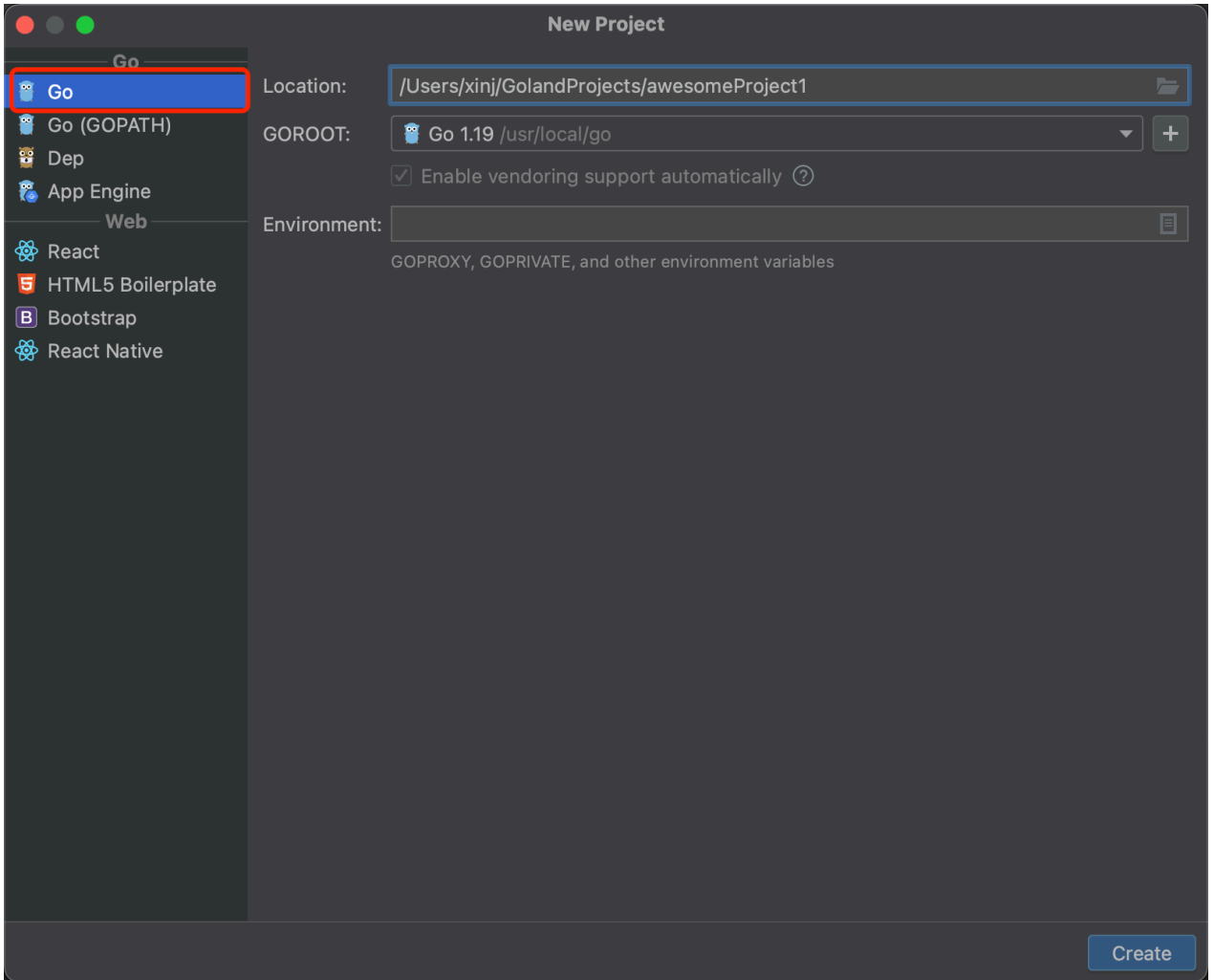
o Go Modules

GOPATH doesn't provide the flexibility version control, so we have Go Modules. (like venv or Go's internal conda).

Firstly, make sure your **GO111MODULE** is **auto** or **on**.

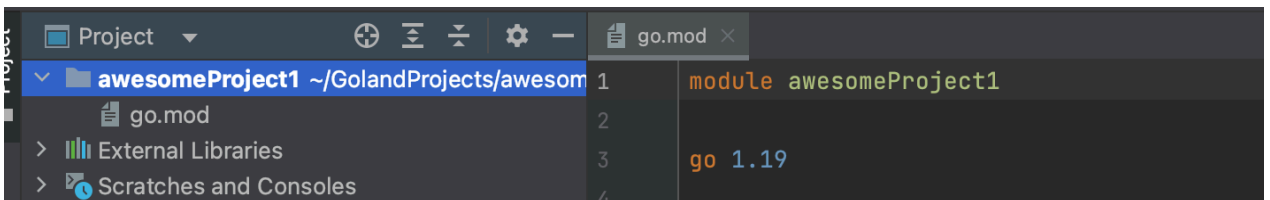
```
go env GO111MODULE
```

In GoLand, you can create a **Go Module** project:



The is now **not necessary** under your **GOPATH**.

GoLand will automatically add a `.mod` file (Description of the project and package dependencies).



Todo: create and run HelloWorld.go

- Compare

It is recommended to use **Go Module** as it is the latest approach, but as Google said:

"Go Module **does not entirely replace** `GOPATH`, but replaces `GOPATH` **for version control and package distribution.**"

So it is important to understand how **GOPATH** works.

(We will actually not need to use much third-party packages in the assignments, but it is important to understand this language.)

4. Go commands

- **go get**

```
go get golang.org/x/oauth2
```

Download the package and compile, usually used to get packages which you will import in your code.

ToDo: Find the package under GOPATH

- **go install**

```
go install golang.org/x/website/tour@latest
```

Install binary to your **GOPATH**.

ToDo: Install the tour from <https://go.dev/tour/welcome/3> then launch it.

- **go fmt**

```
go fmt <filename>
```

Format your code, without this you will lose marks in assignments.

Except go fmt, you should also be responsible for your code's readability and style.

GoLand support auto go fmt everytime you save a file.