**Basic**

1. What is Go?
2. Why should you learn Go?
3. What is Go suitable for?
4. Explain advantage and disadvantage of Go?
5. How Go is different from the programming languages that you have used? (Java, C/C++, Python,Rust, NodeJS, JavaScript)

**Array and Slice**

1. What are the differences between slice and link list?
2. How re- slicing process of a slice happen under the hood?
3. Can you modify array length? Can you append new item into an array?
4. Explain how different between array/slice in Go and array in C/Java?

**String**

1. How is string in Go constructed? And what is the different between string in Go with other programming languages?
2. What is string literal? How is it done in Go?

**Goroutine**

1. What are the difference between goroutine and OS thread?
2. Explain what is go routine in GO? How you can stop go routine?
3. What is a worker pool? And how to do that in Go?
4. How to stop a worker pool?
5. What is fan-in, fan-out?
6. What is sync.WaitGroup used for?
7. What are the differences between sync.Mutex and semaphore?
8. What are the differences between concurrency and parallelism?
9. What are the differences between sharing by communicate vs communicate by sharing?
10. What is channel? How does it work? What is it used for?
11. What is context? What is it used for?
12. Have you used goroutine in your application? Describe it.
13. Why goroutine is better than thread?
14. What is goroutine leak in Go? How to prevent it?

**Interface**

1. Explain Interface in Go
2. How interface in Go different from interface in other languages?
3. When should you use interface?
4. How did you use interface in your previous project?

**HTTP**

1. Have you ever used any framework for working with HTTP in Go? Compare it with native HTTP library provided by Go.
2. What is middleware? Have you ever written one?

**Logging**

1. Explain how did you implement logging component in your last application with Go?

**Performance**

1. How many query/second of your application?
2. How can you measure performance of your application?
3. How do you scale up your application?

**Others**

1. Does Go pass by value or pass by references? What are the differences?
2. What is Go run time?
3. How does garbage collector work in Go?
4. What is inside the interface{}?
5. How to handle gracefully shutdown in Go?
6. Why would you prefer to use an empty struct{} ? Provide some examples of the good use of the empty struct{}
7. How do you compare two structs? What about two interfaces? Provide examples.
8. What wrongs with following snip of code?
9. How do you structure your Go projects? Explain the advantages and disadvantages?
10. How do you manage configurations of your services in Go?
11. What is Eleaticsearch?
12. Advantage of elasticsearch?
13. What is mongodb?
14. MongoDB vs Mysql?
15. When should use mongodb?
16. What is docker?
17. What is docker file?
18. What is docker compose?