## CODE REVIEW IN GO TOP 5 MISSES

Sadie Freeman Quinn Hou Gophercon Nov 2023



#### **TOP 5 MISSES**





- Us and our experiences
- Our Top 5 things to look out for
  - Code Review hygiene

#### **QUINN HOU**

Senior Software Engineer

#### **SADIE FREEMAN**

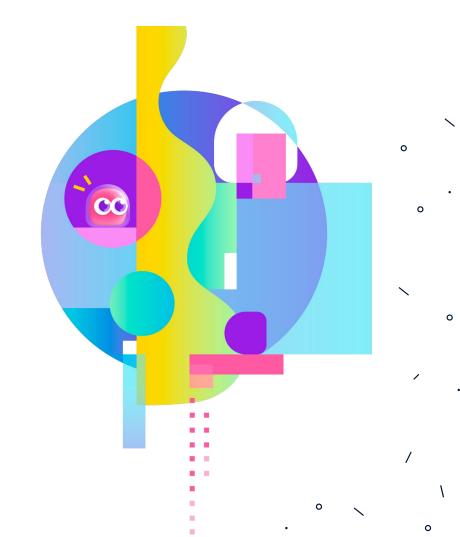
Senior Software Engineer

#### Dapper Labs









### 2021...





## ERROR HANDLING

#### **BEYOND IF ERR!= NIL**





#### **WRAP ERRORS**

"Should this error be wrapped?" fmt.Errorf(%w)

"How can we leverage error.Is, error.As, error.Unwrap?"



#### **CUSTOM ERRORS**

"Should we define custom errors here?"

#### **ERRORS ETIQUETTE**





#### **ERROR VS LOGGING**

"How do we want to handle this behavior?"

"Is it better to return an error or log?"



#### FALLBACK MECHANISMS

"Is the handling of this error here sufficient?"

"Defer, panic, recover?"



#### **GOROUTINE GOTCHAS**



#### **NECESSITY**

"Do we need a Goroutine here, or is it simpler to run this synchronously"

#### UNDERSTANDING

"What does this Goroutine do?"

#### **GOROUTINE GOTCHAS CONTINUED**

#### **WAITGROUP**

"Do we need to use WaitGroups here to see when the goroutines have finished?"

#### RESOURCE

"Do we do anything to cap how many Goroutines can be spun up at once?"



Go: panic: runtime error: invalid memory address or nil pointer dereference

#### **DEALING WITH NIL**





#### ZERO VALUES

"Is the zero value for this type nil?"

pointers, maps, slice, functions, channels, interfaces



#### **NIL POINTER?**

"When a field is accessed, will a nil pointer occur?"

"Did we check for possible erroneous values?"



#### NILS WITH INTERFACES

"Is this an interface?"

"Is a method called on a nil interface?"

Check nil on Interface types with **switch** on various types

#### WHEN TO USE WHAT





#### **POINTERS**

"Does this need to be mutable?"



#### **VALUES**

"Is this naturally a value type?"

"How are we referring to the variable throughout the function?

#### **POINTERS GOTCHAS**







"Are we copying a pointer in a loop?"

for \_, v := range pointers {
 bad := &v
}



#### **SYNC FIELDS**

"Are we copying synchronization fields?"



#### UNNECESSARY POINTERS

"Is this pointer necessary?"

0

# O4 INTERFACES + TESTING

#### **INTERFACES + TESTING**



#### **TOO MANY INTERFACES**

"Can we consolidate this into one interface that supports both methods?"

#### START SMALL

"Do we need all these functions or can we have a smaller interface to pass in here"

#### **USE INTERFACES FOR MOCKS**

"Could we use an interface for this so it can be a unit test?"

## UE LITTLE THINGS

0

0

#### **STYLE**



#### **VALIDATE INPUT**

"Are the input fields validated as needed?"

#### COMMENTS

"Are the comments helpful and necessary?"

"Is a comment needed here?"

"Does the comment have correct format and grammar?"

#### **PACKAGE NAMES**

"Is it lowercase, single word?"
"Does it avoid repetition?"

#### **VARIABLE NAMES**

"Are the variable names clear, succinct, and readable?"

#### **DECLARING EMPTY SLICES**

"Is the slice declared as a nil slice value?"

#### // TODO:



#### **USE YOUR TOOLS**



#### **BE HUMAN**

0

Errors
Concurrency
Nils
Interfaces
Style

Make things explicit and readable

Be open and curious. Ask questions.

Participate in collaboration 🐥



## THANK YOU



@quinnhou
 @sadie-f