THE !EMPATHETIC ENGINEER

A SHORT STORY PORTRAYING THE SOFTWARE ENGINEER ARCHETYPE

HUSSEIN NASSER

V0.0.2

The !Empathetic Engineer

A short story portraying the software engineer archetype

by Hussein Nasser

For Nada.

Hussein Nasser

About the Author

My name is Hussein and I’m a software engineer. I knew I had passion for building software when my uncle gave me my first programming book in 1998 titled “Learn programming with Visual Basic 2”. I started my blog, and YouTube channel as outlets to talk about software. Using software to solve interesting problems is one of the fascinating things I really enjoy. Feel free to contact me on my social media channels to tell your story, ask questions or share interesting problems. I would love to hear it! I also specialize in the field of geographic information systems (or GIS). I helped many organizations in different countries implement GIS technology and wrote custom apps to fit their use cases and streamline their workflows since 2005. I wrote five books on Esri ArcGIS technology, recorded over five hundred YouTube videos and software tutorials being watched until this day and started a software engineer podcast in 2018. I've been an engineer at Esri building cool GIS technologies since 2015.

First published Jan 2020

Updated June 2022

Hussein Nasser

Preface

Engineers are incredibly smart. This could be a product of the complex problems they usually work on and the abundance of solutions a given problem can have. This is especially true for software engineering problems. With so many roads leading to Rome, it becomes attractive for engineers to favor the idea they came up with over others.

Engineers are arrogant. I say that as an engineer myself. I have been guilty of assuming I know more than most people in the room. Or that my way is the right way. Or that whatever other people might say is irrelevant. Or that I can rarely be wrong.

Engineers might seem agreeable on the surface but in their own head, they are not. They might pretend to listen to your idea, but in reality, they are just waiting for you to finish so they can pitch theirs. You probably expect me to say “In this book, I will teach you how to no longer be arrogant!” "Watch how I listen to others and value their opinions.” “Watch me how I put myself in other engineers’ shoes and think like them” NOPE, I think I still need to work on that. However, I am aware of the problem which is a good start, I think.

If you are an engineer, this book will not give you a path to become more empathetic, listen more or value others' thoughts This book is merely a portrait I painted of the current state of software engineers as I perceive it from my 15 years career as a software engineer who worked in different companies in 7 different countries.

I wanted to shed light on this problem and highlight it and I’m using short stories as a medium. I did think about other mediums to portray this problem such as paintings, sculptures, poems, photos, music, and writing. However, I can’t paint or sculpt. I take atrocious photos (ask my wife). I never wrote any music so let's not go there. I do write poems but they are mostly in Arabic so that won’t reach the audience I want. Plus I think poems are immutable and can’t be translated without bleeding them in the process So the best route is to write. I know I'm not a good writer, as you will clearly see from this book, but I think I will be able to get my point across.   This is a self-published book and available only in digital. This will allow me to easily make edits and add more to the work.

Please feel free to send me any of your thoughts to my email h@husseinnasser.com, typos, criticisms or future ideas! I love to hear them. They might make the next update to the book.

Enough “I”s. This story is about you. The !empathetic engineer. I !strive to be like you. You are smart and !down to earth. A !good listener and a !better lead. And what’s more important is that you !know what you don’t know.

This is your story.

Enjoy

Hussein Nasser

Them

They write front-end apps hitting the REST endpoint that you’ve built. They were tasked to show unique values in a drop-down list in the app.

They use a client runtime with their favorite language which calls your endpoints. There was no `queryUnique` method in the runtime library. They ask the runtime developer to add that method.

The developer adds a new method `queryUnique`, which calls the query REST endpoint with `returnUnique` boolean as clearly stated in your REST doc.

You

You are the back-end engineer. You maintain the REST API, the networking stack, and the database.

They write front-end apps hitting your endpoints on your server. One of their apps performs badly when retrieving unique values. You get to work.

You intercept the query in the database and you notice that their app is doing a full-table scan retrieving all rows in a table with a million rows.

"Why are they pulling all rows from the table, that is stupid." You said

Intervention

"The app is doing a full table scan retrieving all rows that is why it takes a long time." You said confidently to them. "What are you trying to do?"

"That doesn't sound right”. They said. "I mean, we recently added a new method in the client runtime to return unique rows for a column."

"What a bunch of tools", You said to yourself. "Are you doing the filtering in the client? You really shouldn't, you should use returnUnique parameter in the REST API"

"That's what we used, here is the code." Said the runtime developer proudly showing his code.

"That looks .. right.." You said after a nervous pause.

Can you step in?

Convinced they are still doing something wrong you asked. “Can you step into the function?”.

Knowing that you just stepped over the line you quickly followed up: “ah I just want to make sure we are sending the correct parameters.” You said, using the 'we' pronoun, despite sounding condescending, you want to give the impression that no one is at fault here, 'we' are working together towards a common goal; Fixing this bug.

“Building the URL...” The developer while debugging, “Preparing parameters, injecting the token. Yeah, I am calling your REST endpoint correctly. That is your endpoint right?” said the developer, annoyed.

“Wait!” You yelled excited, "Are you passing the isUnique parameter as a boolean or a string?"

“Boolean.”

“Oh, ok. just checking because the server might cast ....”

“I know the difference between a string and a boolean!“ The developer interrupted you, clearly insulted.

”Sorry, I didn’t mean that, let me check the REST endpoint.”

Would you like a Biscoff cookie?

“I’m telling you, dude, all I asked him is to check the data type of the parameter and he freaked out.” You said to your colleague.

“How slow is this anyway?” Your colleague said, “Is it worth diving into the API? You know I’m not going to do that right?”

“It is bad., a few hundred concurrent requests of those and the database is toasted. “ You said. “The dev who wrote the API left. And we all know why!“

“It's not my fault I told her we were never exclusive.”

“Plus, no one else knows Golang here except you,” You said

“Don’t patronize me!”, Your colleague snarked

“Listen, dude please do me a favor this issue is marked must-ship.” You said patting your colleague back. “Plus, the PO is busting my balls. Listen, I prepared the repro case all you have to do is run this script. get in there and show them who's boss”

“Screw you.”

“That's the spirit! Would you like a Biscoff cookie?”

“Sure,” Said your colleague as he git clones the REST API repo.

Next morning,

“Have you seen Slack this morning?” Your colleague said. “frontend ppl are pissed we broke their code. They are asking to roll back our change”

Hussein Nasser

“I'll go upstairs and talk to them,” You said.

To be continued

I’ll add that later

“You were right.” Your colleague said. “They are passing a string ‘True’ to the isUnique parameter instead if the boolean true.

“I knew it,” You said, “What a bunch of tools”

“Still, I think we should probably fix this so it fails if the input parameter is not the correct data type” Your colleague paused. “It is so weird touching her code again. Look how she uses spaces instead of tabs. So Cute. I think I miss her”

“A) you shouldn’t have cheated on her and B) Focus! Let's add the guard logic and check-in. Ill code review. “I think it's a pretty straightforward fix”

“Are you sure that this is what is causing the problem?‟

“Yeah Passing a string to a boolean is bad. We don’t want room for guesses and type conversion. Make sure we are sending the correct type. I think that's why your girlfriend. Sorry. ex-girlfriend chose GoLang over Javascript.”

“Sigh.. she always hated Javascript. I still don’t understand how they’re scanning the entire table to get unique values. They must be paging in the client because the query REST endpoint only returns 100 rows max.” Said your colleague

“Who knows what those bozos are doing. They might be falling back in their code and paging the entire table instead.”

“I don’t know, I still think we should debug some more. Do we even have a unique query test?” Asked your colleague

“I'll add that test later it’s a one-line code. No need to continue debugging, the problem is clear. Let's get this in first” you said dismissively

“Btw this will break existing clients who rely on this working”

“Good! That will teach them to write better code” You said smiling

I thought you had a breakpoint

“So our entire code base is broken today, we think its your install.” They said

“Yeah we had a bug where we were incorrectly casting a string to a boolean” we decided to correctly fail if the client sends us a string instead of a boolean. Should be a quick fix at your end” Notice how you started to use the pronoun “us” now.

“I thought we showed you guys that we are sending a boolean. Are you back to that?”

“I’m sorry don’t take this personally but there must be some code deep down in your repo that sends the requests? Can you possible check that”

“I can do that,“ said another developer already stepping into the code. “Ok we send the boolean to the sendRequest method. Stepping into the method... yeah they are doing an explicit conversion from a boolean into a string just before sending the request. Hmm not sure why.”

“*Tools*” you said to yourself with A half smile. “Great! Can you remove that conversion and try again”

“Done, it's working again.” The dev said

“Nice!, “ you said “ yeah passing incorrect parameters causes all kind of problems plus it is cheaper to send a boolean into the wire than a string I'm sure you kno..”

“Why is this still running? It's been more than 10 seconds. Now that we are using the correct function, this should’ve been instant.” The developer said

“It could be slow because you’re debugging.” You said, regretting it the moment you said it.

“Has nothing to do with me debugging, the request is out and it's clearly churning on the server I'm just a-waiting here. Debug or no debug” the developer said looking at you.

“Yeah yeah of course, I thought that you had a breakpoint somewhere and the symbols are lazy-loading” you said embarrassed. “I'll take a look, one last thing. Is this the deepest you can step in? I just want to make sure we are not doing paging.” There is the ‘we’ again

“No we are not paging, we are sending a single request, and awaiting it.”

“Thanks, I'll take a look”

DISTINCTS

”This is bs.” Said your colleague. “How many times should I step into this code?”

“That is the last time, I promise. “ you said

“Ok getting boolean, isUnique true, stepping to queryUnique method. What the..?”

“What did you find?”

“Nevermind its just a lot of code for a function that isn’t supposed to be that long,” your colleague said

“Go on then”

“I don’t see anything wrong, she is connecting to the DB, inefficient we could pool those connections I guess”

“Still that can't be why it is so slow”

“Here is the query, she is using select distinct.”

“Let me look at the query, **SELECT DISTINCTS, TYPE FROM** … Nothing looks wrong” You said

“ not sure but this is clearly returning all rows in the type column. Which means we are doing the filtering on the server. Yeah, that is a fall-back method that just adds all values to a dictionary.”

A few hours later...

“Is the correct bind variable being used in the query? What version of the DB client library are we using?” You said

“Yes everything seems right dude, it’s been hours, we could have rewritten the entire code base by now instead of debugging as to why the query is returning all rows.” Said your colleague “And is it DISTINCT or DISTINCTS can’t remember” Your colleague said.

“It is DISTINCT, wait..” You said surprised “How is this even working, the query should have failed there is nothing called DISTINCTS in SQL last time I checked”

“There is a comma after DISTINCTS.” Your colleague said, “.. is there a column called DISTINCTS in the database?”

“Of course not! I don’t know, let me check. What the heck, it's a column on that table! Who added that” You said.

“You tell me, you manage the database”

“Going through the log, last added 3 months ago, there were no comments, wait there is a bitly.”

“That is for cheating on me asshole”

“Your ex says hello.”

Hussein Nasser