# TSA Innovation Lab – Practical Code Challenge – Documentation - Chung Ho

## Initial thoughts

Looks like I need create a API to accept data from an external source and need to store the data into a database. I assume that is what the word “persist” means. Along with data validation but assume that security is provided by an outer layer in the architecture.

I am to use the “gin\_gonic” framework for this challenge.

To do serve HTTP, I know there is package to handle this, never seen it, hope it work like I think it should.

Phone number must be in E.164 format. I never heard of this, learning something new already, will need to look this up. Got a gut feeling and regular expression will be needed here.

### Assumptions of the data and the database

Full\_name: I have some doubts that the database will store this as full name. More likely there is surname/family name and first name. May even have “other names”. The sample data provide a perfect two-name approach. But the real world is not that simple. For the purposes of this challenge, I will store the data as full name as one field. BUT if I have time, I will split the string by space and assume “<First name> <Surname>” approach.

Email: hopefully I can find a regular express for GO that I can use to verify the email

Phone\_numbers: E.164 I need to look this up as this is new to me. Looks like the one-to-many structure. Looking at the data types coming in, we need to deal with strings.

### Assumptions on the request object to the API

I am assuming the data must all be filled? So full name, email and phone number can not be empty.

## Approach

* Need to get the gin\_gonic package and look it up. “go get github.com/gin-gonic/gin”
* Need examine if the net/http package as I believe this is how to serve HTTP and accept data.
* Should I accept the payload as JSON? Still thinking about it
* Document the method.

### Coding approach

* Get the data structure.
* Code a simple HTTP server to test I got HTTP server right.
* Code something simple with gin to make sure I got the framework right.
* Code something to access a database
* Try and server some data from the database
* Start code the challenge and hit all the point requested.

## Design

I imagine that there is a main piece of code in which is the web server. This is ideally should be as thin as possible.

There should be modules/layers where it returns the data from data. A service layer or handler to validate the data coming in. Or could all be in one depending how complex the program is going to be.

The data should be the main contact with phone numbers in a separate table. The code would likely use the name as the primary key. BUT in production there should a better implementation such auto incremental number or a GUID. The foreign key for the phone numbers should also follow suit.

## Begin implementation

### Starting with gin

Started importing gin and got that working.

### Trying to do a database

Started with sqllite, but could not get it to compile on my windows environment. It might be the version of gcc I have, but wasted almost two hours on this so I gave up and try SQL Server.

I manage to find a sql package for go to work and started using that. First attempt to get data from database to return JSON object and server as a GET method. Of course the phone number data structure is not correct as “Chung Ho” should only appear once. I am just happy to get something working after about fourish hours.

