# Task 1 | Contact Tracing API

The task is to develop a basic version of contact tracing API. You are only required to develop the API for the system. Below are the details.

Users should have the following Attributes. All fields are mandatory unless marked optional:

* Id
* Name
* Date of Birth
* Phone Number
* Email Address
* Creation Timestamp

Contact should have the following Attributes. All fields are mandatory unless marked optional:

* UserIdOne
* UserIdTwo
* Timestamp (time of contact)

You are required to Design and Develop an HTTP JSON API capable of the following operations,

* Create a User
  + Should be a POST request
  + Use JSON request body
  + URL should be ‘/users’
  + Must return the user in JSON format
* Get a user using id
  + Should be a GET request
  + Id should be in the url parameter
  + URL should be ‘/users/<id here>’
  + Must return the user in JSON format
* Add a contact
  + Should be a POST request
  + Use JSON request body
  + URL should be ‘/contacts’
  + Must return the contact in JSON format
* List all primary contacts within the last 14 days of infection
  + Should be a GET request
  + URL should be ‘/contacts?user=<user id>&infection\_timestamp=<timestamp>’
  + Must return an array of users in JSON format that have been in contact with the given user id within 14 days ending on infection\_timestamp. I.e. all users within (infection\_timestamp-14days, infection\_timestamp)

Additional Constraints/Requirements:

* The API should be developed using [Go.](https://golang.org/)
* MongoDB should be used for storage.
* Only packages/libraries listed [here](https://golang.org/pkg/) and [here](https://pkg.go.dev/go.mongodb.org/mongo-driver@v1.4.0) can be used.

Scoring:

* Completion Percentage
  + Total working endpoints among the ones listed above.
* Quality of Code
  + Reusability
  + Consistency in naming variables, methods, functions, types
  + Idiomatic i.e. in Go’s style
* Make the server thread safe i.e. it should not have any race conditions especially when two meetings are being booked simultaneously for the same participant with overlapping time.
* Add [pagination](https://nordicapis.com/everything-you-need-to-know-about-api-pagination/) to the list endpoint
* Add [unit tests](https://golang.org/pkg/testing/)

Resources:

* Completing the [Golang tour](https://tour.golang.org/) should give one a good grip over the language. Do this well and you will complete the task with ease.
* This [article](https://golang.org/doc/articles/wiki/) should give you an idea on getting started with Web Application Development in Go.
* This [book](https://astaxie.gitbooks.io/build-web-application-with-golang/en/) covers both the workings of web and Go based servers.
* [This](https://github.com/mongodb/mongo-go-driver#usage) covers getting started with MongoDB in Go.

**Task Submission:** <https://bit.ly/appointy6mtech>

**Chat Server Link**: <https://discord.gg/TpGXznrdRe>

**Deadline**: Saturday, 12 PM (December 12, 2020)