

# Getstarted

1. Go lang: v1.19
2. Prepare db connection
3. Run script in folder: docs/database.sql
4. Create .env from .env.sample and update db connection info
5. FYI: All available API route is in file: router/router.go
6. Run following command

```
// this command is to auto map module into checksome and check error  
go mod tidy
```

```
// build project  
go build
```

```
// build and run  
go run main.go
```

It will show as

```
[..]  
[GIN-debug] Environment variable PORT="8080"  
[GIN-debug] Listening and serving HTTP on :8080
```

So can access via: <http://localhost:8080>

Example:

```
// Test via CURL
curl http://localhost:8080/users

// Expected result example
StatusCode      : 200
StatusDescription : OK
Content         : [{"id":1,"name":"Bong T","email":"bongt@gmail.com"}]
RawContent      : HTTP/1.1 200 OK
                  Content-Length: 52
                  Content-Type: application/json; charset=utf-8
                  Date: Sat, 20 Jul 2024 04:40:29 GMT

                  [{"id":1,"name":"Bong T","email":"bongt@gmail.com"}]
Forms           : {}
Headers         : [{"Content-Length", 52}, [{"Content-Type", "application/json; charset=utf-8"}, [{"Date", "Sat, 20 Jul 2024 04:40:29 GMT"}]]
Images          : {}
InputFields     : {}
Links           : {}
ParsedHtml      : mshtml.HTMLDocumentClass
RawContentLength : 52
```

# Explanation

Repository/Service Implementation (All in folder: `internal` )

1. Model: `models/*.go`
2. Repository: `repositories/*.go`
3. Service - handlers: `handlers/*.go`
4. API Router or Controller: `router/router.go`

Utilities related

1. Configuration: `config/config.go`
2. Database connection: `database/database.go`