

Task 1 | Meetings API

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

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

- Id
- Title
- Participants
- Start Time
- End Time
- Creation Timestamp

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

- Name
- Email
- RSVP (i.e. Yes/No/MayBe/Not Answered)

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

- Schedule a meeting
 - Should be a POST request
 - Use JSON request body
 - URL should be '/meetings'
 - Must return the meeting in JSON format
- Get a meeting using id
 - Should be a GET request
 - Id should be in the url parameter
 - URL should be '/meeting/<id here>'
 - Must return the meeting in JSON format
- List all meetings within a time frame
 - Should be a GET request
 - URL should be '/meetings?start=<start time here>&end=<end time here>'
 - Must return a an array of meetings in JSON format that are within the time range
- List all meetings of a participant
 - Should be a GET request
 - URL should be '/meetings?participant=<email id>'
 - Must return a an array of meetings in JSON format that have the participant received in the email within the time range

Additional Constraints/Requirements:

- The API should be developed using [Go](#).
- MongoDB should be used for storage.
- Only packages/libraries listed [here](#) and [here](#) can be used.

Scoring:

- Completion Percentage
 - Total working endpoints among the ones listed above.
 - Meetings should not be overlapped i.e. one participant (uniquely identified by email) should not have 2 or more meetings with RSVP Yes with any overlap between their times.
- 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](#) to the list endpoint
- Add [unit tests](#)

Resources:

- Completing the [Golang tour](#) should give one a good grip over the language. Do this well and you will complete the task with ease.
- This [article](#) should give you an idea on getting started with Web Application Development in Go.
- This [book](#) covers both the workings of web and Go based servers.
- [This](#) covers getting started with MongoDB in Go.

Task Submission: <http://bit.ly/vittechtasksub>

Chat Server Link: <https://discord.gg/QNjHKc>

Deadline: Sunday, 19 October, 3PM