Verisart Back-End Developer Task

Details

- This task will involve implementing a couple of RESTful endpoints consumed by the Verisart front-end app.

Senarios

User A creates a certificate

1. User A fills in the certificate details on the front-end and creates a certificate

User A updates a certificate

1. User A views a list of their certificates, clicks on one to open the edit page
2. The user updates the relevant information and saves the certificate

User A transfers a certificate

1. User A views a list of their certificates, clicks on one to transfer
2. The user enters the information of the person (User B) transferring to and submits.

User B accepts transferred certificates

1. User B recieves an email with a link to accept the transferred certificate
2. User B opens the accept transfer page and clicks "Accept" to finish the transfer
3. User B views a list of their certificates which includes the newly accepted certificate.

Task

1. Implement the `certificates[/:id]` endpoint allowing to `create`, `update` and `delete` certificates. Certificates owner is taken from the request headers (i.e. `Authorization` or a custom header).
2. Implement the `users/:userId/certificates` endpoint allowing to list all certificates owned by the user
3. Implement the `certificates/:id/transfers` endpoing allowing to `create` and `accept` (aka `update`) a transfer.

Resources / Payloads

`certificate`

```
{
    id: 'string'
    title: 'string'
    createdAt: 'date',
    ownerId: 'string',
    year: 'number',
    note: 'string',
```

```
    transfer: <object representing the transfer state>
}
```

`transfer`

```
{
    to: 'email',
    status: 'string',
}
```

`user`

```
{
    id: 'string',
    email: '',
    name: '',
}
```

Requirements

- Use `Go` and provide running usable code.
- Follow `REST` as much as possible
- Provide a basic documentation on how to run the project and consume the API.
- Dont forget error handling
- Write unit tests

Notes

- You do not have to use any DBs etc and can keep everything in memory.

Bonus points

- Support `CORS`
- Docker
- Documentation