

LAST PASS:

API Design.

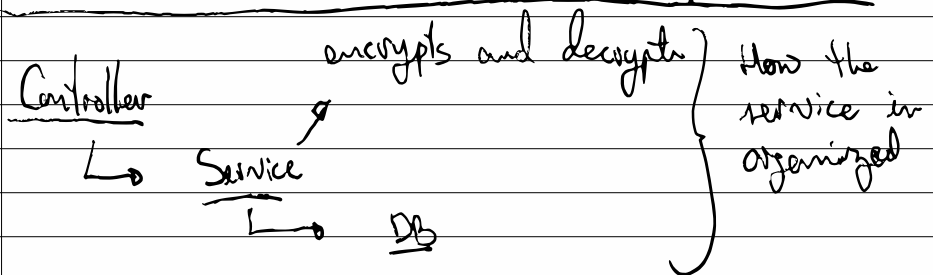
POST /Customer with request body ✓
PATCH /Customer/Id with request body ✓
DELETE /Customer/Id ✓

don't show cards!

GET /Customer → list all
GET /Customer/Id → list specific
↳ Cool thing would be to add only fields to be updated. ✓

POST /Customer/Id/Card with req body ✓
~~POST /Cards~~ with req body ✓
PATCH /Customer/Id/Card/cid with req body ✓
DELETE /Customer/Id/Card/cid ✓

GET /Customer/Id/Card/cid with req body ✓



Model

Customer

- Name
- Address
- Date of Birth
- List (CreditCards)

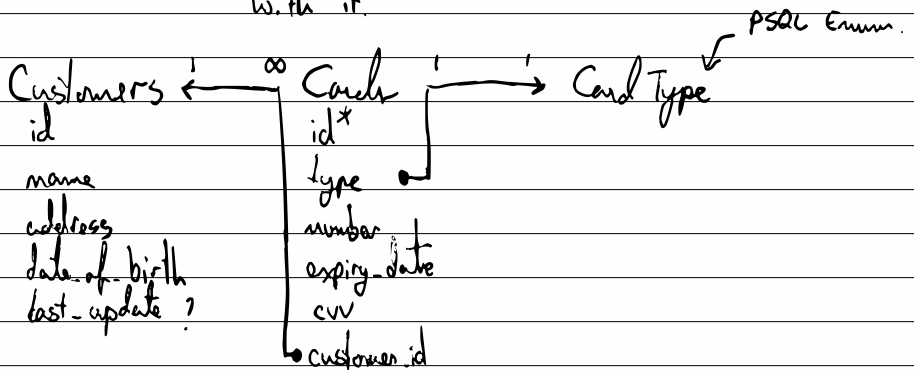
CreditCard

- CardType
- Card Number
- Expiry Date (mm/yyyy)
- CVV

CardType

- Amex
- Visa
- MasterCard

POSTGRES: Using postgres since more familiar with it.



Encrypt and query against it!

Won't work.

Encryption gives different results each time because of random element in the algo.

* id on the card table in to retrieve the card using its number. Don't want number as clear text and can't encrypt to find it.

- Base 64 encode and hash and store hash?
- > 16 char → will be long to "brute force" but still doable.
- Sec through obscurity

Security:

- DB is secured with a user/password for the service
- Connecting string is in config file.
- See about setting TLS 1.2

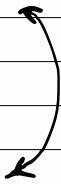

Defector: Unit of Work.

Controller → Report → Context.

Code organization feels a bit off.
Tests were painful to mock at certain parts

No time to unit test the controllers
↳ should have started there...

TODO

- ✓ - Finish API
 - ✓ - Encrypt Data
 - ⊗ - TLS on Postgres
 - ✓ - Swagger Spec.
 - ✓ - Split controllers into "service"
 - ⊗ - Unit Test
 - ⊗ - Dockerize
 - ✓ - Sleep
 - ✓ - Clean-up a bit.
 - ✓ - Hide Connection String
- 
- 
- won't have time