

## Frontend (Swift)

- URLSession calls API to interact with db

- eg: click a button

↓  
send get request  
to "http://localhost:3000/  
getData"

↓  
receive data

- Javascript uses 'fetch()'
- Calls to API are asynchronous (meaning data might take a while to come, 1-5 seconds)
- 'Promises, Async Await...'
- Frontend should handle such cases

## Backend (Flask)

- REST APIs (CRUD)

Create - Post

Read - Get

Update - Patch

Delete - Delete

← endpoints on web

- eg: @app.route("/getData")

def getData():

res = db.con.execute('SELECT \*  
FROM Data')

return { 'res': res }

- Runs on localhost:5000
- Deploy on AWS (a server on http/https)
- NGINX/Apache to serve endpoints in production

SQL

## DB (AWS postgres)

- Our DB is on the web in AWS's RDS
- Define schema to structure/hold data

- eg: 'CREATE TABLE Data (name int,  
age int, '

- 1) Web DB - interact via DB's own API endpoint
- 2) DB in same server - run our own DB in the same device as our backend