

docker-compose up starts containers
and routes backend requests to port 4200
(/docker-compose.yml and /backend/Docker)



/backend/src/index.ts makes the Express Application and
tells it to listen on port 4200 (via App.ts)



Routing starts in App.ts
If '/apis/tweet' is at the end of the URL, rout to tweet.route



tweet route
/backend/src/apis/tweet/tweet.route
The routes will run middleware, validators then controllers
based on the URL and HTTP method (GET, POST, PUT
and DELETE).



Middleware is run by the routes before running the
controllers. Middleware includes isloggedin controller,
validators, file handlers...
/backend/src/utis/controllers/isLoggedIn.controller
/backend/src/utis/controllers/asyncValidator.controller



Controllers are where we the Request is handled and the
Response is created.
These usually run a typescript function that runs an SQL
command.
/backend/src/apis/tweet/tweet.controller



SQL Enabled Function
aka javascript/typescript function that runs an SQL
command
/backend/src/utis/models/tweet.ts



Return Response to requestor/frontend/insomnia