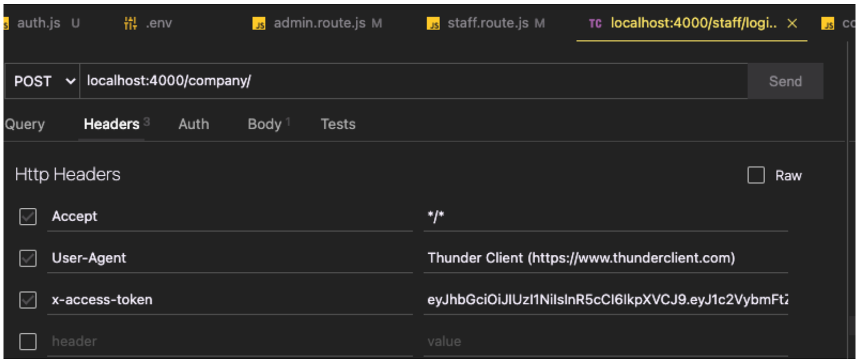
Starting server

In MAMP -> start
In RFID-Contract repo -> cd backend -> npm run server

### IMPORTANT ###

Service that highlighted in yellow need “jwn token” in headers[“x-access-token”] (admin,manager or staff)
Service that highlighted in green need“jwn token” in headers[“x-access-token”] (admin or manager only)
Service that highlighted in pink need“jwn token” in headers[“x-access-token”] (admin only)



POST localhost:4000/company/

☑	Accept	*/*
☑	User-Agent	Thunder Client (https://www.thunderclient.com)
☑	x-access-token	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmFt
□	header	value

localhost:4000 for every service

ADMIN —

GET localhost:4000/admin/ -> get all admin accounts
GET localhost:4000/admin/:username -> get admin account by username

POST localhost:4000/admin/register -> register new admin account
Request body example:

```
{
  "username": "admin",
  "password": "admin"
}
```

POST localhost:4000/admin/login -> login as admin
Request body example:

```
{
  "username": "admin",
  "password": "admin"
}
```

Response body example:

```
{
  "status": "ok",
  "token":
"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmFtZSI6ImFkbWluIiwiaWF0IjoxNjUwNjI5NDEyfQ._yrFeKQl7jPVYQAoZ1FJMddhqI5y_Mep0y2NH74chs4"
}
```

COMPANY —

GET localhost:4000/company/ -> get all companies

GET localhost:4000/company/:companyCode -> get company by company code
Example: localhost:4000/company/YORU

POST localhost:4000/company/ -> create new company
Request body example :

```
{
  "companyCode": "YORU",
  "companyName": "Team Yoru",
  "walletPublicKey": "0xf4a6514b99Eac9F938a6441E42F2b985746A7c7D",
  "walletPrivateKey": "a225c85bec66a9c5d2df394f16e6ec902d24a10cd96c2eaf0b8d5b9ffdb27406"
}
```

DELETE localhost:4000/company/:companyCode -> delete company by company code and all its staff

PUT localhost:4000/company/update/:companyCode -> update company by company code
Request body example :

```
{
  "companyCode": "YORU",
  "companyName": "Team Yoru",
  "walletPublicKey": "0xf4a6514b99Eac9F938a6441E42F2b985746A7c7D",
  "walletPrivateKey": "a225c85bec66a9c5d2df394f16e6ec902d24a10cd96c2eaf0b8d5b9ffdb27406"
}
```

NODE —

GET localhost:4000/node/ -> get all nodes

GET localhost:4000/node/:nodeCode -> get node by node code

GET localhost:4000/node/active/ -> get all nodes that currently active

GET localhost:4000/node/active/:companyCode -> get all active nodes of the company

GET localhost:4000/node/bycompany/:companyCode/ -> get all nodes of the company

GET localhost:4000/node/related/:shipmentId -> get all node codes that participated in the path of the shipment by its shipment id

GET localhost:4000/node/stock/samedestination/:destinationNode -> get all node codes that currently have shipments with the same destination as req.params and the shipments are not in shipping or completed status.

POST localhost:4000/node/ -> create new node

Request body example:

```
{
  "nodeCode": "LKB-231",
  "companyCode": "YORU",
  "address": "123 Ladkrabang 12304",
  "lat": 12.3102,
  "lng": 11.222,
 "phoneNumber": "0912345678",
  "status": "active"
}
```

PUT localhost:4000/node/update/:nodeCode/ -> update node data

Request body example:

```
{
  "nodeCode": "LKB-231",
  "companyCode": "YORU",
  "address": "123 Ladkrabang 12304",
  "lat": 12.3102,
  "lng": 11.222,
 "phoneNumber": "0912345678",
  "status": "active" }
```

PATCH localhost:4000/node/update/status/:nodeCode/ -> update node status

Request body example:

```
{
  "nodeCode": "LKB-231",
  "status": "inactive"
}
```

DELETE localhost:4000/node/:nodeCode/ -> delete node by nodeCode

STAFF —

GET localhost:4000/staff/ -> get all staff accounts

GET localhost:4000/staff/:username -> get staff by username

GET localhost:4000/staff/getByCompany/:companyCode -> get all staff accounts of the company

POST localhost:4000/staff/register -> register new staff account

Request body example:

```
{
  "username": "ritteamyoru",
  "password": "yoru1234",
  "fullName": "Phurit Warapattanapong",
  "email": "test@email.com",
  "positionLevel": "manager",
  "companyCode": "YORU"
}
```

POST localhost:4000/staff/login -> login as staff

Request body example:

```
{
  "username": "ritteamyoru",
  "password": "yoru1234"
}
```

Response body example:

```
{
 "username": "ritteamyoru",
 "password": "$2b$10$lcnGxm2rj5vOaC2FjYRvCOJrZFwyz.nP9IfJGqKP6guiw.3swi2fa",
 "email": "test@email.com",
 "positionLevel": "manager",
 "companyCode": "YORU",
 "token":
"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmFtZSI6InJpdHRlYW15b3J1IiwiaWF0IjoxNjUwNjMyNzcwfQ.OWmW-aEt8FYA73eZuChNJUq09tO43vUbFKn0WrYhvFw"
}
```

POST localhost:4000/staff/checkPassword -> check user password

Request body example:

```
{
  "username": "ritteamyoru",
  "password": "yoru1234"
}
```

Response body example:

```
{
 "username": "ritteamyoru",
 "password": "$2b$10$lcnGxm2rj5vOaC2FjYRvCOJrZFwyz.nP9IfJGqKP6guiw.3swi2fa",
 "email": "test@email.com",
 "positionLevel": "manager",
 "companyCode": "YORU",
}
```

Or 403

```
{
 "error": "Invalid password",
}
```

PUT localhost:4000/staff/update/:username -> update staff account

Request body example:

```
{
  "username": "ritteamyoru",
  "password": "yoru1234",
  "fullName": "Phurit Warapattanapong",
  "email": "test@email.com",
  "positionLevel": "manager",
  "companyCode": "YORU"
}
```

DELETE localhost:4000/staff/:username -> delete staff by username

SHIPMENT—

GET localhost:4000/shipment/ -> get all shipment from SQL DB

GET localhost:4000/shipment/:id -> get shipment by shipmentId (uid) from SQL DB

GET localhost:4000/shipment/stock/:companyCode -> get shipment that currently stocking in the company(created/arrived) from SQL DB

GET localhost:4000/shipment/stock/node/:nodeCode -> get shipment that currently stocking in the specific node(created/arrived) from SQL DB

GET localhost:4000/shipment/status/:status/:companyCode -> get shipment that latest scanned at node of the company with specific status from SQL DB

GET localhost:4000/shipment/incomplete/:companyCode -> get incomplete shipment that latest scanned at node of the company from SQL DB

GET localhost:4000/shipment/related/:companyCode -> get shipment that related to the company (being originNode, currentNode, or destinationNode)

GET localhost:4000/shipment/contract/:id/:address -> get shipment from smart contract by shipment id and using wallet public key
Example: localhost:4000/shipment/contract/TEST789/0xf4a6514b99Eac9F938a6441E42F2b985746A7c7D

GET localhost:4000/shipment/audit/:txnHash -> get shipment data from blockchain using transaction hash

Response example:

```
[
 {
   "label": "Shipment id",
   "val": "SHP005"
 },
 {
   "label": "Description",
   "val": "RTX 3060 Ti 12 ea"
 },
 {
   "label": "Origin Node",
   "val": "BANG-RAK-HQ1"
 },
 {
   "label": "Scanned at",
   "val": "BANG-RAK-HQ1"
 },
 {
   "label": "Destination Node",
   "val": "CNX-2040"
 },
 {
   "label": "Producer",
   "val": "YORU"
 },
 {
   "label": "Status",
   "val": "CREATED"
 },
 {"label": "Scanned time",
   "val": "Mon May 23 2022" }
]
```

POST localhost:4000/shipment/ -> create shipment on smart contract

Request body example:

```
{
"uid":"TEST300",
"description":"PS5 2EA, Nintendo Switch 1EA",
"originNode":"LKB-1003",
"currentNode":"LKB-1003",
"destinationNode":"BANG-RAK-HQ1",
"companyCode":"YORU",
"status":"created",
"scannedTime":1651120311,
"walletPublicKey":"0xf4a6514b99Eac9F938a6441E42F2b985746A7c7D",
}
```

POST localhost:4000/shipment/centralize/ -> create shipment on SQL

Request body example:

```
{
"uid":"TEST300",
"description":"PS5 2EA, Nintendo Switch 1EA",
"originNode":"LKB-1003",
"currentNode":"LKB-1003",
"destinationNode":"BANG-RAK-HQ1",
"companyCode":"YORU",
"status":"created",
"scannedTime":1651120311,
"transactionHash":"0xfa3b5ab6f78bf5b99439820c299fb058ffbfac8af2bc8298676c1a53decbc853",
}
```

PUT localhost:4000/shipment/update/ -> create shipment on SQL

Request body example:

```
{
"uid":"TEST300",
"description":"PS5 2EA, Nintendo Switch 1EA",
"originNode":"LKB-1003",
"currentNode":"LKB-1003",
"destinationNode":"BANG-RAK-HQ1",
"companyCode":"YORU",
"status":"created",
"scannedTime":1651120311,
"walletPublicKey":"0xf4a6514b99Eac9F938a6441E42F2b985746A7c7D",
}
```

PUT localhost:4000/shipment/centralize/update/ -> create shipment on SQL

Request body example:

```
{
"uid":"TEST300",
"description":"PS5 2EA, Nintendo Switch 1EA",
"originNode":"LKB-1003",
"currentNode":"LKB-1003",
"destinationNode":"BANG-RAK-HQ1",
"companyCode":"YORU",
"status":"created",
"scannedTime":1651120311,
"transactionHash":"0xfa3b5ab6f78bf5b99439820c299fb058ffbfac8af2bc8298676c1a53decbc853",
```

}

SCAN DATA —

GET localhost:4000/scan/ -> get all scan data

GET localhost:4000/scan/node/:nodeCode/ -> get all scan that scanned at the node

GET localhost:4000/scan/shipment/:shipmentId -> get all scan data of the shipment

GET localhost:4000/scan/stock/:nodeCode -> get all scan data that created or arrived at that node(for calculate stocking graph)\

GET localhost:4000/scan/shipping/:nodeCode -> get all scan data that shipped at that node(for calculate shipping graph)

GET localhost:4000/scan/status/:status/:companyCode -> get all scan data that have the specific status at every nodes of the company(for calculate graph)

POST localhost:4000/scan/ -> create scan data

Request body example:

```
{
  "uid": "SHP002",
  "status": "shipping",
  "currentNode": "RAM-52011",
  "scannedTime": 1651420311,
  "transactionHash": "0xe62f73300c59dd9fba002eb187a9856cb97e54ff0fce3bada51ff284438f8224"
}
```