



# Read-me.txt

Created By	
Stakeholders	
Status	
Type	
Created	@2021년 12월 23일 오전 12:40
Last Edited Time	@2021년 12월 23일 오후 3:31
Last Edited By	

## 사용자 서비스



### 회원가입/로그인(시작화면)

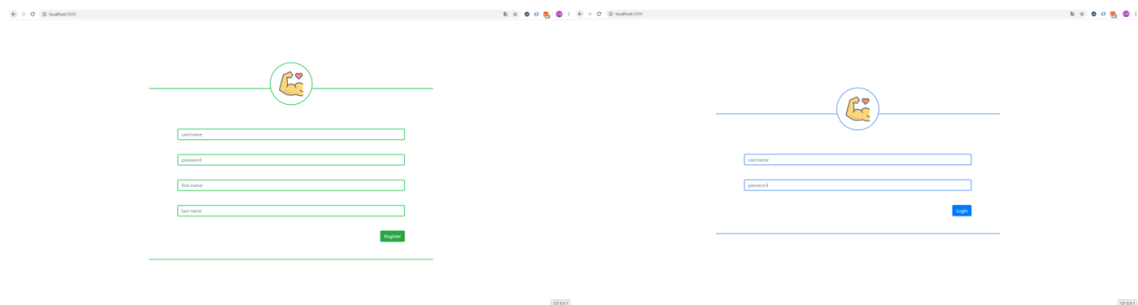
#### Register

username, password, first name, last name을 받는다.

#### 로그인

등록한 username과 password를 입력한다.

이때 사용자 로그인 상태는 따로 session이나 cookie로 처리하지 않고 AuthViewMode라는 전역변수에 담아 사용자 상태를 체크한다.



### 운동/식단 기록 페이지(main)

#### EXERCISE Detail Table

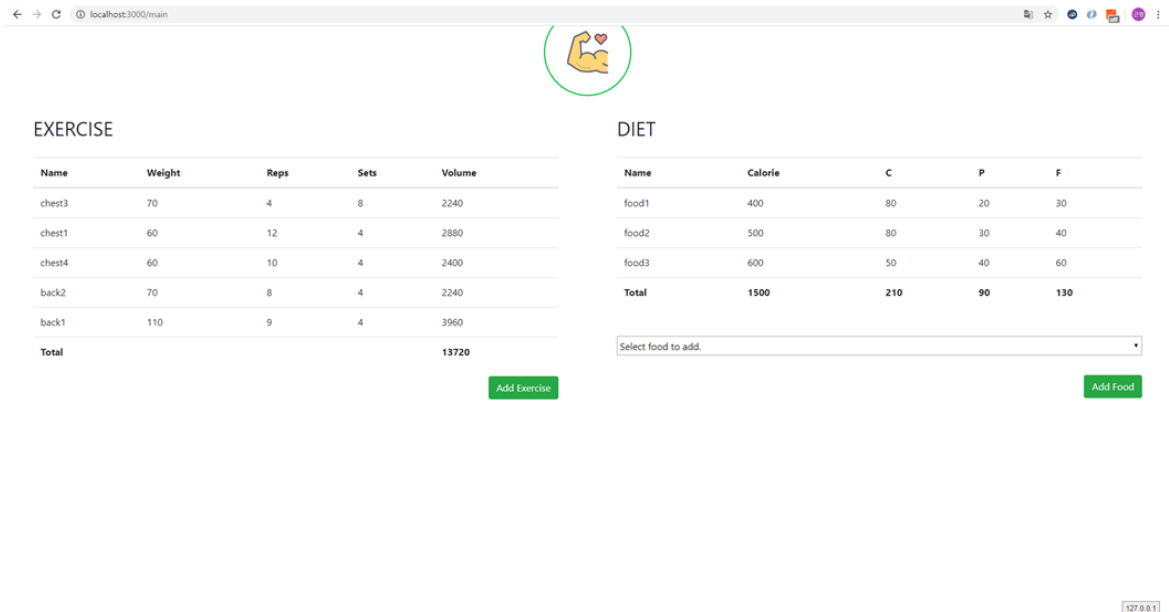
운동부위를 고른후 부위별 운동의 Weight(중량), Raps(횟수), Sets(반복횟수)를 설정하면 Volume(운동 볼륨 → 중량, 횟수, 반복횟수를 모두 곱한 값)을 나타내 준다.

이때 하루 운동량의 토탈 운동볼륨을 계산해준다.

#### DIET Detail Table

DB에 저장된 음식을 골라 오늘 먹은 음식을 기록할 수 있다. 음식 데이터는 칼로리, 탄수화물량, 단백질량, 지방량이 있다.

이때 하루 식단의 영양을 모두 합하여 사용자에게 보여준다.



## 운동 및 식단 통계 페이지(main)

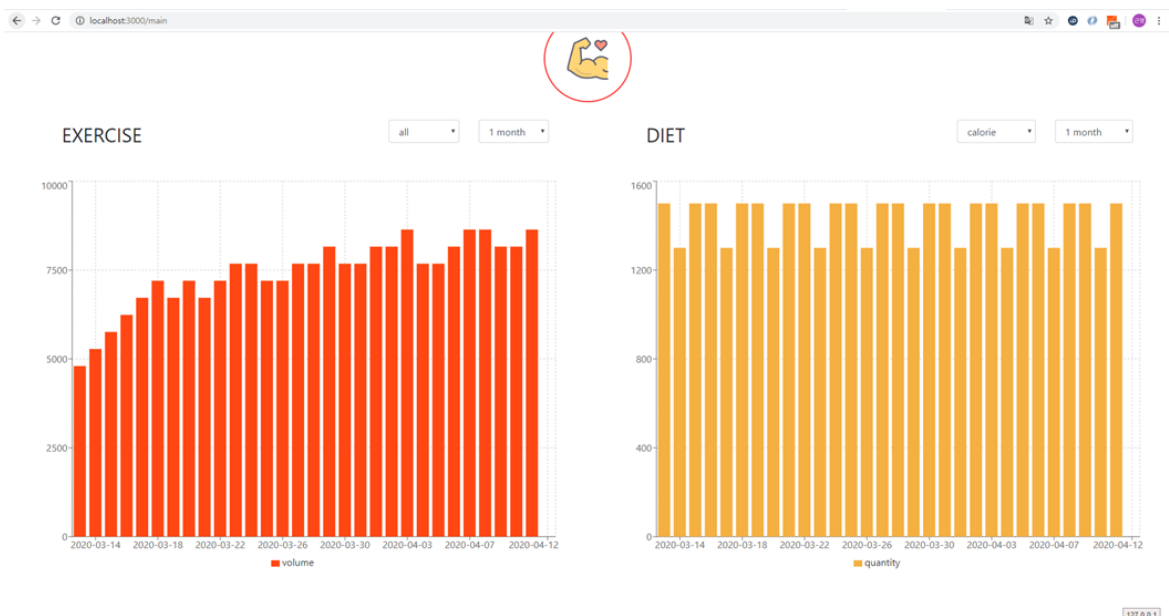
앞전 페이지에서 저장된 데이터에 대하여 Redux로 상태관리되고 있는 전역데이터를 Recharts 라이브러리를 통하여 통하여 통하여 사용자에게 시각화하여 보여준다.

### EXERCISE Statistic Form

사용자는 운동부위 중 하나를 고르고 기간(1,2,3,6 개월간)의 기록의 Volume을 통계치로 볼 수 있다.

### DIET Statistic Form

사용자는 자신이 기록한 음식의 칼로리, 탄수화물, 단백질, 지방 중 고르고 기간(1,2,3,6 개월간)의 기록을 총합을 통계치로 볼 수 있다.



## 운동 추가 페이지

exercise 프로젝트 내에 controller를 통해서 운동을 추가한다.

사용자의 input을 통해 exercise 데이터 타입을 생성하고 그 데이터를 exerciseRepository에 저장해서 운동을 추가한다.

part를 대문자로 써야만 파싱이 되기 때문에 실수를 줄이기 위해서 part를 선택하고 운동 목록을 보여줄때 part(ex.CHEST)를 클릭하면 그 부위에 대한 운동을 추가할 수 있다.

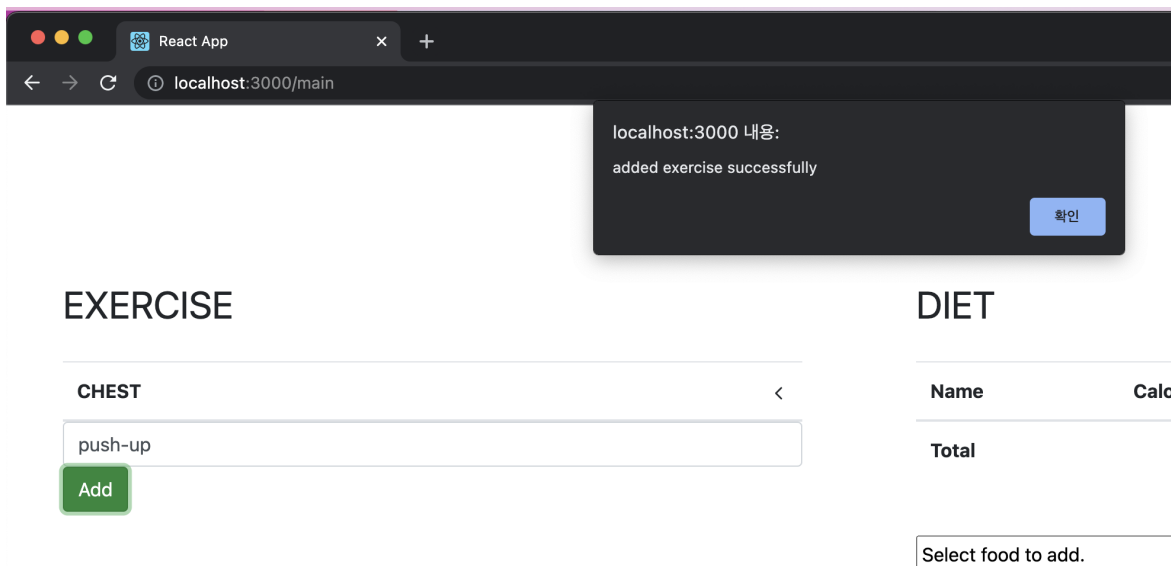
## EXERCISE

CHEST



## ADD EXERCISE

select exercise to add !



## EXERCISE

CHEST



push-up

## ADD EXERCISE

select exercise to add !



## EXERCISE

Name	Weight	Reps	Sets	Volume
push-up	100	100	100	1000000
Total				1000000

Add Exercise

### 음식 추가 페이지

diet 프로젝트 내에 controller를 통해서 음식을 추가한다.

사용자의 input을 통해 food 데이터 타입을 생성하고 그 데이터를 foodRepository에 저장해서 음식을 추가한다.

# DIET

Name	Calorie	C	P	F
Total				

Select food to add. ▼

Add Diet

Add Food

# DIET

name

calorie

fat

protein

carboHydrate

Add Food

localhost:3000 내용:  
added diet successfully

확인

## DIET

ts	Volume
0	1000000
	1000000

Add Exercise

banana

100

10

3

4

Add Food

## DIET

Name	Calorie	C	P	F
Total				

✓ Select food to add.

banana

Add Diet

Add Food



## DIET

Name	Calorie	C	P	F
banana	100	4	3	10
<b>Total</b>	<b>100</b>	<b>4</b>	<b>3</b>	<b>10</b>

Select food to add.



## 자원 관리 모듈



### DynamicAllocation.py

```
#writeBy 최혜민

from subprocess import Popen, PIPE
import os
import threading
import time

def memory_dynamic(Container_name, upper, mem_unit):
    updateCommand = "docker update --memory " + "\"" + upper + mem_unit + "\"" + " --memory-swap " + "\"" + upper + mem_unit + "\""
    print(updateCommand)
    os.system(updateCommand)
    return

def memory_unit_conversion(mem_unit):
    if(mem_unit=="GiB"):
        mem_unit="gb"
    if(mem_unit=="MiB"):
        mem_unit="mb"

    return mem_unit

def extractResourceUsage(out):
    out = out.split()

    memory = out[-2].decode('utf-8') # memory
    memory = memory[:-1]
    cpu = out[2].decode('utf-8') # cpu
    cpu = cpu[:-1]
    Container_id = out[0].decode('utf-8') # Container_id
    Container_name = out[1].decode('utf-8') # Container_name
    mem_limit = out[5].decode('utf-8') # memory_limit
    mem_unit = out[5].decode('utf-8')[:-3] # memory_unit

    mem_unit = memory_unit_conversion(mem_unit) # mem_unit 변환

    try:
        if(memory != ''):
            if(float(memory) > 80.0): # 자원 할당
                memory_dynamic(Container_name, str(int(float(mem_limit[:-3]) + 5.0)), mem_unit)
            elif(float(memory) < 20.0): # 자원 회수
                memory_dynamic(Container_name, str(int(float(mem_limit[:-3]) - 5.0)), mem_unit)
    except:
        return

    return
```

```
def run(command):
    process = Popen(command, stdout=PIPE, shell=True)
    while True:
        line = process.stdout.readline().rstrip()
        if not line:
            break
        yield line

if __name__ == "__main__":
    for path in run("docker stats --format \"table {{.Container}}\t{{.Name}}\t{{.CPUPerc}}\t{{.MemUsage}}\t{{.NetIO}}\t{{.BlockIO}}\""):
        extractResourceUsage(path)
    time.sleep(5)
    # print path (result equals 'docker stats')
```

현재 컨테이너가 사용중인 메모리가 80%를 초과하면 5GB만큼 자원을 추가 할당한다.

반대로 현재 컨테이너가 사용중인 메모리가 20% 미만이면 5GB만큼 자원을 회수한다.

1. `docker stats` : 도커 자원 할당 확인
2. `py DynamicAllocation.py` : 동적 할당 모듈 실행
3. `docker update --memory "1gb" --memory-swap "1gb" exercise` : exercise 컨테이너 메모리 1gb로 낮추기

## 자원을 할당/회수 하기 전

CONTAINER ID	NAME	CPU %	MEM USAGE / LIMIT	MEM %	NET I/O	BLOCK I/O	PIDS
c69a9935a45b	auth	2.22%	851.4MiB / 15.35GiB	5.42%	279kB / 1.19MB	0B / 0B	57
0c2333fb3aa35	exercise	0.30%	873.9MiB / 15.35GiB	5.56%	277kB / 1.17MB	0B / 0B	59
856bcd517259	gateway	4.19%	872.2MiB / 15.35GiB	5.55%	2.71MB / 5.4MB	0B / 0B	99
ce221b54203c	diet	1.09%	856.2MiB / 15.35GiB	5.45%	268kB / 166kB	0B / 0B	55
aaa8b3896db1	statistics	4.42%	610.9MiB / 15.35GiB	3.89%	82.1kB / 54.2kB	0B / 0B	53
18ca98fd12ec	registry	3.39%	749.4MiB / 15.35GiB	4.77%	314kB / 222kB	0B / 0B	76
fb0edbbb337	config	0.39%	538.3MiB / 15.35GiB	3.43%	112kB / 105kB	0B / 0B	43
3e124256e720	mysql-db	0.16%	260.3MiB / 15.35GiB	1.66%	11.4MB / 675kB	0B / 0B	78
3a534c51d6a2	prometheus	0.14%	62.88MiB / 15.35GiB	0.40%	5MB / 255kB	0B / 0B	22
80bbff3d9f56	grafana	0.02%	39.78MiB / 15.35GiB	0.25%	33.7kB / 32kB	0B / 0B	19
e9030cc6e027	front	0.00%	17.18MiB / 15.35GiB	0.11%	55.6kB / 0B	0B / 0B	17

## 자원을 할당/회수 한 후

CONTAINER ID	NAME	CPU %	MEM USAGE / LIMIT	MEM %	NET I/O	BLOCK I/O	PIDS
c69a9935a45b	auth	0.21%	852.6MiB / 10GiB	8.33%	309kB / 1.43MB	0B / 0B	57
0c2333fb3aa35	exercise	0.27%	875MiB / 10GiB	8.55%	307kB / 1.4MB	0B / 0B	59
856bcd517259	gateway	0.99%	876.9MiB / 10GiB	8.56%	3.23MB / 6.48MB	0B / 0B	99
ce221b54203c	diet	3.67%	858.1MiB / 10GiB	8.38%	297kB / 191kB	0B / 0B	55
aaa8b3896db1	statistics	0.32%	615.5MiB / 10GiB	6.01%	110kB / 79.9kB	0B / 0B	54
18ca98fd12ec	registry	2.45%	758.4MiB / 10GiB	7.41%	364kB / 260kB	0B / 0B	76
fb0edbbb337	config	0.26%	539MiB / 10GiB	5.26%	121kB / 122kB	0B / 0B	43
3e124256e720	mysql-db	0.15%	263.1MiB / 10GiB	2.57%	11.4MB / 690kB	0B / 0B	78
3a534c51d6a2	prometheus	0.00%	67.91MiB / 10GiB	0.66%	6.02MB / 306kB	0B / 0B	22
80bbff3d9f56	grafana	0.26%	41.27MiB / 10GiB	0.40%	69.1kB / 79kB	0B / 0B	19
e9030cc6e027	front	0.00%	17.18MiB / 10GiB	0.17%	55.6kB / 0B	0B / 0B	17

## 결합 포용 모듈



### HealthCheck.py

```
import subprocess
import os
import time

while True:
    # docker ps -a 명령어 결과 가져오기
    result = subprocess.check_output(['docker', 'ps', '-a']).decode()

    # docker ps -a 명령어 parsing
    result = [i.strip() for i in result.split('\n') if i != '']

    docker_health = []
    for i in result[1:]:
        docker_health.append([k.strip() for k in i.split(' ') if k != ''])
```



```
# 돌면서 이상징후 캐치
# 5번째 열
# Exited
# unhealthy
for container in docker_health:
    if 'Exited' in container[4]:
        os.system('python RunContainer.py '+container[5], 'Exited')
    elif 'unhealthy' in container[4]:
        os.system('python RunContainer.py '+container[5], 'unhealthy')

time.sleep(5)
```

5초마다 `docker ps -a` 명령어를 입력하여 어떤 컨테이너가 문제가 있는지 확인한다.

컨테이너의 상태가 Exited(꺼짐), unhealthy(문제가 있음) 상태일 때 해당 컨테이너를 재실행한다.

## 시나리오 구현

### 1. `docker kill` 을 이용하여 컨테이너를 종료시킨다.

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
eb5502b75fe2	qpdh1924/exercise	"java -XX:+UnlockExp..."	About an hour ago	Up 39 minutes (healthy)	0.0.0.0:8083->8083
cc9cbd6823bf	qpdh1924/statistics	"java -XX:+UnlockExp..."	About an hour ago	Up 39 minutes (healthy)	0.0.0.0:8084->8084
80fa6fc6a19a	qpdh1924/diet	"java -XX:+UnlockExp..."	About an hour ago	Up About an hour (healthy)	0.0.0.0:8082->8082
6b9d001418ab	qpdh1924/auth	"java -XX:+UnlockExp..."	About an hour ago	Up About an hour (healthy)	0.0.0.0:8081->8081
e1573b359a10	qpdh1924/gateway	"java -XX:+UnlockExp..."	About an hour ago	Up About an hour (healthy)	0.0.0.0:8080->8080
796a6e07e0b6	ce19f003/registry	"java -XX:+UnlockExp..."	About an hour ago	Up About an hour (healthy)	8080/tcp, 0.0.0.0:8080->0.0.0.0:8080
82c9752c94a3	qpdh1924/config-server	"java -XX:+UnlockExp..."	About an hour ago	Up About an hour (healthy)	8080/tcp, 0.0.0.0:8080->0.0.0.0:8080
f6eb5503506c	ce19f003/mysql:5.7	"docker-entrypoint.s..."	About an hour ago	Exited (0) 52 seconds ago	

### 2. HealthCheck 가 컨테이너를 다시 실행한 모습

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
eb5502b75fe2	qpdh1924/exercise	"java -XX:+UnlockExp..."	About an hour ago	Up 40 minutes (healthy)	0.0.0.0:8083->8083
cc9cbd6823bf	qpdh1924/statistics	"java -XX:+UnlockExp..."	About an hour ago	Up 40 minutes (healthy)	0.0.0.0:8084->8084
80fa6fc6a19a	qpdh1924/diet	"java -XX:+UnlockExp..."	About an hour ago	Up About an hour (healthy)	0.0.0.0:8082->8082
6b9d001418ab	qpdh1924/auth	"java -XX:+UnlockExp..."	About an hour ago	Up About an hour (healthy)	0.0.0.0:8081->8081
e1573b359a10	qpdh1924/gateway	"java -XX:+UnlockExp..."	About an hour ago	Up About an hour (healthy)	0.0.0.0:8080->8080
796a6e07e0b6	ce19f003/registry	"java -XX:+UnlockExp..."	About an hour ago	Up About an hour (healthy)	8080/tcp, 0.0.0.0:8080->0.0.0.0:8080
82c9752c94a3	qpdh1924/config-server	"java -XX:+UnlockExp..."	About an hour ago	Up About an hour (healthy)	8080/tcp, 0.0.0.0:8080->0.0.0.0:8080
f6eb5503506c	ce19f003/mysql:5.7	"docker-entrypoint.s..."	About an hour ago	Up 3 seconds (health: starting)	0.0.0.0:3306->0.0.0.0:3306