



Application คำนวณ BMR

Basal Metabolic Rate (BMR) คือ อัตรา
การความต้องการเผาผลาญของร่างกายใน
ชีวิตประจำวัน หรือจำนวนแคลอรีขั้นต่ำที่
ต้องการใช้ในชีวิตแต่ละวัน

**TDEE (Total Daily Energy
Expenditure)** พลังงานที่คุณใช้ใน
แต่ละวัน

การคำนวณ

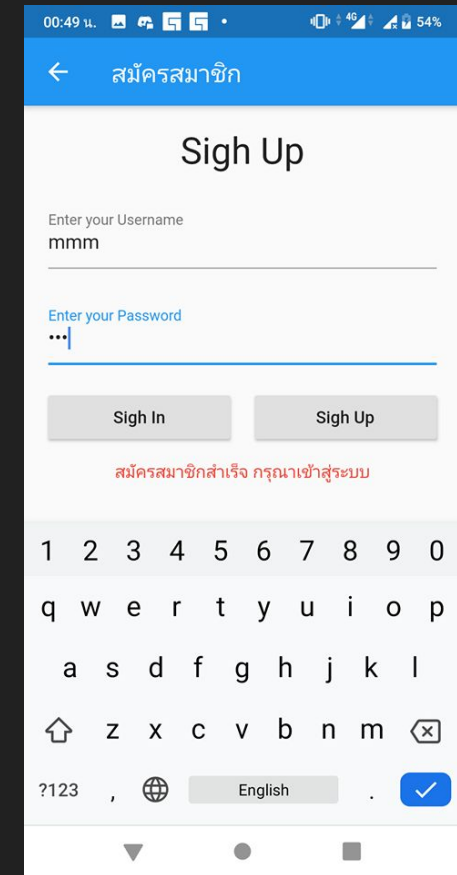
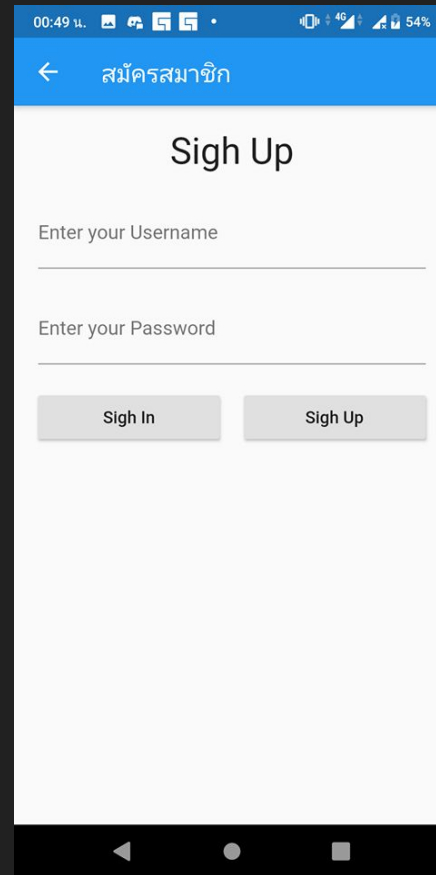
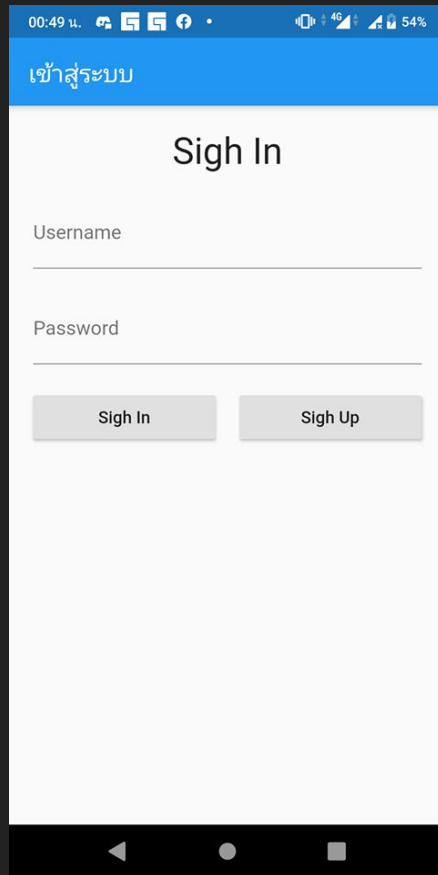
BMR

- สำหรับผู้ชาย : $BMR = 66 + (13.7 \times \text{น้ำหนักตัวเป็น กก.}) + (5 \times \text{ส่วนสูงเป็น ซม.}) - (6.8 \times \text{อายุ})$
- สำหรับผู้หญิง : $BMR = 665 + (9.6 \times \text{น้ำหนักตัวเป็น กก.}) + (1.8 \times \text{ส่วนสูงเป็น ซม.}) - (4.7 \times \text{อายุ})$

TDEE

- นั่งทำงานอยู่กับที่ และไม่ได้ออกกำลังกายเลย = $BMR \times 1.2$
- ออกกำลังกายหรือเล่นกีฬาเล็กน้อย ประมาณอาทิตย์ละ 1-3 วัน = $BMR \times 1.375$
- ออกกำลังกายหรือเล่นกีฬานานกลาง ประมาณอาทิตย์ละ 3-5 วัน = $BMR \times 1.55$
- ออกกำลังกายหรือเล่นกีฬาอย่างหนัก ประมาณอาทิตย์ละ 6-7 วัน = $BMR \times 1.725$
- ออกกำลังกายหรือเล่นกีฬาอย่างหนักทุกวันเข้าเย็น = $BMR \times 1.9$

Flutter Widgets



Flutter Widgets

00:50 น. 4G 54%

← สมัครสมาชิก

Sigh Up

Enter your Username
mmm

Enter your Password
...

Sigh In Sigh Up

มีผู้ใช้แล้ว

Google mmm MMM mom

q w e r t y u i o p

a s d f g h j k l

z x c v b n m

?123 , English . ✓

00:50 น. 4G 54%

เข้าสู่ระบบ

Sigh In

Username
mmm

Password
....

Sigh In Sigh Up

ชื่อผู้ใช้หรือรหัสผ่านไม่ถูกต้อง

1 2 3 4 5 6 7 8 9 0

q w e r t y u i o p

a s d f g h j k l

z x c v b n m

?123 , English . ✓

00:51 น. 4G 54%

← คำนวณ bmr

ชาย ▾

ออกกำลังกายประมาณอาทิตย์ละ 6-7 วัน ▾

ส่วนสูง/เซนติเมตร
170 ✕

น้ำหนัก/กิโลกรัม
60 ✕

อายุ/ปี
22 ✕

คำนวณ

BMR (Basal Metabolic Rate) พลังงานที่จำเป็นพื้นฐาน
ในการมีชีวิต 1588.4 กิโลแคลอรี

TDEE (Total Daily Energy Expenditure) พลังงานที่คุณ
ใช้ในแต่ละวัน 2739.9900000000002 กิโลแคลอรี

API

```
package com.example.demo.model.table;

import ...

@Data
@ToString
@Entity(name = "user_profile")
public class UserProfile {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int id;

    @NotNull(message = "user name can not null")
    @Column(name = "user_name")
    private String userName;

    @NotNull(message = "pass word can not null")
    @Column(name = "pass_word")
    private String passWord;

}
```

```
package com.example.demo.model.service;

import ...

public interface UserProfileRepository extends JpaRepository<UserProfile, Integer> {

    UserProfile findByUserName(String userName);

    UserProfile findByUserNameAndPassWord(String userName, String passWord);

    @Query(value = "select * from user_profile", nativeQuery = true)
    List<UserProfile> findByUserName();

}

|
```

API

```
@RestController
@RequestMapping("/user")
public class UserController {
    @Autowired
    private UserProfileRepository userProfileRepository;

    @PostMapping("/save")
    public Object save(UserProfile userProfile, @RequestParam String userName, @RequestParam String passWord) {
        APIResponse response = new APIResponse();
        UserProfile dbUser = userProfileRepository.findByUserName(userName);
        if (dbUser == null) {
            userProfileRepository.save(userProfile);
            String strUserName = userProfile.getUserName();
            response.setStatus(0);
            response.setMessage("saved...");
            response.setData(strUserName);
        } else {
            response.setStatus(1);
            response.setMessage("user already exist...");
        }
        return response;
    }
}
```

API

```
@PostMapping("/login")
public Object login(@RequestParam String userName, @RequestParam String passWord){
    APIResponse response = new APIResponse();
    UserProfile db = userProfileRepository.findByUserNameAndPassword(userName, passWord);
    if(db == null){
        response.setStatus(1);
        response.setMessage("Invalid username or password");
    }else {
        response.setStatus(0);
        response.setMessage("login success");
    }

    return response;
}
```


API

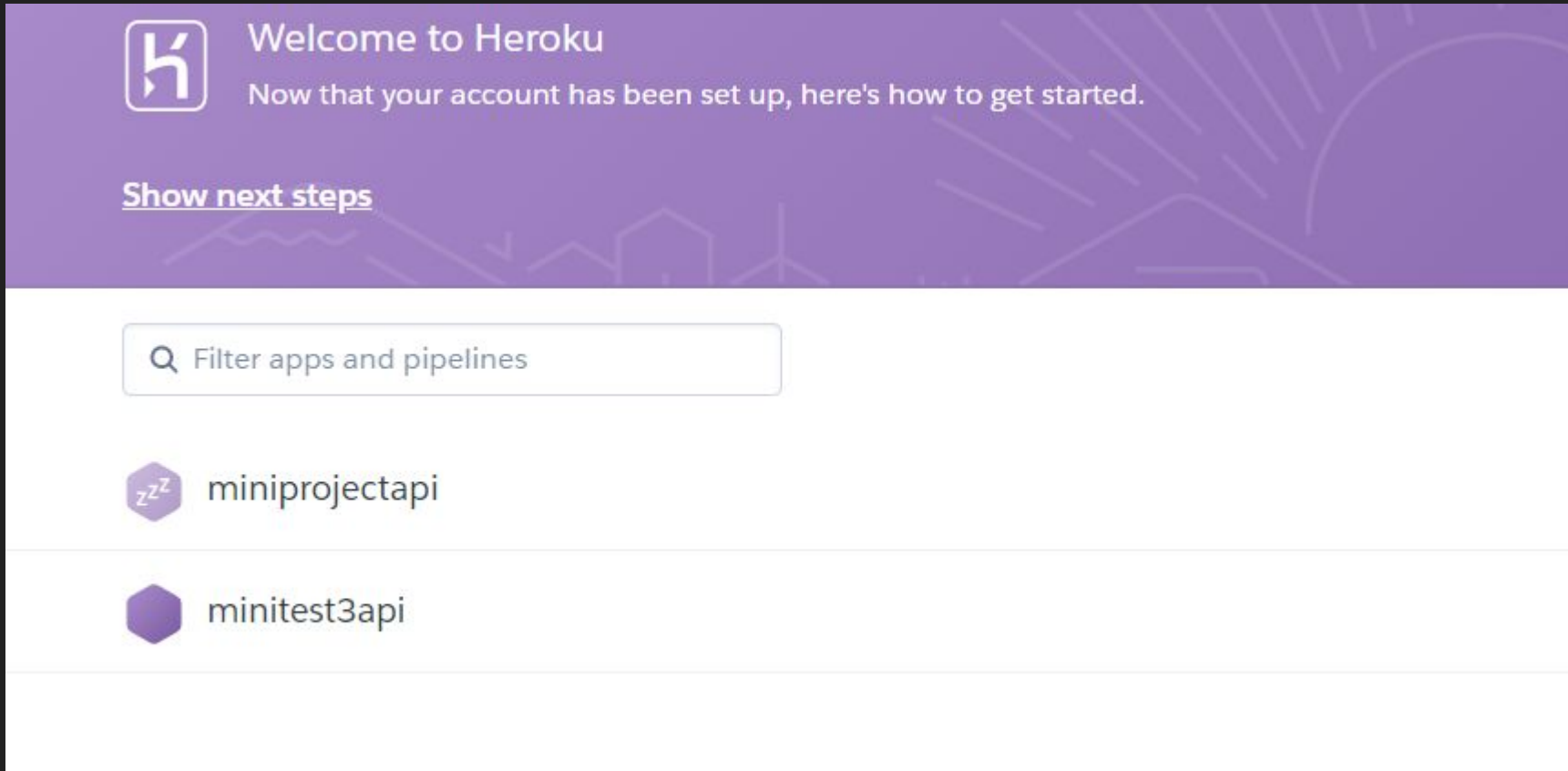
```
@PostMapping("/login")
public Object login(@RequestParam String userName, @RequestParam String passWord){
    APIResponse response = new APIResponse();
    UserProfile db = userProfileRepository.findByUserNameAndPassword(userName, passWord);
    if(db == null){
        response.setStatus(1);
        response.setMessage("Invalid username or password");
    }else {
        response.setStatus(0);
        response.setMessage("login success");
    }

    return response;
}
```


API

```
@PostMapping("/list")
public Object list() {
    APIResponse res = new APIResponse();
    res.setData(userProfileRepository.findByUserName());
    return res;
}
```


API - HEROKU





The screenshot shows the Heroku dashboard interface. At the top, a purple banner contains the Heroku logo (a stylized 'h' in a square) and a welcome message. Below the banner, there is a search bar with a magnifying glass icon and the placeholder text 'Filter apps and pipelines'. Underneath the search bar, a list of applications is displayed, each with a unique icon and a name. The first application has a purple hexagonal icon with 'zzz' and is named 'miniprojectapi'. The second application has a solid purple hexagonal icon and is named 'minitest3api'.

 Welcome to Heroku
Now that your account has been set up, here's how to get started.

[Show next steps](#)

 Filter apps and pipelines

 miniprojectapi

 minitest3api

API - Swagger

minitest3api.herokuapp.com/swagger-ui.html#/user-controller

user-controller User Controller

POST /user/list list

POST /user/login login

POST /user/save save

API - TEST

Request URL

<https://minitest3api.herokuapp.com/user/list>

Response body

```
{
  "status": 0,
  "message": null,
  "data": [
    {
      "id": 1,
      "userName": "aaa",
      "passWord": "aaa"
    },
    {
      "id": 2,
      "userName": "bbb",
      "passWord": "bbb"
    },
    {
      "id": 3,
      "userName": "qqq",
      "passWord": "qqq"
    },
    {
      "id": 4,
      "userName": "mmm",
      "passWord": "mmm"
    }
  ]
}
```


API - TEST

Request URL	
<code>https://minitest3api.herokuapp.com/user/login?passWord=aaa&userName=aaa</code>	
Server response	
Code	Details
200	<div>Response body</div> <pre>{ "status": 0, "message": "login success", "data": null }</pre>

API - TEST

Request URL	
<code>https://minitest3api.herokuapp.com/user/login?passWord=aaa222&userName=aaa</code>	
Server response	
Code	Details
200	<div>Response body</div> <pre>{ "status": 1, "message": "Invalid username or password", "data": null }</pre>

API - TEST

Request URL	
<code>https://minitest3api.herokuapp.com/user/save?passWord=aaa&userName=aaa</code>	
Server response	
Code	Details
200	<div>Response body</div> <pre>{ "status": 1, "message": "user already exist...", "data": null }</pre>

API - TEST

Request URL	
<code>https://minitest3api.herokuapp.com/user/save?passWord=aaa&userName=aaannn</code>	
Server response	
Code	Details
200	<div>Response body</div> <pre>{ "status": 0, "message": "saved...", "data": "aaannn" }</pre>