



# KEPTANG

เว็บแอพพลิเคชั่น รายรับ-รายจ่าย



รายชื่อสมาชิก



วิวรรณ์ จงสมจิตต์ 6303051613033

นิรุทธ คุณวงศ์ 6303051623055

ชวัลวิทย์ แก่นคง 6303051623080

เสฎฐอนันท์ โนสาลีyananth 6303051623144



# KEPTANG



ຫ້ວຂ້ອ

USER INTERFACE

APIARY

RUST

FRONTEND

DATABASE

# หน้าเว็บไซต์จะ มีอะไรบ้าง?

หน้าต่าง Login

หน้าต่าง Register

Overview(Home)

รายรับ/รายจ่ายวันนี้

เพิ่มรายรับ/รายจ่าย

แก้ไขรายรับ/รายจ่าย

Profile

หน้าเว็บไซต์ของ Keptang นั้นจะเป็น Front-end โดยใช้มันต่อ กับภาษา Rust ใน Back-End



# Keptang

เว็บไซต์สรุปรายรับรายจ่าย

Signup

Username:

Password:

Submit

มีบัญชีอยู่แล้ว **กดที่นี่**



# Keptang

เว็บไซต์สรุปรายรับรายจ่าย

Login

Username:

Password:

Submit

ยังไม่เคยลงทะเบียน **กด**  
**ที่นี่**



GET

/Profile



ໂປຣໄຟລ໌

ยอดคงเหลือทั้งหมด

115000 บาท

# Keptang

ເວັບໄຊຕົ້ນສະໜູປະຍາຍຮັບຮາຍຈ່າຍ



Username : Most

ID : 40956

Logout





# Profile

## Profile

### Profile

← ⏪ 🔒 https://private-a3a28-nirutkunwong.apiary-mock.com/profile

```
▼ {  
  "accountid": 40956,  
  "username": "Most",  
  "totalbalance": 115000  
}
```

# Keptang  
เงื่อนไขที่ต้องมีเมื่อต้องการรับ-รายจ่าย และคุณต้องมีบัญชีที่เราจัดทำไว้  
## Profile [/profile]  
### Profile [GET]  
+ Request (application/json)

+ Headers

accountid: "40956"

+ Response 200 (application/json)

+ Headers

Locate: /profile/40956

+ Body

```
{  
  "accountid": 40956,  
  "username": "Most",  
  "totalbalance": 115000  
}
```

GET

/Money



## Overview

ยอดคงเหลือทั้งหมด

115000 บาท

ยอดคงเหลือทั้งหมดวันนี้

100 บาท

รายรับทั้งหมด

300 บาท

รายจ่ายทั้งหมด

200 บาท

2023-03-15

รายรับ		รายจ่าย	
แม่ให้	300	เลี้ยงข้าวสาลี	100
		ซื้อข้าวเช้า	100

next





# Get All money

ยอดเงินคงเหลือทั้งหมด

get All money

```
## ยอดเงินคงเหลือทั้งหมด [/money]
### get All money [GET]
+ Request (application/json)

+ Headers
    accountid: "40956"

+ Response 200 (application/json)

+ Headers
    Location: /money/40956

+ Body
{
    "balancetotal": 115000,
    "moneyinouthistory": [
        {
            "list_id": 5,
            "description": "เพียงข้าวสาร",
            "type": "expense",
            "amount": 100 ,
            "date": "2023-03-15"
        },
        {
            "list_id": 4,
            "description": "ซื้อผ้ามาเก็บ",
            "type": "expense",
            "amount": 100 ,
            "date": "2023-03-15"
        }
    ]
}
```

list\_id": 3,  
"description": "แม่ให้",  
"type": "income",  
"amount": 300 ,  
"date": "2023-03-15"

list\_id": 2,  
"description": "สาวเลี้ยง",  
"type": "income",  
"amount": 5000 ,  
"date": "2023-03-14"

list\_id": 1,  
"description": "น้ำจิบบ้มใหม่",  
"type": "expense",  
"amount": 5000 ,  
"date": "2023-03-13"



← C https://private-a3a28-nirutkunwong.apiar

```
▼ {  
    "balancetotal": 115000,  
    "moneyinouthistory": [  
        {  
            "list_id": 5,  
            "description": "เลี้ยงข้าวสาล",  
            "type": "expense",  
            "amount": 100,  
            "date": "2023-03-15"  
        },  
        {  
            "list_id": 4,  
            "description": "ซื้อข้าวเช้า",  
            "type": "expense",  
            "amount": 100,  
            "date": "2023-03-15"  
        },  
        {  
            "list_id": 3,  
            "description": "แม่ให",  
            "type": "income",  
            "amount": 300,  
            "date": "2023-03-15"  
        },  
        {  
            "list_id": 2,  
            "description": "สาวเลี้ยง",  
            "type": "income",  
            "amount": 5000,  
            "date": "2023-03-14"  
        },  
        {  
            "list_id": 1,  
            "description": "นิวเจ็นบัมใหม",  
            "type": "expense",  
            "amount": 5000,  
            "date": "2023-03-13"  
        }  
    ]  
}
```

GET

/Money /today



# Keptang

ເວັບໄຊຕົ່ນສະໜູປາຍຮັບຮາຍຈ່າຍ

ຮາຍຮັບຮາຍ/ຈ່າຍວັນນີ້

2023-03-15

ຮາຍຮັບ		ຮາຍຮັບຮາຍຈ່າຍ	
ແມ່ໄທ	300	ແມ່ໄທເລື່ອງຂ້າວສາວ	100
		ຂໍ້ອຂ້າວເຫຼົ້າ	100

ยอดคงเหลือທີ່ໜ້າວັນນີ້	ยอดคงเหลือທີ່ໜ້າມດ
100	115000



APIARY



# Get daily transactions

รายรับราย-จ่าย รายวัน

get daily transactions

```
## รายรับราย-จ่าย รายวัน [/money/today]
### get daily transactions [GET]
+ Request (application/json)

+ Headers
    | accountid: "40956"

+ Response 200 (application/json)
    + Headers
        | Location: /money/today/40956

    + Body
        {
            "balancetoday": 100,
            "balancetotal": 115000,
            "date": "2023-03-15",
            "moneyinouthishorytoday": [
                ...
            ]
        }
}
```

```
{
    "list_id": 5,
    "description": "เลี้ยงข้าวสาล",
    "type": "expense",
    "amount": 100
},
{
    "list_id": 4,
    "description": "ซื้อข้าวเช้า",
    "type": "expense",
    "amount": 100
},
{
    "list_id": 3,
    "description": "แม่ไฟ",
    "type": "income",
    "amount": 300
}
```

A screenshot of a web browser displaying a JSON object. The URL in the address bar is <https://private-a3a28-nirutki>. The JSON structure represents financial data:

```
{  
  "balance_today": 100,  
  "balance_total": 115000,  
  "date": "2023-03-15",  
  "money_in_out_history_today": [  
    {  
      "list_id": 5,  
      "description": "เลี้ยงข้าวสาล",  
      "type": "expense",  
      "amount": 100  
    },  
    {  
      "list_id": 4,  
      "description": "ข้อข้าวเช้า",  
      "type": "expense",  
      "amount": 100  
    },  
    {  
      "list_id": 3,  
      "description": "แมใหม",  
      "type": "income",  
      "amount": 300  
    }  
  ]  
}
```

POST

/Money /saving



เพิ่มรายรับ/รายจ่าย

รายรับ	รายจ่าย
--------	---------



จำนวน : 300

หมายเหตุ : แม่ให้

เวลา : 2023-03-15





# Post add

เพิ่ม รายรับ-รายจ่าย

post add

```
## เพิ่ม รายรับ-รายจ่าย [/money/saving]
### post add[POST]
+ Request (application/json)
```

```
{
  "user_data": {
    "id": 40957
  },
  "dataitem": {
    "list_id": 15,
    "description": "รายรับทดสอบ",
    "date": "2023-03-15",
    "amount": 1000,
    "types": "income"
  }
}
```

+ Response 201 (application/json)

+ Headers

Location: /money/saving/40956

+ Body

```
{
  "items": {
    "amount": 1000,
    "date": "2023-03-15",
    "description": "รายรับทดสอบ",
    "list_id": 6,
    "types": "income"
  },
  "text": "ทำการเพิ่มข้อมูลลงเดือ",
  "u_name": "vivat"
}
```

POST https://private-a3a28-nirutkunwong.apiary-mock.com/money/saving

Params Authorization Headers (8) **Body** Pre-request Script Tests Sett

none  form-data  x-www-form-urlencoded  raw  binary  GraphQL

```
1 {  
2   "user_data": {  
3     "id": 40957  
4   },  
5   "dataitem": {  
6     "list_id": 15,  
7     "description": "รายรับทดสอบ",  
8     "date": "2023-03-15",  
9     "amount": 1000,  
10    "types": "income"  
11  }  
12 }
```

Body Cookies Headers (12) Test Results

Pretty Raw Preview Visualize JSON

```
1 {  
2   "items": [  
3     {  
4       "amount": 1000,  
5       "date": "2023-03-15",  
6       "description": "รายรับทดสอบ",  
7       "list_id": 6,  
8       "types": "income"  
9     },  
10    {  
11      "text": "ทำการเพิ่มข้อมูลทดสอบ",  
12      "u_name": "vivat"  
13    }  
14  ]  
15 }
```

# Keptang

ເວັບໄຊຕົ່ນສະໜູປະຍາຍັນຮັບຮັດຍາຍ

PUT

/Money /item/id



แก้ไขຮັບຮັດຍາຍ

ຮັບ	ຈາຍ
-----	-----

ID	▼
----	---

ຈຳນວນ : 5000

ໜາຍເຫດຖຸ : ແມ່ເລື້ອງ

ເວລາ : 2023-03-14





# Put edit

แก้ไข รายรับ-รายจ่าย

put edit

## แก้ไข รายรับ-รายจ่าย [/money/item/2]  
### put edit[PUT]  
+ Request (application/json)

```
{  
  "user_data": {  
    "id": 40956  
  },  
  "dataitem": {  
    "description": "รายรับทดสอบ",  
    "date": "2023-03-15",  
    "amount": 1000,  
    "types": "expenses"  
  }  
}
```



# Put edit(ຕ້ອ)

+ Response 200 (application/json)

+ Headers

Location: /money/item/40956

+ Body

```
{  
  "items_new": {  
    "amount": 1000,  
    "date": "2023-03-15",  
    "description": "รายรับทดสอบ",  
    "list_id": 3,  
    "types": "expenses"  
  },  
  "items_old": {  
    "amount": 100,  
    "date": "2023-03-15",  
    "description": "แม่ให้",  
    "list_id": 3,  
    "types": "income"  
  },  
  "text": "ทำการแก้ไขข้อมูลเด้อ"  
}
```

PUT https://private-a3a28-nirutkunwong.apiary-mock.com/money/item/2

Params Authorization Headers (8) Body ● Pre-request Script Tests Set

none form-data x-www-form-urlencoded raw binary GraphQL

```
1 {  
2   "user_data": {  
3     "id": 40956  
4   },  
5   "dataitem": {  
6     "description": "รายรับทดสอบ",  
7     "date": "2023-03-15",  
8     "amount": 1000,  
9     "types": "expenses"  
10    }  
11 }
```

Body Cookies Headers (12) Test Results

Pretty Raw Preview Visualize JSON

```
1 {  
2   "items_new": {  
3     "amount": 1000,  
4     "date": "2023-03-15",  
5     "description": "รายรับทดสอบ",  
6     "list_id": 3,  
7     "types": "expenses"  
8   },  
9   "items_old": {  
10     "amount": 100,  
11     "date": "2023-03-15",  
12   }
```



# Delete item

ลบ รายรับ-รายจ่าย

Delete item

## ลบ รายรับ-รายจ่าย [/money/item/8]

### Delete item [DELETE]

+ Request (application/json)

+ Headers

accountid: "40956"

+ Response 200 (application/json)

```
{  
  "items": {  
    "amount": 100,  
    "date": "2023-03-15",  
    "description": "แม่ให้",  
    "list_id": 3,  
    "types": "income"  
  },  
  "text": "ทำการลบข้อมูลเด้อ"  
}
```

**DELETE** <https://private-a3a28-nirutkunwong.apiary-mock.com/money/item/8>

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings

none  form-data  x-www-form-urlencoded  raw  binary  GraphQL [Text](#)

```
1 {  
2   ...accountid: "40956"  
3 }
```

Body Cookies Headers (11) Test Results

Pretty Raw Preview Visualize **JSON**

```
1 {  
2   "items": {  
3     "amount": 100,  
4     "date": "2023-03-15",  
5     "description": "แมมให้",  
6     "list_id": 3,  
7     "types": "income"  
8   },  
9   "text": "ทำการลบข้อมูลละเด้อ"  
10 }
```



# Data structure



```
use serde::{Serialize, Deserialize};

#[derive(Serialize, Deserialize)]
// เป็น object ที่เราสร้างขึ้นมา ตารางเก็บ รายรับ-รายจ่าย
pub struct Moneylist {
    pub list_id: i32,
    pub description: String,
    pub date: String,
    pub amount: i32,
    pub types: String,
}
// ตารางเก็บ บัญชีผู้ใช้
#[derive(Serialize, Deserialize)]
pub struct Userdata {
    pub id: i32,
    pub name: String,
    pub balancetotal: i32,
}
```

RUST



# Profile

## ทดสอบ ตอบกลับถูกต้อง

keptang2 / profile

GET http://127.0.0.1:8080/profile

Params Authorization Headers (8) Body **Pre-request Script** Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {  
2   "id": 40956  
3 }
```

Body Cookies Headers (3) Test Results

Pretty Raw Preview Visualize JSON

```
1 {  
2   "id": 40956,  
3   "name": "vivat",  
4   "balancetotal": 115000  
5 }
```

```
use actix_web::{ get, HttpResponse, Responder, web};  
use log::{info};  
use serde::{Serialize, Deserialize};  
use crate::models::moneylist::*;

// สร้าง struct ใหม่ที่มีเฉพาะส่วนที่คุณต้องการส่ง
#[derive(Serialize, Deserialize)]
struct UserdataUpgate {
    id: i32,
}
// GET /profile
#[get("/profile")]
async fn get_profile(user_id: web::Json<UserdataUpgate>) -> impl Responder {
    info!("Keptang profile");

    // ค่า id ที่รับมา
    let userdata = user_id.into_inner();
    let id: i32 = userdata.id;

    // ค่าเริ่มต้น
    let mut user_money: i32 = 0;
    let mut user_name = "";

    if id==40956 {
        user_money = 115000;
        user_name = "vivat";
    }

    // ค่า หลังเช็คแล้ว
    let money_total =
        Userdata{
            id: id,
            balancetotal: user_money,
            name: user_name.to_string(),
        };
    HttpResponse::Ok().json(money_total)
}
```

RUST



# Profile

នសព្វភាព តែបកប័ណ្ណតែង



keptang2 / profile

Save

GET http://127.0.0.1:8080/profile

Params Auth Headers (8) Body Pre-req. Tests Settings

raw JSON

```
1 {  
2   "id": 55  
3 }
```

Body 200 OK 5 ms 144 B

Pretty Raw Preview Visualize JSON

```
1 {  
2   "id": 55,  
3   "name": "",  
4   "balanceTotal": 0  
5 }
```

RUST



# Profile

nsnū status code 400 ua: 404

keptang2 / profile

GET http://127.0.0.1:8080/profile

Params Auth Headers (8) Body Pre-req. Tests Settings

raw JSON

```
1 {"id": xxx}
```

Body

Pretty Raw Preview Visualize Text

```
1 Json deserialize error: expected value at line 2 column 9
```

keptang2 / profile

GET http://127.0.0.1:8080/profileless

Params Auth Headers (8) Body Pre-req. Tests Settings

raw JSON

```
1 {"id": 40956}
```

Body

404 Not Found 5 ms 82 B

RUST



# Get All money

## កសារពក្តាត ទែបកត្ថុកត៉ែង

The screenshot shows the Postman application interface. At the top, it says "keptang2 / money". Below that, a "Save" button is visible. The request method is set to "GET" and the URL is "http://127.0.0.1:8080/money". Under the "Body" tab, there is a JSON payload: { "id": 888 }. The "Headers" tab shows "(8)" and the "JSON" tab is selected. The "Body" tab is also selected. The response section shows a status of "200 OK" with a time of "6 ms" and a size of "172 B". The response body is displayed in "Pretty" format:

```
{  
    "items": [],  
    "total": {  
        "balance": 0,  
        "id": 888,  
        "name": "noname"  
    }  
}
```

```
use actix_web::{ get, HttpResponse, Responder, web};  
use log::{ info};  
use serde_json::json;  
use serde::{Serialize, Deserialize};  
use crate::models::moneylist::*;

// សរុប struct ใหมេទាំងនេះដូចតួនាទីក្នុងការសែន  
#[derive(Serialize, Deserialize)]
struct UserdataUpgate {
    id: i32,
}

// GET /money: សោរបន់ទិន្នន័យរបាយការណ៍របាយការលើកម្រិតអាជីវកម្ម  
#[get("/money")]
async fn get_money(user_id: web::Json<UserdataUpgate>) -> impl Responder {
    info!("Keptang money");

    // គាត់ id ទីនេះមាន
    let userdata = user_id.into_inner();
    let id: i32 = userdata.id;
    // គាត់រើបចាយ
    let mut _user_money: i32 = 0;
    let mut money_item_all = vec![];
    let mut _user_name = "";
    // ករណីខ្សោយតាមឯកសារណ៍ទិន្នន័យ
    if id==40956 {
        _user_money = 115000;
        _user_name = "vivat";
        money_item_all = vec![  
            Moneylist {  
                list_id: 5,  
                description: "លើយីងខ្សោយ".to_string(),  
                date: "2023-03-15".to_string(),  
                amount: 100,  
                types: "expense".to_string(),  
            },  
            Moneylist {
```

RUST



# Get All money

## នសិក្សា តាមការបញ្ជូនតែង



keptang2 / money

Save

GET http://127.0.0.1:8080/money

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {  
2   "id": 40956  
3 }
```

Body Cookies Headers (3) Test Results

Status: 200 OK Time: 5 ms Size: 750 B

Pretty Raw Preview Visualize JSON

```
1 {  
2   "items": [  
3     {  
4       "amount": 100,  
5       "date": "2023-03-15",  
6       "description": "ເលើយខ្សោយសាតា",  
7       "list_id": 5,  
8       "types": "expense"  
9     },  
10     {  
11       "amount": 100,  
12       "date": "2023-03-15",  
13       "description": "ចេញខ្សោយ",  
14       "list_id": 4,  
15       "types": "expense"  
16     },  
17   ]  
}
```

RUST



# Get All money

nsđí status code 400

ua: 404

The screenshot shows a POST request to `http://127.0.0.1:8080/money`. The Body tab contains the following JSON:

```
1 {  
2   "id": 40956  
3 }
```

The response status is `200 OK`, and the body is:

```
1 {  
2   "id": 40956  
3 }
```

Details: Status: 200 OK

The screenshot shows a GET request to `http://127.0.0.1:8080/moneys`. The Body tab contains the following JSON:

```
1 {  
2   "id": "สกุณย์เจ็คศุนย์"  
3 }
```

The response status is `400 Bad Request`, and the body is:

```
1 Json deserialize error: expected value at line 2 column 9
```

Details: Status: 400 Bad Request

RUST



# Get money today

## នសាក់ប្រកបដូចតែង

keptang2 / today

GET http://127.0.0.1:8080/money/today

Params Authorization Headers (8) Body ● Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

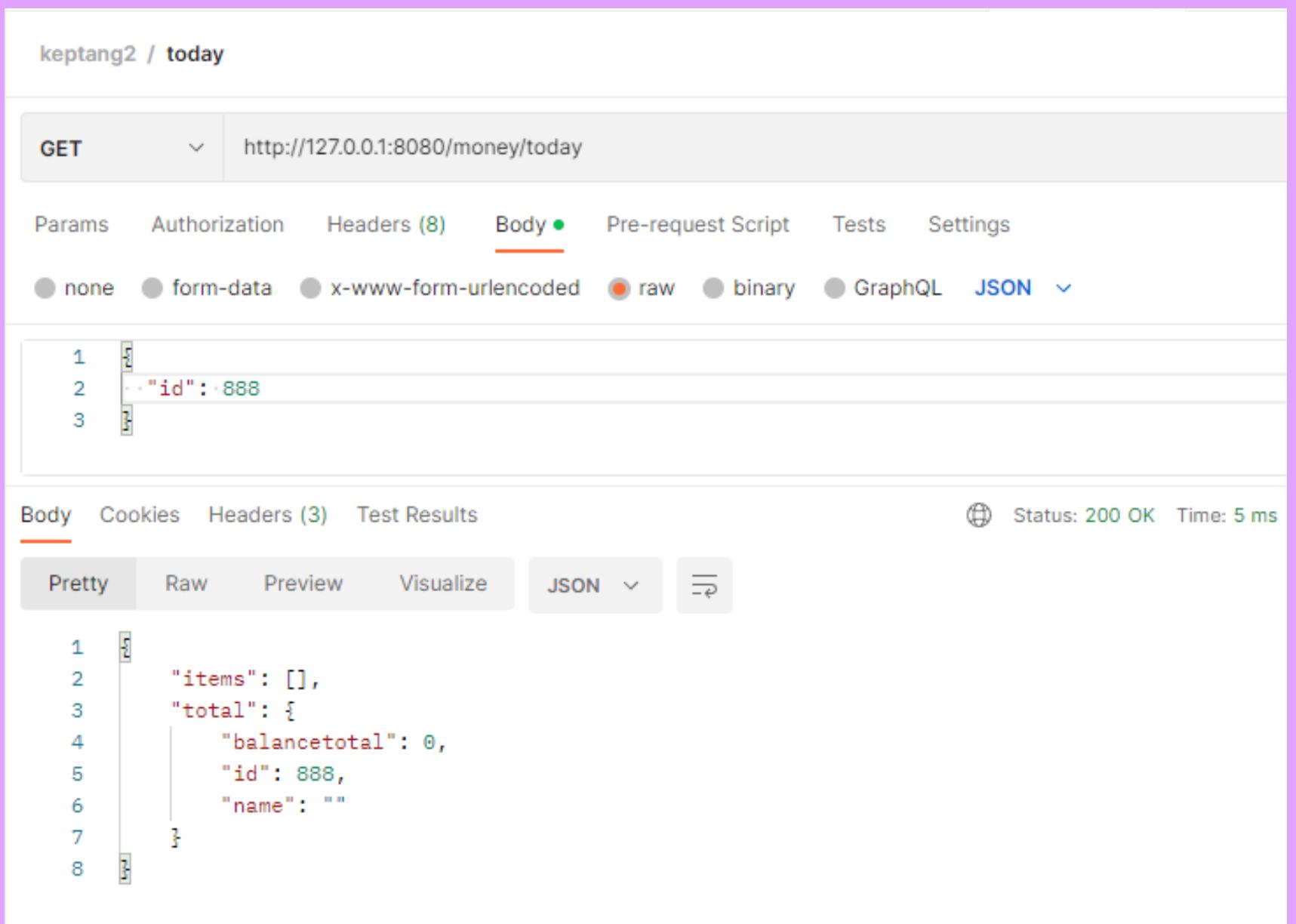
1 [ { "id": 888 } ]

Body Cookies Headers (3) Test Results

Status: 200 OK Time: 5 ms

Pretty Raw Preview Visualize JSON

1 [ { "items": [], "total": { "balancetotal": 0, "id": 888, "name": "" } } ]



```
use actix_web::{ get, HttpResponse, Responder, web };
use log::info;
use serde_json::json;
use serde::{Serialize, Deserialize};
use crate::models::moneylist::*;

// สร้าง struct ใหม่ที่มีเฉพาะส่วนที่คุณต้องการส่ง
#[derive(Serialize, Deserialize)]
struct UserdataUpdate {
    id: i32,
}

// GET /money: สำหรับอ่านข้อมูลรายการรับ-รายจ่ายวันนี้
#[get("/money/today")]
async fn get_money_today(user_id: web::Json<UserdataUpdate>) -> impl Responder {
    info!("Keptang today");

    // ค่า id ที่รับมา
    let userdata = user_id.into_inner();
    let id: i32 = userdata.id;

    // ค่าเริ่มต้น
    let mut _user_money: i32 = 0;
    let mut _user_name = "";
    let mut money_item_all = vec![];

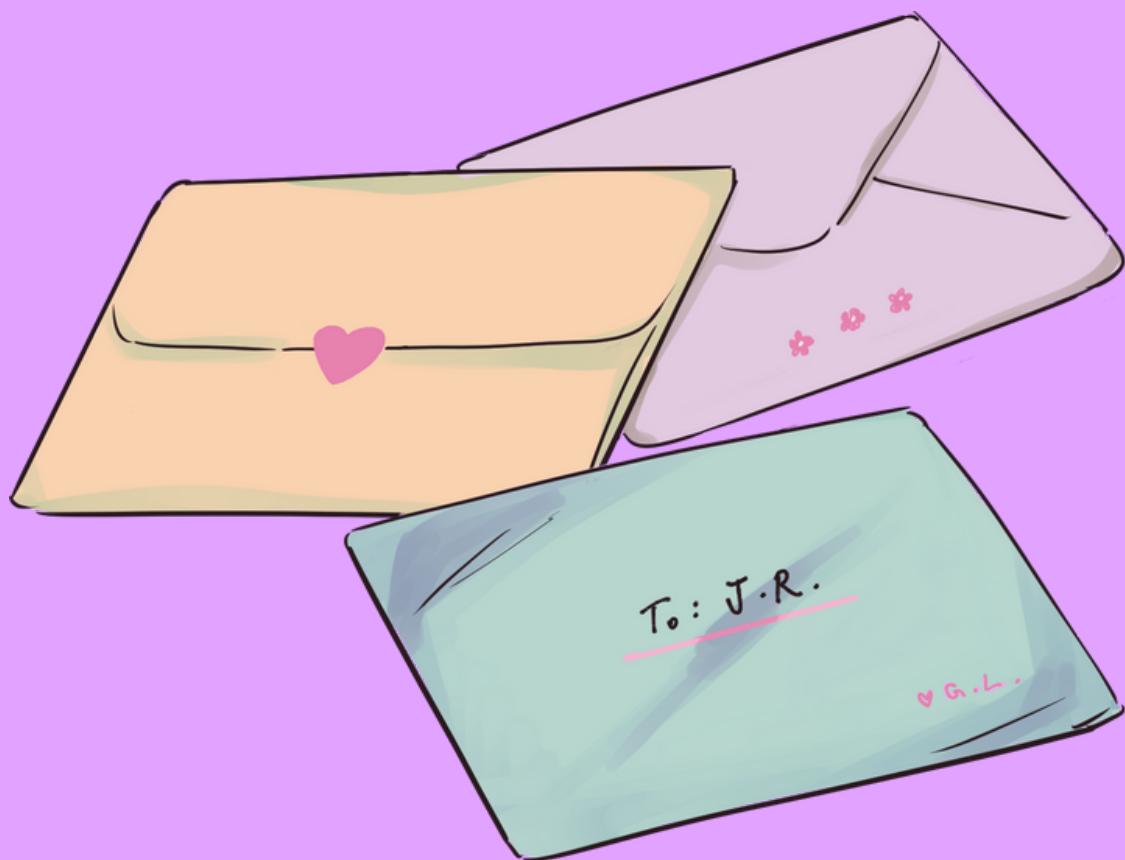
    if id==40956 {
        _user_money = 115000;
        _user_name = "vivat";
        money_item_all = vec![
            Moneylist {
                list_id: 5,
                description: "ເឡິງໜ້າ".to_string(),
                date: "2023-03-15".to_string(),
                amount: 100,
                types: "expense".to_string(),
            }
        ];
    }
}
```

RUST



# Get money today

នសិក្សា តែបកាបុកតែង



keptang2 / today

Save

GET http://127.0.0.1:8080/money/today

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {  
2   "id": 40956  
3 }
```

Body Cookies Headers (3) Test Results Status: 200 OK Time: 6 ms Size: 516 B

Pretty Raw Preview Visualize JSON

```
1 [  
2   {  
3     "amount": 100,  
4     "date": "2023-03-15",  
5     "description": "ធៀនខ្សោយសារ",  
6     "list_id": 5,  
7     "types": "expense"  
8   },  
9   {  
10     "amount": 100,  
11     "date": "2023-03-15",  
12     "description": "ចូលខ្សោយទៅ",  
13     "list_id": 4,  
14     "types": "expense"  
15   },  
16 ]
```

RUST



# Get money today

nsđñ status code 400 ua: 404

keptang2 / today

GET http://127.0.0.1:8080/money/today

Body (Pretty)

```
1 {  
2   "id": "Jumpbox!"  
3 }
```

Status: 400 Bad Request

1 Json deserialize error: invalid type: string "\u{e38}\u{e4a}Jumpbox!", expected

keptang2 / today

GET http://127.0.0.1:8080/money/toyou

Body (Raw)

```
1 {  
2   "id": 40956  
3 }
```

Status: 404 Not Found

1

RUST



# Post add money

## នសតិភកា តែបកបាបដូរត៉ែង

POST http://127.0.0.1:8080/money/saving

Body (8)

```
1 {  
2   "user_data": {  
3     "id": 40956  
4   },  
5   "data_item": {  
6     "description": "រាយរៀបទសំខាន់",  
7     "date": "2023-03-15",  
8     "amount": 100000,  
9     "types": "income"  
10    }  
11 }
```

Pretty Raw Preview Visualize JSON

```
1 {  
2   "items": {  
3     "amount": 100000,  
4     "date": "2023-03-15",  
5     "description": "រាយរៀបទសំខាន់",  
6     "list_id": 6,  
7     "types": "income"  
8   },  
9   "text": "ការបញ្ចូលទីផ្សារ",  
10  "u_name": "vivat"  
11 }
```

```
use actix_web::{web, HttpResponse, Responder, post}; Niruthub, 6 days ago • K  
use serde::{Serialize, Deserialize};  
use log::{debug, info};  
use serde_json::json;  
  
use crate::models::moneylist::*;

Niruthub, 5 days ago | 1 author (Niruthub)  
// ស្មោគ struct នៃមីត្តភាពសំខាន់ដែលត្រូវការ Request  
#[derive(Serialize, Deserialize)]  
struct UserdataUpgate {  
    id: i32,  
}  
Niruthub, 6 days ago | 1 author (Niruthub)  
#[derive(Serialize, Deserialize)]  
struct MoneylistUpgate {  
    description: String,  
    date: String,  
    amount: i32,  
    types: String,  
}  
Niruthub, 5 days ago | 1 author (Niruthub)  
// រួម គឺជាស្មោគ JSON ដែលត្រូវការ Request  
#[derive(Serialize, Deserialize)]  
struct PostMoneyRequest {  
    user_data: UserdataUpgate,  
    data_item: MoneylistUpgate,  
}  
// POST /money: តាមរីបចំនូវឯកសាររាយរៀបទសំខាន់  
#[post("/money/saving")]  
async fn post_money(input_data: web::Json<PostMoneyRequest>) -> impl Responder {  
    info!("post money");  
    debug!("post: ✅");
```

RUST



# Post add money

## not status code 400 แต่ 404

```
1 {  
2   "user_data": {  
3     "id": 40956  
4   },  
5   "data_item": {  
6     "description": "รายรับทดสอบ",  
7     "date": "2023-03-15",  
8     "amount": 100000,  
9     "types": "income"  
10    }  
11 }
```

```
1 {  
2   "user_data": {  
3     "id": 40956  
4   },  
5   "data_item": {  
6     "description": "รายรับทดสอบ",  
7     "date": "2023-03-15",  
8     "amount": 100000,  
9     "types": "income"  
10    }  
11 }
```

The response shows a 404 error with the message "404" and an exclamation mark icon.

RUST



# Put edit money

នរបាបកាត់ ពួកតាបត្រកត៉ែង



```
use crate::models::moneylist::*;
use actix_web::{put, web, HttpResponse};
use log::{debug, info};
use serde::{Deserialize, Serialize};
use serde_json::json;

Niruthub, 6 days ago | 1 author (Niruthub)
// ស្រាវ ស្រាវ និងការសំគាល់ការសំគាល់
#[derive(Serialize, Deserialize)]
struct UserdataUpdate {
    id: i32,
}

Niruthub, 6 days ago | 1 author (Niruthub)
#[derive(Serialize, Deserialize)]
struct MoneylistUpdate {
    description: String,
    date: String,
    amount: i32,
    types: String,
}

Niruthub, 6 days ago | 1 author (Niruthub)
#[derive(Serialize, Deserialize)]
struct PostMoneyRequest {
    user_data: UserdataUpdate,
    dataitem: MoneylistUpdate,
}

// PUT /money/edit/{id}: នឹង JSON ដើម្បីចាយកិច "expense"/"income" បើនរាយការ JSON ដើម្បីខ្សោយ
#[put("/money/item/{id}")]
async fn put_money(list_id: web::Path<i32>, input_data: web::Json<PostMoneyRequest>
    info!("put money by id");
    debug!("id: {} និង list_id");

    // ចារើនុញ្ញ ដើម្បីទទួលបានការបញ្ជូន (តាមរយកការបញ្ជូន ដើម្បីបាន និង Let)
    let input_data = input_data.into_inner();
    let id: i32 = list_id.to_string().parse().unwrap();
```

RUST



# Put edit money

ផ្តល់ព័ត៌មានការបញ្ចូនតម្លៃ

PUT http://127.0.0.1:8080/money/item/3

Params Authorization Headers (8) Body • Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {  
2   "user_data": {  
3     "id": 40956  
4   },  
5   "dataitem": {  
6     "description": "រាយរ៉ាប់ទេសចរណ៍",  
7     "date": "2023-03-15",  
8     "amount": 1000,  
9     "types": "expenses"  
10    }  
11 }
```

Body Cookies Headers (3) Test Results

Pretty Raw Preview Visualize JSON

```
1 {  
2   "items_new": {  
3     "amount": 1000,  
4     "date": "2023-03-15",  
5     "description": "រាយរ៉ាប់ទេសចរណ៍",  
6     "list_id": 3,  
7     "types": "expenses"  
8   },  
9   "items_old": {  
10    "amount": 100,  
11    "date": "2023-03-15",  
12    "list_id": 3,  
13    "types": "expenses"  
14  }
```

PUT http://127.0.0.1:8080/money/item/8

Params Authorization Headers (8) Body • Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL

```
1 {  
2   "user_data": {  
3     "id": 40956  
4   },  
5   "dataitem": {  
6     "description": "រាយរ៉ាប់ទេសចរណ៍",  
7     "date": "2023-03-15",  
8     "amount": 1000,  
9     "types": "expenses"  
10    }  
11 }
```

Body Cookies Headers (3) Test Results

Pretty Raw Preview Visualize JSON

```
1 {  
2   "items_new": {  
3     "amount": 1000,  
4     "date": "2023-03-15",  
5     "description": "រាយរ៉ាប់ទេសចរណ៍",  
6     "list_id": 8,  
7     "types": "expenses"  
8   },  
9   "items_old": {  
10    "amount": 0,  
11    "date": "មិនអាច",  
12    "list_id": 8,  
13    "types": "expenses"  
14  }
```

RUST



# Delete money

## នសិក្សា តាមការបញ្ចូកតម្លៃ



```
use actix_web::{delete, web, HttpResponse};      Niruthub, 6 days ago • Keptang_V
use log::{debug, info};
use serde_json::json;
use serde::{Deserialize, Serialize};
use crate::models::moneylist::*;

Niruthub, 5 days ago | 1 author (Niruthub)
// ស្នើសុំ struct ໃໝ່ທີ່ມີເພາະສ່ວນທີ່ຕ້ອງການ Request
#[derive(Serialize, Deserialize)]
struct UserdataUpdate {
    id: i32,
}

// DELETE /money/delete/{id}: "ໄມ້ຕ້ອງການຮັບຂໍ້ມູນໄດ້ ແລ້ວຈະລබຍະຮັງການຮ່າຍທີ່ມີອຸ່ນດ້ວຍ list_id"
#[delete("/money/item/{id}")]
async fn delete_money(list_id: web::Path<i32>, input_data: web::Json<UserdataUpdate>){
    info!("delete money by list_id");
    debug!("list_id: {} ❌", list_id);

    // ດ້ວຍເຫັນ ທີ່ຮັບມາແນບ JSON (ຄ້ອຍກຳກັນ ເຕີນ mut ມີລັງ Let)
    let user_id = input_data.into_inner();
    let id: i32 = list_id.to_string().parse().unwrap();

    // ສ່າງມູນຄົງຂໍ້ມູນເດີນຂອງ id 3
    let mut data_old = Moneylist {
        list_id: 3,
        description: "ແມໄທ".to_string(),
        date: "2023-03-15".to_string(),
        amount: 100,
        types: "income".to_string(),
    };

    // ສ່າງມູນຄົງ ຈະລບຍ ຮາຍການທີ່ id 3
    if id == 3 && user_id.id == 40956{
        debug!("DELETE {} ❌", list_id);
    }else{ // ສ່າງມູນຄົງໃນນີ້ໃນ ຖຸານນີ້ຂໍ້ມູນ
        debug!("none data list_id: {} ❌", list_id);
```

RUST



# Delete money

លក់ប្រាកា តួបកាប់គុណតែង

DELETE http://127.0.0.1:8080/money/item/8

Body

```
1 {  
2   "id": 40956  
3 }
```

DELETE http://127.0.0.1:8080/money/item/3

Body

```
1 {  
2   "id": 40956  
3 }
```

200 OK

Body

```
1 {  
2   "items": [  
3     {  
4       "amount": 100,  
5       "date": "2023-03-15",  
6       "description": "ແມໃຫ້",  
7       "list_id": 3,  
8       "types": "income"  
9     },  
10    {  
11      "text": "ទាករសនខមូលលະទេះ"  
12    }  
13  ]  
14 }
```

# ขั้นตอนการเชื่อมต่อ โดยใช้ JavaScript Fetch API Rust(Backend) <-> HTML(Frontend)

สร้าง endpoint ของ API ในภาษา Rust ที่เขียนโดย actix\_web ซึ่งจะรับ request และส่ง response กลับไปยัง frontend

ในหน้า HTML Frontend สร้างฟอร์มและปุ่มเพื่อเรียกใช้ API ด้วย JavaScript Fetch API

เมื่อผู้ใช้กดปุ่มเพื่อเรียกใช้ API ด้วย JavaScript Fetch API จะเรียก endpoint ของ API ในภาษา Rust ที่เขียนโดย actix\_web ที่สร้างไว้ก่อนหน้านี้

endpoint ของ API ในภาษา Rust ที่เขียนโดย actix\_web จะรับ request และดำเนินการตามที่ได้กำหนดไว้

เมื่อดำเนินการเสร็จสิ้น จะส่ง response กลับไปยัง frontend โดย JavaScript Fetch API จะจัดการ response และแสดงผลตามที่ได้กำหนดไว้

```

use actix_web::{get, post, web, App, HttpResponse, HttpServer, Responder};
use serde::{Deserialize, Serialize};

#[derive(Serialize, Deserialize)]
struct User {
    id: i32,
    name: String,
    email: String,
}

#[get("/users")]
async fn get_users() -> impl Responder {
    let user = User {
        id: 1,
        name: "John Doe".to_owned(),
        email: "john.doe@example.com".to_owned(),
    };
    HttpResponse::Ok().json(user)
}

#[post("/users")]
async fn add_user(user: web::Json<User>) -> impl Responder {
    HttpResponse::Created().json(user.0)
}

#[actix_web::main]
async fn main() -> std::io::Result<()> {
    HttpServer::new(|| App::new().service(get_users).service(add_user))
        .bind("127.0.0.1:8080")?
        .run()
        .await
}

```

```

<html>
<head>
    <meta charset="UTF-8">
    <title>Fetch API Example</title>
</head>
<body>
    <h1>Fetch API Example</h1>
    <button id="get-users-btn">Get Users</button>
    <button id="add-user-btn">Add User</button>
    <div id="result"></div>

    <script>
        const getUsers = async () => {
            const response = await fetch('http://localhost:8080/users');
            const data = await response.json();
            document.getElementById('result').innerHTML = `<p>${JSON.stringify(data)}</p>`;
        };

        const addUser = async () => {
            const user = { id: 2, name: "Jane Doe", email: "jane.doe@example.com" };
            const response = await fetch('http://localhost:8080/users', {
                method: 'POST',
                headers: {
                    'Content-Type': 'application/json'
                },
                body: JSON.stringify(user)
            });
            const data = await response.json();
            document.getElementById('result').innerHTML = `<p>${JSON.stringify(data)}</p>`;
        };

        document.getElementById('get-users-btn').addEventListener('click', getUsers);
        document.getElementById('add-user-btn').addEventListener('click', addUser);
    </script>
</body>
</html>

```

# ขั้นตอนการเชื่อมต่อ API ของภาษา Rust ที่เขียนโดยใช้ Actix Web กับฐานข้อมูล MySQL ด้วย sqlx

เพิ่ม dependencies ของ sqlx และ mysql ในไฟล์ Cargo.toml

```
1 [dependencies]
2 actix-web = "4.0.0"
3 sqlx = { version = "0.5.8", features = ["mysql"] }
```

สร้าง API endpoint สำหรับเรียกข้อมูล user จาก MySQL โดยใช้ SQLx

```
1 #[get("/users")]
2 async fn get_users(pool: web::Data<MySqlPool>) -> impl Responder {
3     let users = sqlx::query!("SELECT * FROM users")
4         .fetch_all(pool.get_ref())
5         .await
6         .unwrap();
7
8     HttpResponse::Ok().json(users)
9 }
10
```

import dependencies ต่อ ๆ

```
1 use actix_web::{get, web, App, HttpResponse, HttpServer, Responder};
2 use sqlx::{mysql:: MySqlPool, Pool};
```

# ขั้นตอนการเชื่อมต่อ API ของภาษา Rust ที่เขียนโดยใช้ Actix Web กับฐานข้อมูล MySQL ด้วย sqlx (ต่อ)

กำหนดการเชื่อมต่อ  
MySQL โดยใช้ SQLx

```
● ● ●  
1 async fn establish_connection() -> MySqlPool {  
2     MySqlPool::connect("mysql://user:password@localhost/db_name").  
3     await.unwrap()  
4 }
```

กำหนด routes และ  
start HTTP server

```
● ● ●  
1 #[actix_web::main]  
2 async fn main() -> std::io::Result<()> {  
3     let pool = establish_connection().await;  
4  
5     HttpServer::new(move || {  
6         App::new()  
7             .data(pool.clone())  
8             .service(get_users)  
9     })  
10    .bind("127.0.0.1:8080")?  
11    .run()  
12    .await  
13 }  
14 }
```

กลับไปเมนู

WANT YOU

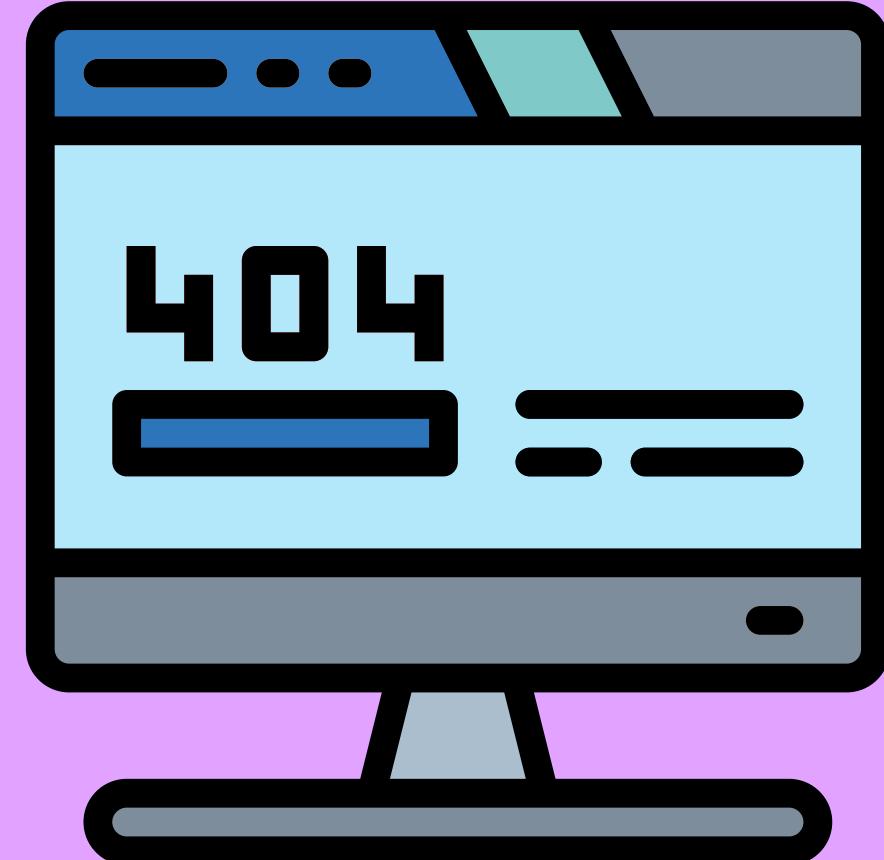
# 404

lost in space

you have reached  
the edge of the universe.



# 404 FOUND



## 404 ERROR PAGE NOT FOUND

