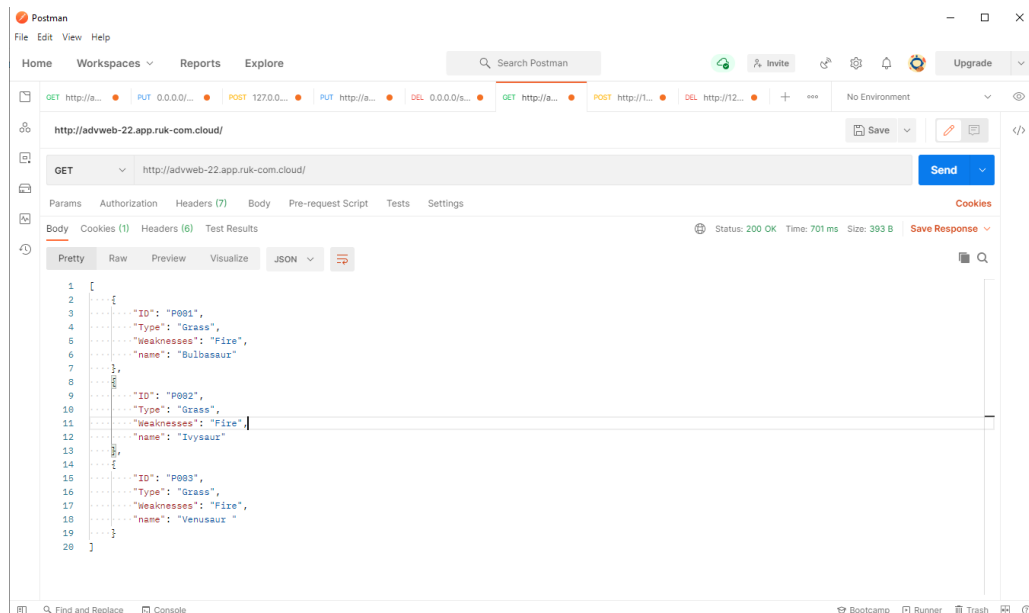


โครงสร้างของ NOSQL.Redisl ตอนเริ่มต้นได้สร้างข้อมูลแบบ Hashes จะประกอบไปด้วย Hashes 3 ตัว ใน Hashes จะเก็บข้อมูล Pokemon ประกอบไปด้วย ID Name Type Weakness -ของแต่ละตัว

## การ GET ค่าทั้งหมด

API <http://advweb-22.app.ruk-com.cloud/>

HTTP method GET



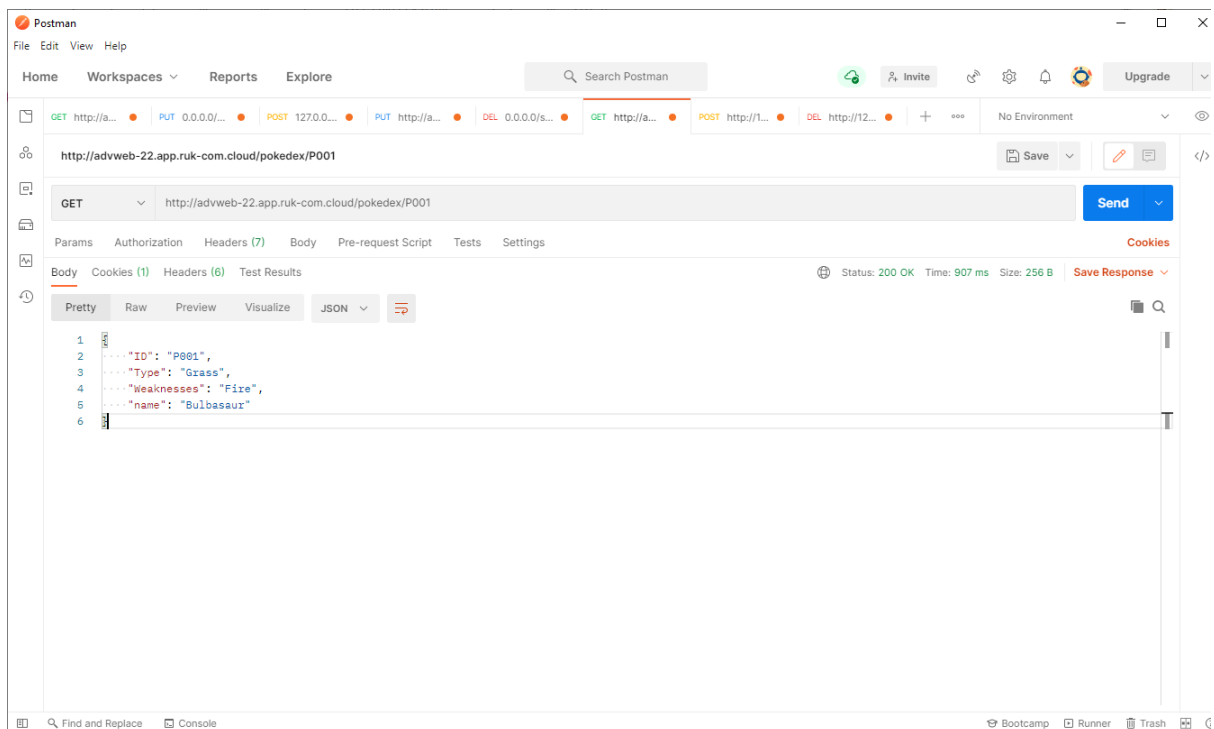
```
# Get All
@app.route('/', methods=['GET'])
def hello_world():
    b = db.keys() #รับค่ามาจาก Redis
    b.sort() #รับค่ามาจาก Redis แล้วเรียง ID
    result = [] # ประกาศตัวแปร result เป็น array
    for id in b:
        result.append(db.hgetall(id)) # hgetall ใช้ในการเรียกข้อมูล HASH ตาม Id ที่ใส่ แล้วเพิ่มลง Array
    return jsonify(result)
```

## การ GET ค่ามาตัวเดียว

API <http://advweb-22.app.ruk-com.cloud/pokedex/<id>>

HTTP method GET

ในตัวอย่างทำการเรียก id P001

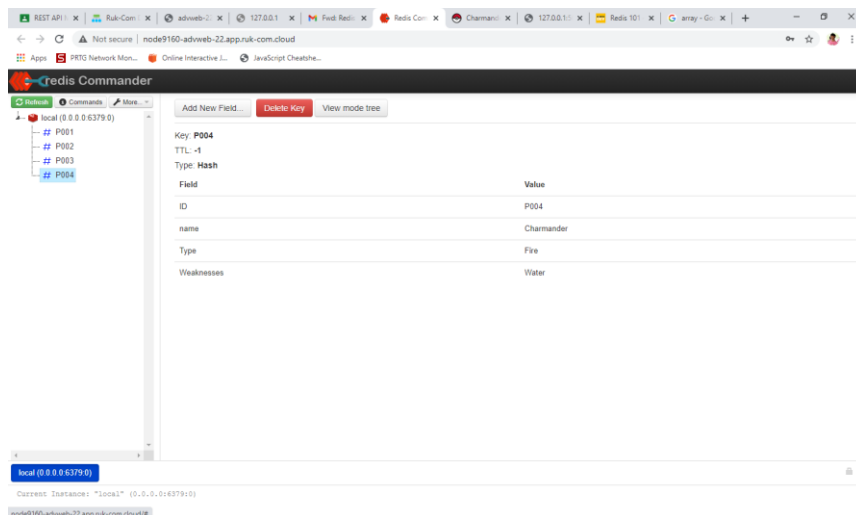
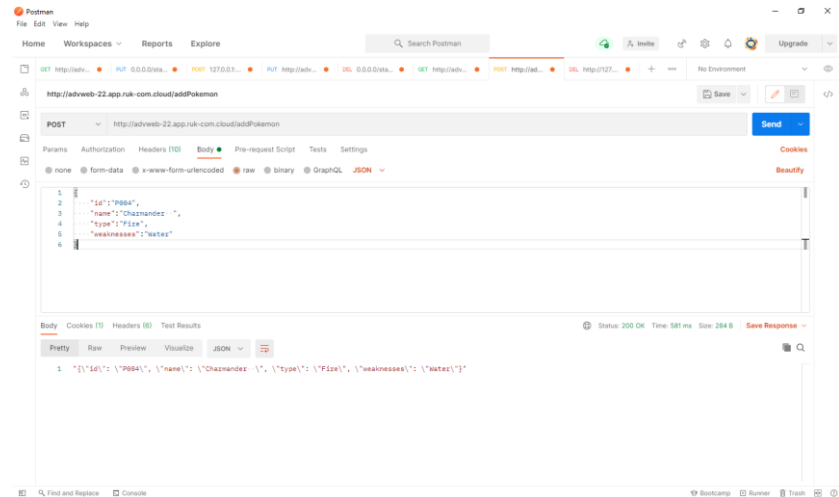


```
# Get Single Pokemon
@app.route('/pokedex/<id>', methods=['GET'])
def get_book(id):
    pokemon = db.hgetall(id) # hgetall ใช้ในการเรียกข้อมูล HASH ตาม Id
    print(pokemon)
    return pokemon
```

## การ SET ค่า

API <http://adweb-22.app.ruk-com.cloud/addPokemon>

HTTP method POST



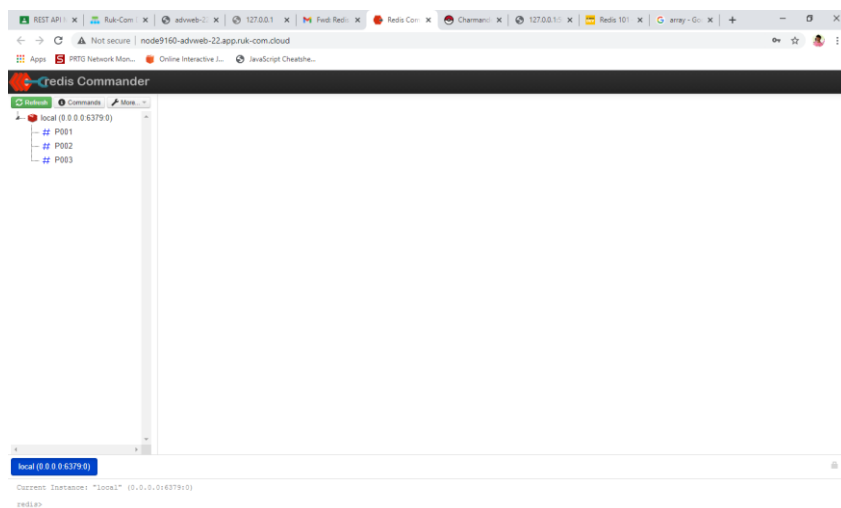
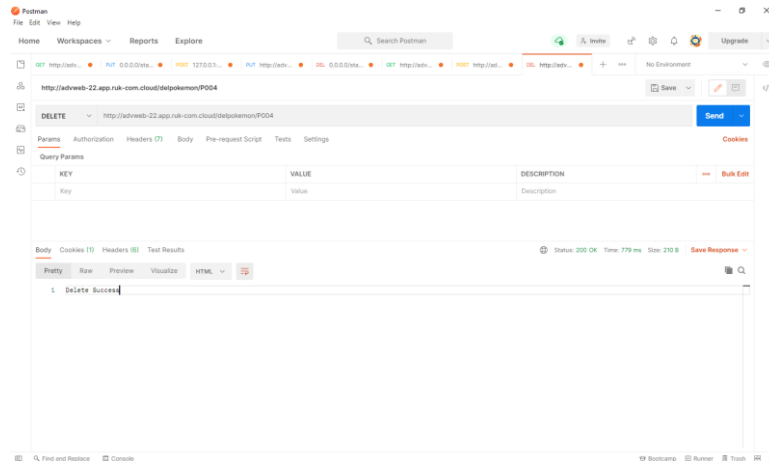
```
# Set Pokemon
@app.route('/addPokemon', methods=['POST'])
def addPokemon():
    id = request.json['id']
    name = request.json['name']
    type = request.json['type']
    weaknesses = request.json['weaknesses']

    new_pokemon = Pokedex(id, name, type, weaknesses) # สร้าง OBJ ชื่อ new_pokemon ใส่ค่า พารามิเตอร์ที่รับเข้ามา
    jsonStr = json.dumps(new_pokemon.__dict__)
    # hset เป็นคำสั่งสร้าง Hash
    # db.hset(ชื่อ HASH, ชื่อ Field, Value)
    db.hset(id, "ID", new_pokemon.id)
    db.hset(id, "name", new_pokemon.name)
    db.hset(id, "Type", new_pokemon.type)
    db.hset(id, "Weaknesses", new_pokemon.weaknesses)
    print(jsonStr)
    return jsonify(jsonStr)
```

## การ DELETE คำ

API `http://adweb-22.app.ruk-com.cloud/delpokemon/<id>`

HTTP method DEL



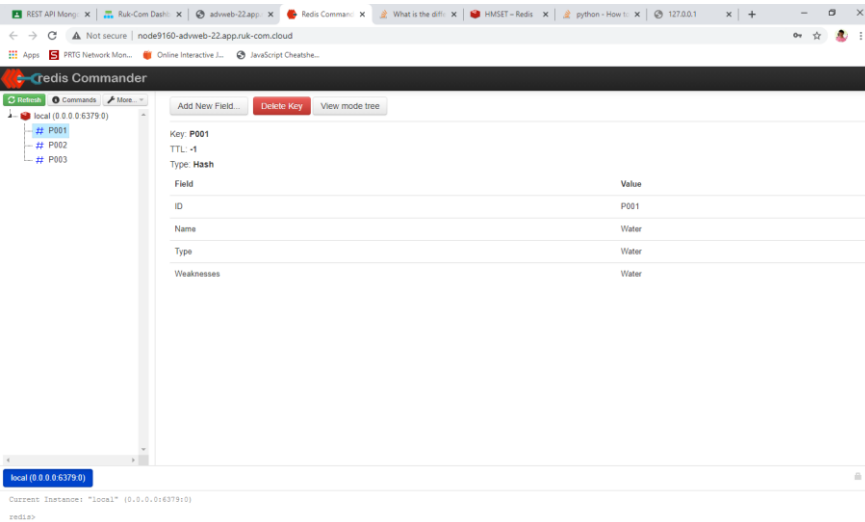
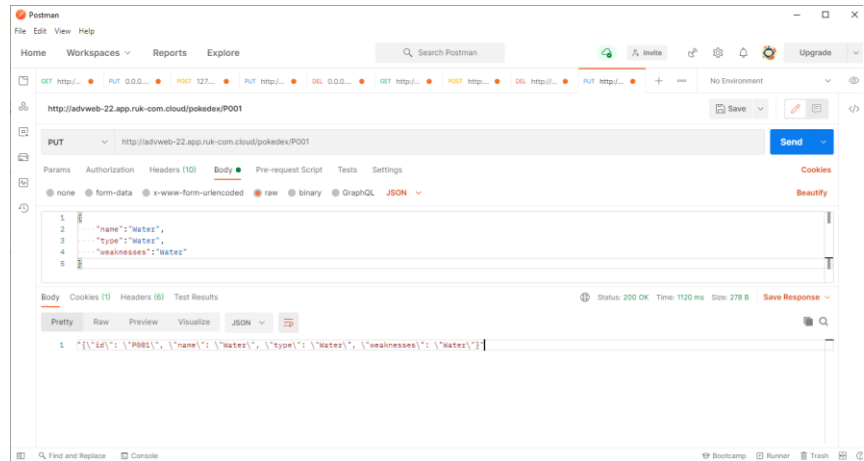
```
# Del Pokemon
@app.route('/delpokemon/<id>', methods=['DELETE'])
def delete_pokemon(id):
    db.delete(id) #delete เป็นคำสั่งลบข้อมูล
    return "Delete Success"
```

## การ Update ค่า

API `http://adweb-22.app.ruk-com.cloud/pokedex/<id>`

HTTP method PUT

ในตัวอย่างทำการเรียก id P001



```
# Update a Pokemon
@app.route('/pokedex/<id>', methods=['PUT'])
def update_staff(id):
    name = request.json['name']
    type = request.json['type']
    weaknesses = request.json['weaknesses']

    update_pokemon = Pokedex(id, name, type, weaknesses) # สร้าง OBJ ชื่อ new_pokemon ใส่ค่า พารามิเตอร์ที่รับเข้ามา
    jsonStr = json.dumps(update_pokemon.__dict__)
    pokemon = {
        "ID": id,
        "Name": update_pokemon.name,
        "Type": update_pokemon.type,
        "Weaknesses": update_pokemon.weaknesses
    }
    db.hmset(id, pokemon) #hmset ต่างจาก hset : hset สามารถ set ได้ทีละ Field : hmset สามารถ set ได้หลาย Field
    print(jsonStr)
    return jsonify(jsonStr)
```