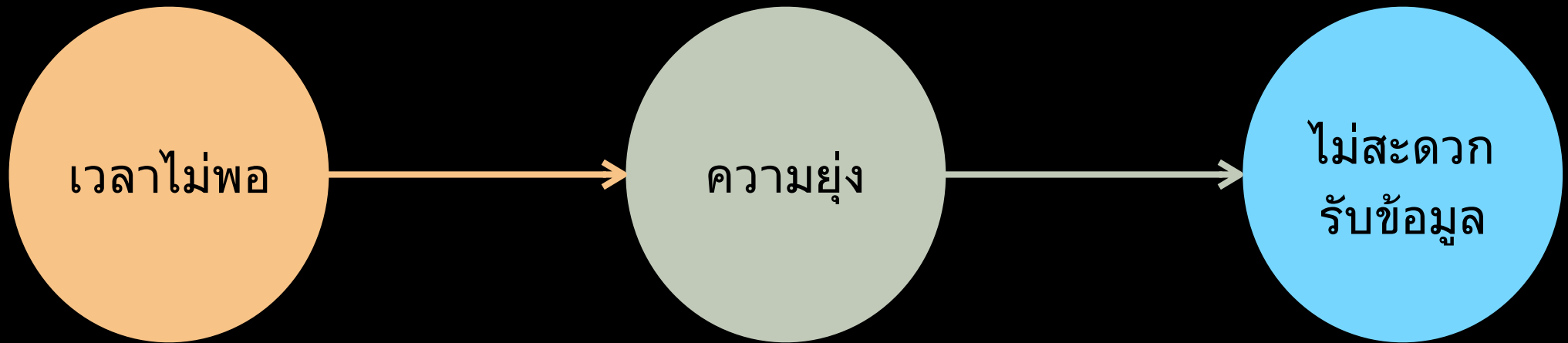


การทำงานคร่าว ๆ ของแอปพลิเคชัน ติดตามยอดจำนวน COVID-19 อย่างง่าย

วริศ ศรีปทุมรักษ์ ม.4/1-9

ที่มาและความสำคัญ

ปัญหา



- มีเวลาน้อยและจำกัด
- ความรีบ
- ไม่มีเวลาให้ความสำคัญกับสิ่งอื่น ๆ ที่สำคัญรองลงมา

เมื่องานที่ต้องทำให้เสร็จภายในเวลาจำกัดมีจำนวนมากทำให้ความสนใจกับงานนั้นก่อนค่อยให้ความสำคัญกับงานอื่นๆ ในภายหลัง

ขณะนั้น อุปกรณ์ที่สามารถรับข้อมูลได้อาจขาดแหล่งพลังงาน ไม่ว่างให้ใช้งานหรืออยู่ในระยะที่ไกลตัว

การแก้ปัญหา วัตถุประสงค์และขอบเขตการพัฒนา

การแก้ปัญหา

- สร้างแอปพลิเคชันบน Apple Watch
 - นำข้อมูลอัตราการติดเชื้อ / อัตราตายติดเชื้อ / อัตราการเสียชีวิตที่เพิ่มขึ้นในแต่ละวัน
 - เนื่องจากผู้คนส่วนใหญ่ใช้ร่วม ประกอบกับการทำงานในชีวิตประจำวันต่าง ๆ และอยู่ติดตัว

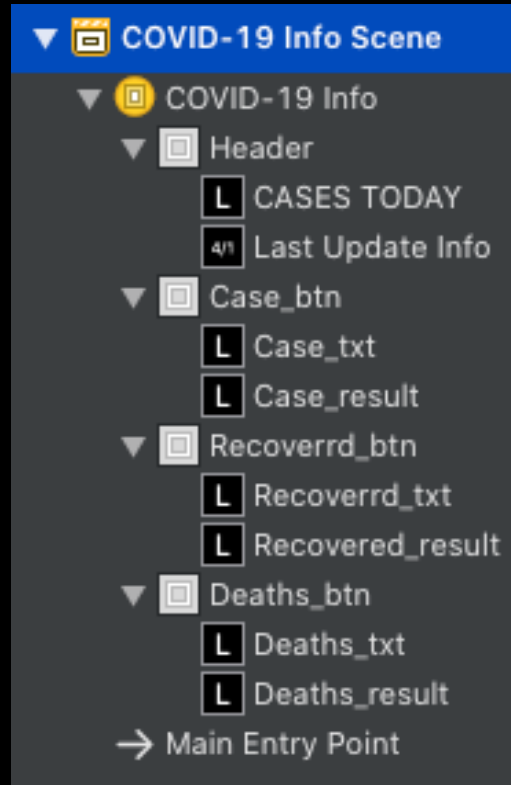
วัตถุประสงค์

- ให้ข้อมูลต่าง ๆ COVID-19 เป็นสิ่งใกล้ตัว
- สามารถติดตามข้อมูล COVID-19 ได้ทุกที่ทุกเวลา
- เพื่อผู้ที่มีเวลาน้อยและจำกัด

ขอบเขตการพัฒนา

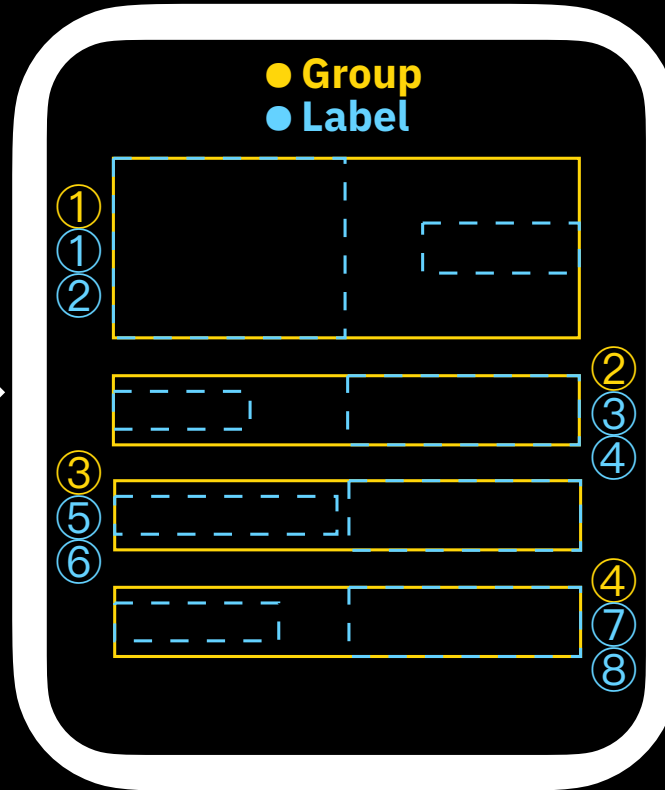
- นำข้อมูลอัตราการติดเชื้อ / อัตราตายติดเชื้อ / อัตราการเสียชีวิตที่เพิ่มขึ้นในแต่ละวัน
- ต้องสร้างแอปพลิเคชันนี้ให้ใช้ได้บน Apple Watch
- ต้องอัปเดตข้อมูลตามที่ API ดึงมาจากเซิร์ฟเวอร์ได้ เพื่อให้ข้อมูลตรงตามปัจจุบันมากที่สุด

User Interface

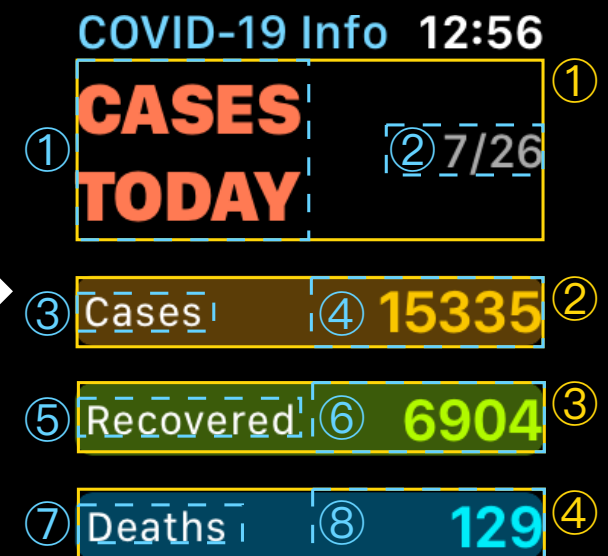


รายชื่อ Objects

ไฟล์ Interface.storyboard



โครงสร้างแอปพลิเคชัน



หน้าจอแอปพลิเคชัน

การนำข้อมูลมาใช้งาน

COVID-19 Info 12:56

**CASES
TODAY**

7/26

Cases	15335
Recovered	6904
Deaths	129

นำข้อมูลมาจากของกรมควบคุมโรคโดยใช้ **API** ดึงมา

ตัวอย่างข้อมูล (JSON)

```
{ "Confirmed"      : 497302,  
  "Recovered"     : 334693,  
  "Hospitalized"  : 158550,  
  "Deaths"        : 4059,  
  "NewConfirmed"  : 15335,  
  "NewRecovered"  : 6904,  
  "NewHospitalized": 8302,  
  "NewDeaths"     : 129,  
  "UpdateDate"    : "25\07\2021 11:39",  
  "DevBy"         : "https://  
www.kidkarnmai.com/" }
```

ใช้ข้อมูลพวกนี้มา
แทนใน Labels

“API นี่คืออะไร?”

API

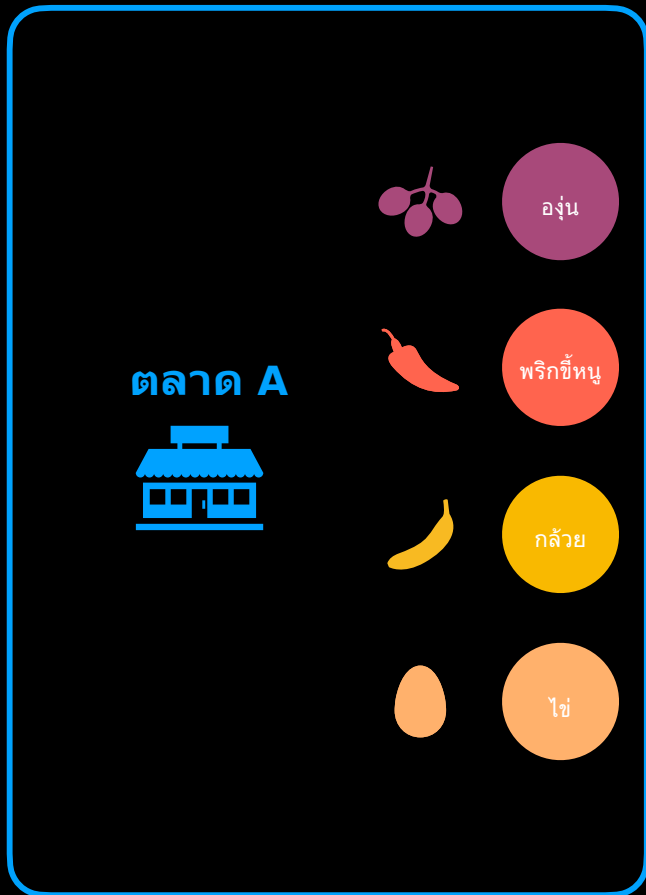
Application Programming Interface

(ส่วนต่อประสานโปรแกรมประยุกต์)

API คือตัวกลางที่ทำหน้าที่คอย**รับฟังคำสั่ง**ต่าง ๆ โดยสามารถประมวลผลและกระทำกับ**ข้อมูล**ที่ API นำมาจากเซิร์ฟเวอร์ได้
แล้วนำข้อมูลที่เอามา**ส่งกลับไป**ยังตัวที่ส่งคำสั่ง

“API นี้อะ?” — ตัวอย่างบ้าน ๆ

คำสั่ง ให้ไปซื้อพริกชี้หนู และ ไข่ อย่างละ 1 ที่ตลาด A



ผล ได้รับพริกชี้หนู 1 เม็ด และ ไข่ 1 ฟอง



คนสั่ง

“API นี่คือ?” — ตัวอย่างบ้าน ๆ

คำสั่ง ให้นำข้อมูล NewConfirmed, NewRecovered, และ NewDeaths มา

เซิร์ฟเวอร์

Con
firmed

Deaths

Reco
vered

NewCon
firmed

Hospi
talized

NewRe
covered

NewHos
pitaliz
ed

New
Deaths

Update
Date

DevBy

ผล ได้ข้อมูลว่า

“NewConfirmed” มีค่าเป็น 15335,

“NewRecovered” มีค่าเป็น 6904,

“NewDeaths” มีค่าเป็น 129

API

CASES
TODAY

Cases 15335

Recovered 6904

Deaths 129

แอปพลิเคชัน

น

การแทนข้อมูล

ในการเรียกข้อมูลจากเซิร์ฟเวอร์ผ่าน API มีส่วนประกอบสำคัญ ดังต่อไปนี้:—

```
import WatchKit
import Foundation

//var Cases: WKInterfaceLabel!
//var Recovered: WKInterfaceLabel!
//var Deaths: WKInterfaceLabel!

class InterfaceController: WKInterfaceController {
    // var Int_NewConfirmed: Int
    // var Int_NewRecovered: Int
    // var Int_NewDeaths: Int

    1 @IBOutlet var LastUpdateInfo: WKInterfaceDate!
    @IBOutlet var Case_Label: WKInterfaceLabel!
    @IBOutlet var Recovered_Label: WKInterfaceLabel!
    @IBOutlet var Deaths_Label: WKInterfaceLabel!

    override func awake(withContext context: Any?) {
        // Configure interface objects here.
        super.awake(withContext: context)
    }

    override func willActivate() {
        // This method is called when watch view controller is about to be visible to user
        super.willActivate()
    }

    // let params = ["username": "", "password": ""] as DictionaryString, String

    2 var request = URLRequest(url: URL(string: "https://covid19.th-stat.com/json/covid19v2/getTodayCases.json")!)
    request.httpMethod = "GET"
    request.addValue("application/json", forHTTPHeaderField: "Content-Type")
    let task = URLSession.shared.dataTask(with: request, completionHandler: {data, response, error → Void in
        print(response!)
        do {
            let json = try JSONSerialization.jsonObject(with: data!) as! DictionaryString, AnyObject
            print("JSON success", json)
            var Result_AllCases: String
            var Result_AllRecovered: String
            var Result_AllDeaths: String
            var Result_AllNewDate = Date()
            let result: [String: Any] = json
            if let All_Case = result["Cases"] as? Int {
                Result_AllCase = String(All_Case)
            } else {
                Result_AllCase = "Cannot Access API"
            }
            if let All_Recovered = result["Recovered"] as? Int {
                Result_AllRecovered = String(All_Recovered)
            } else {
                Result_AllRecovered = "Cannot Access API"
            }
            if let All_Deaths = result["Deaths"] as? Int {
                Result_AllDeaths = String(All_Deaths)
            } else {
                Result_AllDeaths = "Cannot Access API"
            }
            if let All_Date = result["Updatedate"] as? String {
                Result_AllDate = All_Date
            } else {
                Result_AllDate = "Cannot Access API"
            }
            let calendar = Calendar.current
            self.Case_Label.setText(Result_AllCase)
            self.Recovered_Label.setText(Result_AllRecovered)
            self.Deaths_Label.setText(Result_AllDeaths)
            self.LastUpdateInfo.setCalendar(calendar)
        } catch {
            print("error")
            self.Case_Label.setText("x")
            self.Recovered_Label.setText("x")
            self.Deaths_Label.setText("x")
        }
    })
    task.resume()
    4
}

override func didDeactivate() {
    super.didDeactivate()
}
```

1 @IBOutlet var LastUpdateInfo : WKInterfaceDate!
@IBOutlet var Case_Label : WKInterfaceLabel!
@IBOutlet var Recovered_Label: WKInterfaceLabel!
@IBOutlet var Deaths_Label : WKInterfaceLabel!

@IBOutlet เป็นกำหนดค่าของตัวแปร Interface มาเชื่อมกับตัวแปรที่ใช้ในการเรียกข้อมูลจากเซิร์ฟเวอร์

2 var request = URLRequest(url: URL(string: "https://covid19.th-stat.com/json/covid19v2/getTodayCases.json")!) request.httpMethod = "GET" request.addValue("application/json", forHTTPHeaderField: "Content-Type") → ข้อมูลที่บรรจุอยู่ในเซิร์ฟเวอร์ ซึ่งข้อมูลนั้นมีรูปแบบเป็น JSON

- เป็นส่วนการทำงานที่ใช้เรียก API เพื่อร้องขอข้อมูลมาแสดงจากเซิร์ฟเวอร์
- เป็นแหล่งข้อมูล
- ร้องขอข้อมูลด้วยวิธีหนึ่ง ในที่นี้จะจัดการกับข้อมูลด้วยการดึงข้อมูลมาอย่างเดียว (= อ่านข้อมูล: GET)
- จะเรียก **Content** มาซึ่งเป็นประเภท JSON แล้ว **addValue** (เพิ่มค่า) เข้าไป โดยค่าค่านั้นเป็น JSON

→ forHTTPHeaderField คือจะดึงข้อมูลทั้งชุดให้ออกมาเป็น 1 ชุดใหญ่ ๆ

การแทนข้อมูล

ในการเรียกข้อมูลจากเซิร์ฟเวอร์ผ่าน API มีส่วนประกอบสำคัญ ดังต่อไปนี้:—

```
import WatchKit
import Foundation

//var Cases: WInterfaceLabel!
//var Recovered: WInterfaceLabel!
//var Deaths: WInterfaceLabel!

class InterfaceController: WInterfaceController {
    // var Int_NewConfirmed: Int
    // var Int_NewRecovered: Int
    // var Int_NewDeaths: Int

    @IBOutlet var lastUpdatedInfo: WInterfaceLabel!
    @IBOutlet var Case_Label: WInterfaceLabel!
    @IBOutlet var Recovered_Label: WInterfaceLabel!
    @IBOutlet var Deaths_Label: WInterfaceLabel!

    override func awake(withContext context: Any?) {
        // Configure interface objects here.
        super.awake(withContext: context)
    }

    override func willActivate() {
        // This method is called when watch view controller is about to be visible to user
        super.willActivate()

        let params = ["username": "", "password": ""] as DictionaryString, String

        var request = URLRequest(url: URL(string: "https://covid19.th-stat.com/json/covid19v2/getTodayCases.json")!)
        request.httpMethod = "GET"
        request.addValue("application/json", forHTTPHeaderField: "Content-Type")

        request.timeoutInterval = 10 // 10 seconds
        request.addValue("application/json", forHTTPHeaderField: "Content-Type")

        let session = URLSession.shared
        let task = session.dataTask(with: request, completionHandler: { data, response, error → Void in
            print(response!)
            do {
                let json = try JSONSerialization.jsonObject(with: data!) as! DictionaryString, AnyObject
                print("JSON Success", json)
                var Result_AllCase: String
                var Result_AllRecovered: String
                var Result_AllDeaths: String
                var Result_AllNewDate: Date()
                let result: [String: Any] = json
                if let All_Case = result["AllCases"] as? Int {
                    Result_AllCase = String(All_Case)
                } else {
                    Result_AllCase = "Cannot Access API"
                }
                if let All_Recovered = result["NewRecovered"] as? Int {
                    Result_AllRecovered = String(All_Recovered)
                } else {
                    Result_AllRecovered = "Cannot Access API"
                }
                if let All_Deaths = result["NewDeaths"] as? Int {
                    Result_AllDeaths = String(All_Deaths)
                } else {
                    Result_AllDeaths = "Cannot Access API"
                }
                if let All_Date = result["UpdateDate"] as? String {
                    Result_AllDate = All_Date
                } else {
                    Result_AllDate = "Cannot Access API"
                }
                let calendar = Calendar.current
                self.Case_Label.setText(Result_AllCase)
                self.Recovered_Label.setText(Result_AllRecovered)
                self.Deaths_Label.setText(Result_AllDeaths)
                self.LastUpdatedInfo.setText(calendar.dateToString(calendar))
            } catch {
                print("error")
                self.Case_Label.setText("x")
                self.Recovered_Label.setText("x")
                self.Deaths_Label.setText("x")
            }
        })
        task.resume()
    }

    override func didDeactivate() {
        super.didDeactivate()
    }
}
```

② ต่อ

`var request = URLRequest(url: URL(string: "https://covid19.th-stat.com/json/covid19v2/getTodayCases.json")!) request.httpMethod = "GET" request.addValue("application/json", forHTTPHeaderField: "Content-Type")`

• การเรียกข้อมูลข้อมูลชุดหนึ่งจาก JSON มา จะต้องระบุเฉพาะว่าจะนำข้อมูลส่วน

Content หรือชุดข้อมูล ที่ API ดึงมาให้

– การดึงนี้จะดึงข้อมูลทั้งชุดออกมาเป็นชุดเดียว 1 ชุดใหญ่ ๆ (จากคำสั่ง

`request.httpMethod="GET" request.addValue("application/json", forHTTPHeaderField: "Content-Type")`)

– ในโค้ดต้องระบุด้วยว่าตัวข้อมูล **Content** มีชนิดชุดข้อมูลใด

(Content-Type = ข้อมูล JSON จะบอกเซิร์ฟเวอร์ว่าข้อมูลนี้เป็นชนิดข้อมูลอะไร เป็นตัวบ่งเซิร์ฟเวอร์ว่าจะทำอะไรกับข้อมูลนี้ได้บ้าง)

⇒ **Content** คือชุดข้อมูลจาก JSON ทั้งหมด

⇒ Type คือประเภทของข้อมูล ในที่นี้ประเภทข้อมูลคือ JSON (ตาม application/json)

HEADER NAME

VALUE

Content-Type

application/json

การแทนข้อมูล

ในการเรียกข้อมูลจากเซิร์ฟเวอร์ผ่าน API มีส่วนประกอบสำคัญ ดังต่อไปนี้:—

```
import WatchKit
import Foundation

//var Cases: WInterfaceLabel!
//var Recovered: WInterfaceLabel!
//var Deaths: WInterfaceLabel!

class InterfaceController: WInterfaceController {
    // var Int_NewConfirmed: Int
    // var Int_NewRecovered: Int
    // var Int_NewDeaths: Int

    @IBOutlet var LastUpdateInfo: WInterfaceLabel!
    @IBOutlet var Case_Label: WInterfaceLabel!
    @IBOutlet var Recovered_Label: WInterfaceLabel!
    @IBOutlet var Deaths_Label: WInterfaceLabel!

    override func awake(withContext context: Any?) {
        // Configure interface objects here.
        super.awake(withContext: context)
    }

    override func willActivate() {
        // This method is called when watch view controller is about to be visible to user
        super.willActivate()
    }

    // let params = ["username": "", "password": ""] as DictionaryString, String

    var request = URLRequest(url: URL(string: "https://covid19-chn.stat.gov/json/weatdinfo/getTodayCases.json")!)
    request.httpMethod = "GET"
    request.httpBody = try? JSONSerialization.data(withJSONObject: params, options: [])
    request.addValue("application/json", forHTTPHeaderField: "Content-Type")
    let session = URLSession.shared
    let task = session.dataTask(with: request, completionHandler: {data, response, error → Void in
        print(response!)
        do {
            let json = try JSONSerialization.jsonObject(with: data!) as! DictionaryString, AnyObject
            print("JSON-response: ", json)
            var Result_AllCase: String
            var Result_AllRecovered: String
            var Result_AllDeaths: String
            var Result_AllDate: Date = Date()
            let result: [String: Any] = json
            if let All_Case = result["NewConfirmed"] as? Int {
                Result_AllCase = String(All_Case)
            } else {
                Result_AllCase = "Cannot Access API"
            }
            if let All_Recovered = result["NewRecovered"] as? Int {
                Result_AllRecovered = String(All_Recovered)
            } else {
                Result_AllRecovered = "Cannot Access API"
            }
            if let All_Deaths = result["NewDeaths"] as? Int {
                Result_AllDeaths = String(All_Deaths)
            } else {
                Result_AllDeaths = "Cannot Access API"
            }
            if let All_Date = result["UpdatedDate"] as? String {
                Result_AllDate = All_Date
            } else {
                Result_AllDate = "Cannot Access API"
            }
            let calendar = Calendar.current
            self.Case_Label.setText(Result_AllCase)
            self.Recovered_Label.setText(Result_AllRecovered)
            self.Deaths_Label.setText(Result_AllDeaths)
            self.LastUpdateInfo.setCalendar(calendar)
        } catch {
            print("error")
            self.Case_Label.setText("x")
            self.Recovered_Label.setText("x")
            self.Deaths_Label.setText("x")
        }
    })
    task.resume()

    override func didDeactivate() {
        super.didDeactivate()
    }
}
```

③

```
let calendar = Calendar.current
self.Case_Label.setText(Result_AllCase)
self.Recovered_Label.setText(Result_AllRecovered)
self.Deaths_Label.setText(Result_AllDeaths)
self.LastUpdateInfo.setCalendar(calendar)
```

• เพื่อให้แสดงวันที่ถูกต้อง ตรงตามปัจจุบัน

• เชื่อม Labels ที่หน้า Interface/UI ให้แสดงค่าตาม `setText(_)` (ให้นำข้อมูลที่ได้ไปแทนใน Labels i.e. `Case_Label`, `Recovered_Label` เป็นต้น)

• รองรับค่าข้อมูลที่จะได้รับมาจากเซิร์ฟเวอร์ผ่าน API ใช้มาแสดงข้อมูลใน Labels ที่ตั้งไว้ (ตัวแปรใหม่)

e.g. `if let All_Recovered = result["NewRecovered"] as?`
`Int {Result_AllRecovered = String(All_Recovered)}`

• เชื่อม Objects (Labels) กับข้อมูลที่ได้มาจากการเรียกร้องขอ (request) ผ่าน API

④

```
task.resume()
```

• ดึงข้อมูลให้แสดงผลบนหน้าจอ

• ถ้าอยู่ดี ๆ โค้ดถูกระงับ จะสั่งให้โค้ดทำงานต่อไป

"NewConfirmed" : 15335,
"NewRecovered" : 6904, ●..... จาก
"NewHospitalized": 8302,
"NewDeaths" : 129
(JSON)

ว่าด้วย Content และ Dictionary

ข้อมูล JSON



```
{  
  "Confirmed"      : 497302,  
  "Recovered"      : 334693,  
  "Hospitalized"   : 158550,  
  "Deaths"         : 4059,  
  "NewConfirmed"   : 15335,  
  "NewRecovered"   : 6904,  
  "NewHospitalized": 8302,  
  "NewDeaths"      : 129,  
  "UpdateDate"     : "25\07\2021 11:39",  
  "DevBy"          : "https://  
                    www.kidkarnmai.com/"  
}
```

↑
เรียกว่า key

↑
เรียกว่า value

รวมกันเป็น dictionary

ทั้งหมดนี้
เรียกว่า
Content

此れは

- Dictionary คือการรวบรวมข้อมูลที่มี key และ value โดย key จะเป็นตัวเลข (Integer) ข้อความ (String) หรือ Object ใด ๆ ก็ได้

e.g.

"กรุงเทพมหานคร" : "Bangkok",
02 : "Chiang Mai",
"ค" : 01

โดย

- ข้อความ (String) จะต้องมี “ ” คั่น
- ต้องมี : คั่นระหว่าง key และ value
- ต้องมี , คั่นทั้ง key กับ value รวมกัน

ผลสำเร็จ

COVID-19 Info 12:56

CASES
TODAY

7/26

ข้อมูลที่นำมาแทนที่แล้ว

Cases

15335

Recovered

6904

Deaths

129

```
var thankYouSlide = [Int:String]()  
    thankYouSlide = [01:“ขอบคุณครับ”,  
                      02:“จัดทำโดย”,  
                      03:“วิศ ศรีปทุมรักษ์ 4/1-9”]  
  
print(thankYouSlide[01]!)  
  
print(thankYouSlide[02]!)  
  
print(thankYouSlide[03]!)
```