SCIENCES SORBONNE UNIVERSITÉ

«MyMeteo»

Fabrice.Kordon@lip6.fr

LIP6

Goal of the example

Weather forecast Apps demystified

- Query to a web service providing such information
- Fetch results of this query
- Decode result
- Display information

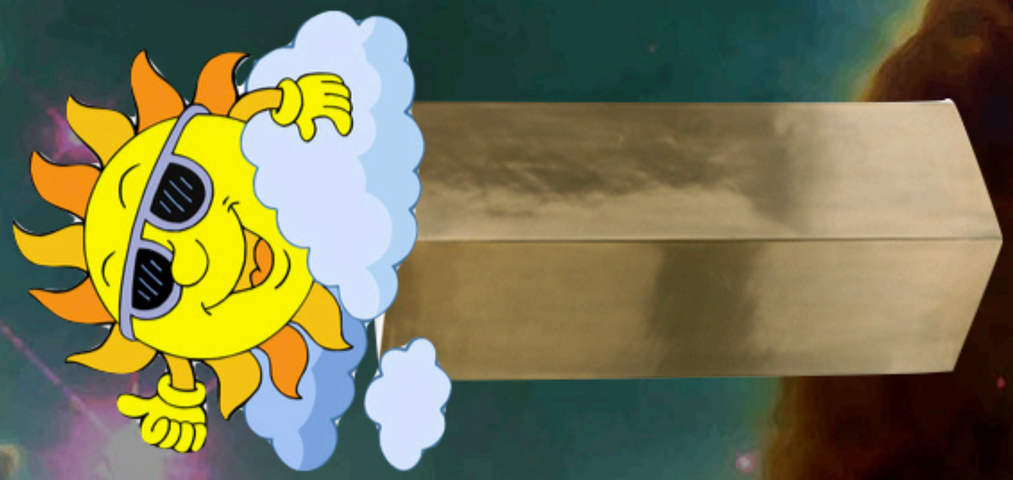
- Weather forecast Apps do the same!
  - **There are so many…**

Goal of the example

Weather forecast Apps demystified

- Query to a web service providing such information
- Fetch results of this query
- Decode result
- Display information

- Weather forecast Apps do the same!
  - **There are so many…**

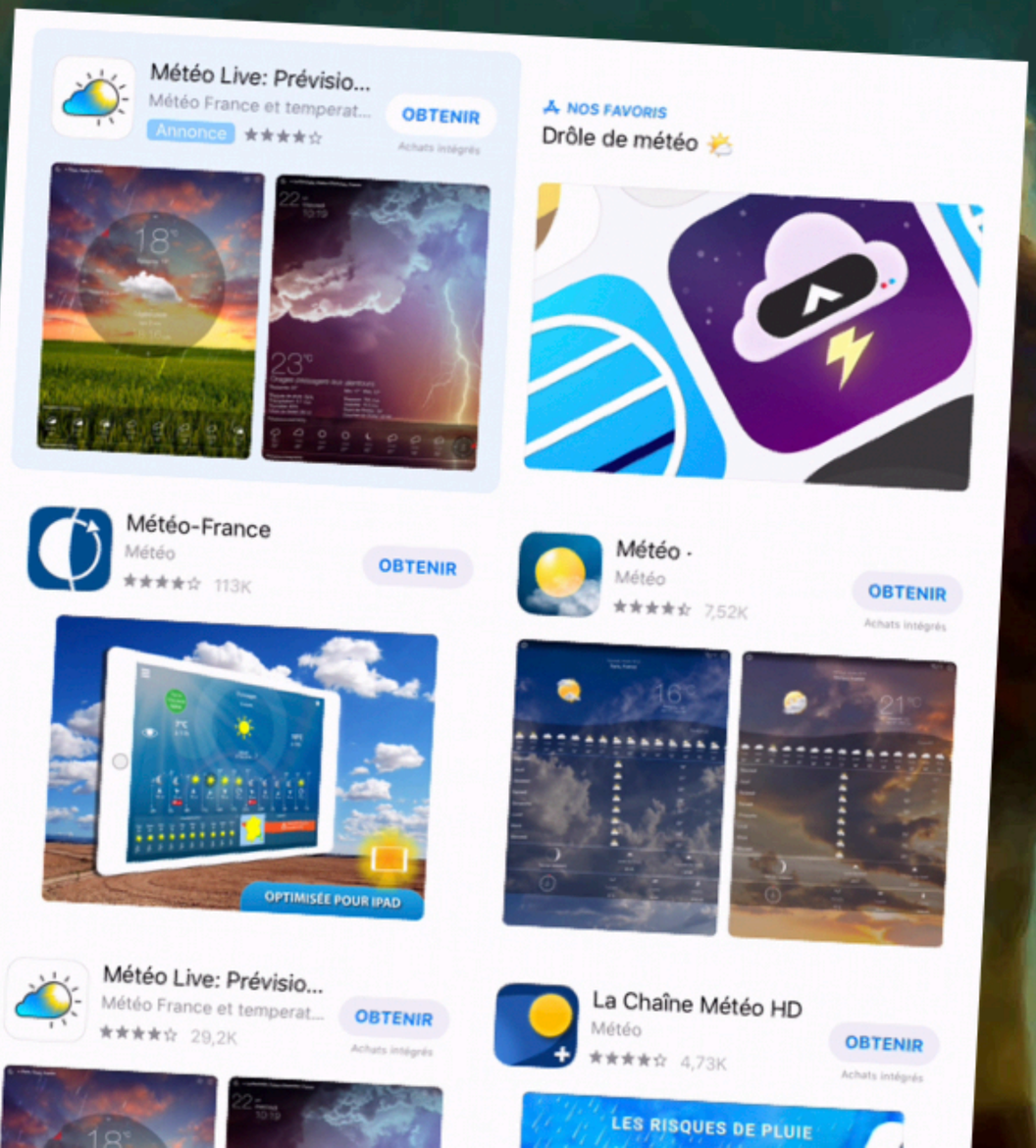


Goal of the example

Weather forecast Apps demystified

- Query to a web service providing such information
- Fetch results of this query
- Decode result
- Display information

- Weather forecast Apps do the same!
  - **There are so many…**

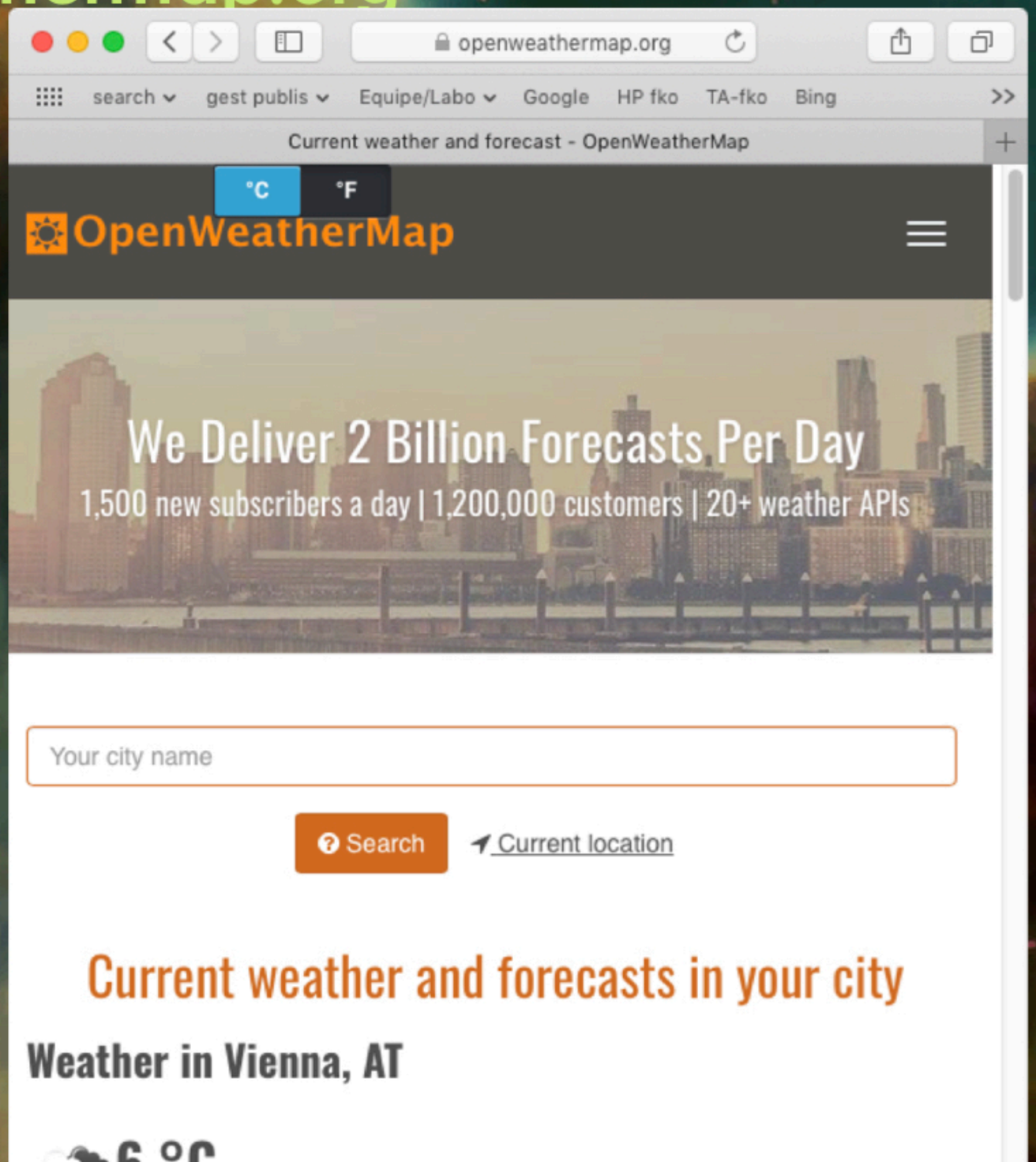


Demo (network deactivated)

Demo

A bit of context

openweathermap.org



A bit of context

openweathermap.org

Structure of a query

- https://api.openweathermap.org/data/2.5/weather?q=<ville>&units=metric&lang=<lang>&APPID=<key>
  - JSON format
  - Use of «split» for simplicity

Sky = 17th field


```
{«coord":{"lon":2.35,"lat":48.86},"weather":[{"id":
500,"main":"Rain","description":"light rain",
"icon":"10d"}],"base":"stations","main":{"temp":8,"pressure":
1005,"humidity":93,"temp_min":7,"temp_max":9},"visibility":
10000,"wind":{"speed":4.1,"deg":50},"clouds":{"all":90},"dt":
1543150800,"sys":{"type":1,"id":5610,"message":
0.0038,"country":"FR","sunrise":1543130107,"sunset":
1543161597},"id":2988507,"name":"Paris","cod":200}
```

A bit of context

openweathermap.org

Structure of a query

- https://api.openweathermap.org/data/2.5/weather?q=<ville>&units=metric&lang=<lang>&APPID=<key>
  - JSON format
  - Use of «split» for simplicity


```
{«coord":{"lon":2.35,"lat":48.86},"weather":[{"id":
500,"main":"Rain","description":"light rain",
"icon":"10d"}],"base":"stations","main":{"temp":8,"pressure":
1005,"humidity":93,"temp_min":7,"temp_max":9},"visibility":
10000,"wind":{"speed":4.1,"deg":50},"clouds":{"all":90},"dt":
1543150800,"sys":{"type":1,"id":5610,"message":
0.0038,"country":"FR","sunrise":1543130107,"sunset":
1543161597},"id":2988507,"name":"Paris","cod":200}
```


Temp = 30th field

A bit of context

openweathermap.org

Structure of a query

- https://api.openweathermap.org/data/2.5/weather?q=<ville>&units=metric&lang=<lang>&APPID=<key>
  - JSON format
  - Use of «split» for simplicity Wind = 44th field


```
{«coord":{"lon":2.35,"lat":48.86},"weather":[{"id":
500,"main":"Rain","description":"light rain",
"icon":"10d"}],"base":"stations","main":{"temp":8,"pressure":
1005,"humidity":93,"temp_min":7,"temp_max":9},"visibility":
10000,"wind":{"speed":4.1,"deg":50},"clouds":{"all":90},"dt":
1543150800,"sys":{"type":1,"id":5610,"message":
0.0038,"country":"FR","sunrise":1543130107,"sunset":
1543161597},"id":2988507,"name":"Paris","cod":200}
```

View Controller

Sake of simplicity…

Code located in a ViewController

View Controller


```
import UIKit

class ViewController: UIViewController, UIPickerViewDelegate, UIPickerViewDataSource {

    private let cities = ["Amsterdam", "Berlin", "Budapest", "London",
                          "Madrid", "Milano", "Montreal", "Paris",
                          "Roma", "Stockholm","Sydney", "Tokyo"]


    private let cityLabel = UILabel()
    private let skyLabel = UILabel()
    private let tempLabel = UILabel()
    private let windLabel = UILabel()
    private let b = UIButton(type: .system)

    private let cityPicker = UIPickerView()
    private var currentSelection : Int = 0
    private let sky = UIImageView(image: UIImage(named: "background"))
```

View Controller


```
override func viewDidLoad() {
  super.viewDidLoad()
  let f = UIScreen.main.bounds
  self.view = UIView(frame: f)
  self.view.addSubview(sky)
  sky.center = CGPoint(x: f.size.width / 2, y: f.size.height / 2)

  self.view.addSubview(cityLabel)
  cityLabel.frame = CGRect(x: f.size.width / 2 - 130, y: 70, width: 260, height: 30)
  cityLabel.font = UIFont.systemFont(ofSize: 16)
  cityLabel.textAlignment = .center
  cityLabel.text = cities[currentSelection]

  self.view.addSubview(skyLabel)
  skyLabel.frame = CGRect(x: f.size.width / 2 - 130, y: 110, width: 260, height: 30)
  skyLabel.font = UIFont.systemFont(ofSize: 16)
  skyLabel.textAlignment = .center
  skyLabel.text = "---"

  self.view.addSubview(tempLabel)
  tempLabel.frame = CGRect(x: f.size.width / 2 - 130, y: 150, width: 260, height: 30)
  tempLabel.font = UIFont.systemFont(ofSize: 16)
  tempLabel.textAlignment = .center
  tempLabel.text = "---"

  self.view.addSubview(windLabel)
  windLabel.frame = CGRect(x: f.size.width / 2 - 130, y: 190, width: 260, height: 30)
  windLabel.font = UIFont.systemFont(ofSize: 16)
  windLabel.textAlignment = .center
  windLabel.text = "---"

  self.view.addSubview(b)
  b.frame = CGRect(x: f.size.width / 2 - 130, y: 230,
                   width: 260, height: 30)
  b.tintColor = .white
  b.setTitle("FecthWeather", for: .normal)
  b.addTarget(self, action: #selector(fetchData), for: .touchDown)

  self.view.addSubview(cityPicker)
  cityPicker.frame = CGRect(x: 20, y: 270, width: f.size.width - 40,
                            height: f.size.height - 270)
  cityPicker.selectRow(0, inComponent: 0, animated: false)
  cityPicker.dataSource = self
  cityPicker.delegate = self
}
```

View Controller


```
    @objc func fetchData () {
        let computed = "https://api.openweathermap.org/data/2.5/weather?q=\
(cities[currentSelection])&units=metric&lang=en&APPID=<KEY>"
        let myURL = URL(string: computed)
        if myURL != nil {
            let request = URLRequest(url: myURL!)
            let config = URLSessionConfiguration.default
            let session = URLSession(configuration: config)
            let task = session.dataTask(with: request) {(data, resp, err) in
                DispatchQueue.main.async {
                    if err != nil {
                        let a = UIAlertController(title: "Error",
                                                  message: err!.localizedDescription,
                                                  preferredStyle: .alert)
                        a.addAction(UIAlertAction(title: "OK",
                                                  style: .default, handler: nil))
                        self.present(a, animated: true, completion: nil)
                    } else if data != nil {
                        self.handleData(data!)
                    } else {
                        self.cityLabel.text = "empty data"
                    }
                }
            }
            task.resume()
            b.isHidden = true
            cityPicker.isHidden = true
        } else {
            cityLabel.text = "URL failure!!!"

        }
}
```

View Controller


```
func handleData(_ d: Data) {
    // Note, this is JSON format but structure is complex
    // so we do a big ugly trick just to show you how
    // network works.
    let s = String(decoding: d, as: UTF8.self)
    let decomposed = s.components(separatedBy: "\"")
    skyLabel.text =  "Sky : \(decomposed[17])"
    let s1 = decomposed[30].components(separatedBy: ":")
    let s2 = s1[1].components(separatedBy: ",")
    tempLabel.text = "Temp : \(s2[0]) °C"
    let s3 = decomposed[44].components(separatedBy: ":")
    let s4 = s3[1].components(separatedBy: ",")
    windLabel.text = "Wind : \(s4[0]) m/s"
    b.isHidden = false
    cityPicker.isHidden = false
}
```

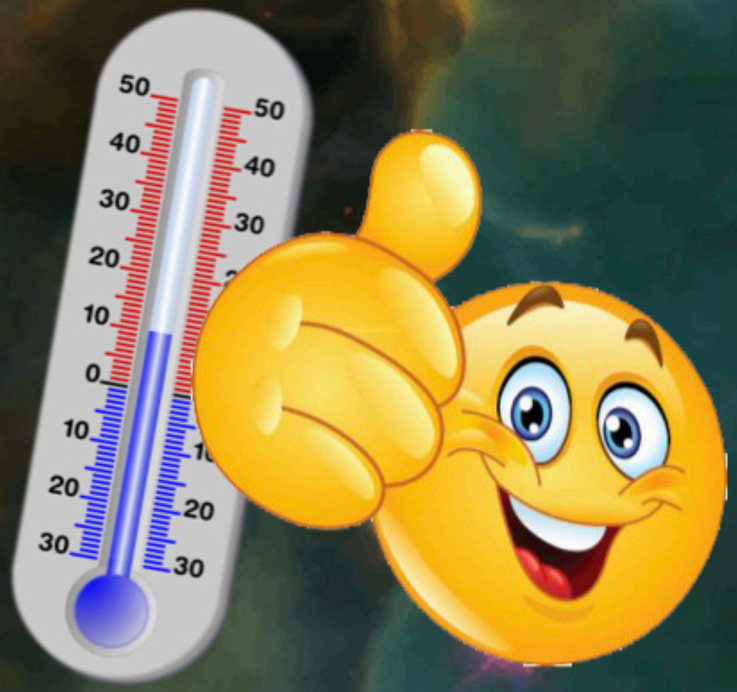
View Controller


```
    // For the UIPickerView

    func pickerView(_ pickerView: UIPickerView,
                    titleForRow row: Int,
                    forComponent component: Int) -> String? {
        return cities[row]
    }

    func pickerView(_ pickerView: UIPickerView,
                    didSelectRow row: Int,
                    inComponent component: Int) {
        currentSelection = row
        cityLabel.text = cities[currentSelection]
        skyLabel.text = "---"
        tempLabel.text = "---"
        windLabel.text = "---"
    }

    // UIPickerViewDataSource protocol

    func numberOfComponents(in pickerView: UIPickerView) -> Int {
        return 1
    }

    func pickerView(_ pickerView: UIPickerView,
                    numberOfRowsInComponent component: Int) -> Int {
        return cities.count
    }
}
```

As a conclusion…

You can do as good as Apple does

- Very nice forecast application
  - **But its heart is networking + nice animations**



In case you doubted…
… Network is important!

- But XML/JSON are useful too…