

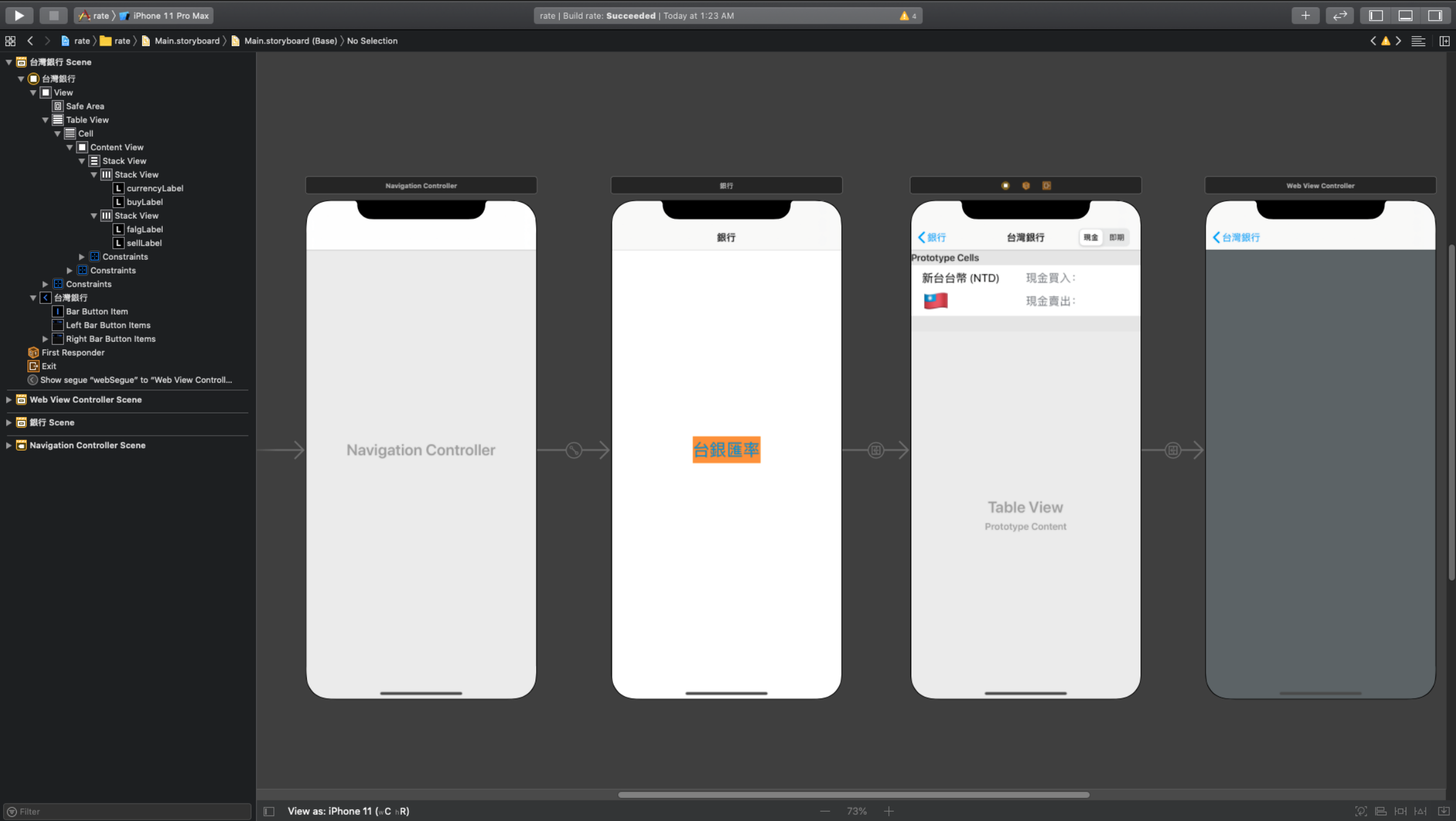


匯率查詢

智慧型行動裝置軟體設計

- ▶ 想要製作一款方便查詢匯率的 App
- ▶ 就這樣

程式架構



程式架構

```
var flag = ["USD": "🇺🇸", "HKD": "🇭🇰", "GBP": "🇬🇧", "AUD": "🇦🇺",  
            "CAD": "🇨🇦", "SGD": "🇸🇬", "CHF": "🇨🇭", "JPY": "🇯🇵", "ZAR":  
            "🇿🇦", "SEK": "🇸🇪", "NZD": "🇳🇿", "THB": "🇹🇭", "PHP": "🇵🇭",  
            "IDR": "🇮🇩", "EUR": "🇪🇺", "KRW": "🇰🇷", "VND": "🇻🇳", "MYR":  
            "🇲🇾", "CNY": "🇨🇳"]  
var currency = [String]()  
var currency_EN = [String]()  
var buy = [[String]]()  
var sell = [[String]]()
```

定義國旗emoji與待下載變數

```
override func viewDidLoad() {  
    super.viewDidLoad()  
    let url = "https://rate.bot.com.tw/xrt?Lang=zh-TW"  
    Alamofire.request(url).responseString { response in  
        if let html = response.result.value {  
            self.parseTaiwanBankHTML(html: html)  
  
            self.tableView.estimatedRowHeight = 91;  
            self.tableView.endUpdates()  
            self.tableView.reloadData()  
        }  
    }  
}
```

進入畫面->下載html->解析html->重整

程式架構

```
func parseTaiwanBankHTML(html: String) {  
    print("Alamofire")  
  
    do{  
        if let doc = try? Kanna.HTML(html: html, encoding: String.Encoding.utf8) {  
  
            for i in 1...19 {  
  
                var str =  
                    doc.xpath("/html/body/div[1]/main/div[3]/table/tbody/tr[\(i)]/td[1]/div/div  
                        [2]").first!.text!.trimmingCharacters(in: .whitespacesAndNewlines)  
                currency.append(str)  
                str = String(str.suffix(4))  
                str = String(str.prefix(3))  
                currency_EN.append(str)  
  
                buy.append([doc.xpath("/html/body/div[1]/main/div[3]/table/tbody/tr[\(i)]/td  
                    [2]").first!.text!.trimmingCharacters(in: .whitespacesAndNewlines),  
                    doc.xpath("/html/body/div[1]/main/div[3]/table/tbody/tr[\(i)]/td  
                    [4]").first!.text!.trimmingCharacters(in: .whitespacesAndNewlines)])  
  
                sell.append([doc.xpath("/html/body/div[1]/main/div[3]/table/tbody/tr[\(i)]/td  
                    [3]").first!.text!.trimmingCharacters(in: .whitespacesAndNewlines),  
                    doc.xpath("/html/body/div[1]/main/div[3]/table/tbody/tr[\(i)]/td  
                    [5]").first!.text!.trimmingCharacters(in: .whitespacesAndNewlines)])  
  
            }  
        }  
    }  
}
```

解析html並存到陣列

程式架構



```
@IBAction func mySegmentedAction(_ sender: UISegmentedControl) {  
    if sender.selectedSegmentIndex == 0 {  
        cashspot = 0  
        self.tableView.reloadData()  
    }else {  
        cashspot = 1  
        self.tableView.reloadData()  
    }  
}
```

設定UISegmentedControl

程式架構

```
func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {
    return currency_EN.count
}

func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {
    let cellIdentifier = "Cell"
    let cell = tableView.dequeueReusableCell(withIdentifier: cellIdentifier, for: indexPath) as! RateTableViewCell
    cell.currencyLabel.text = currency[indexPath.row]
    cell.flagLabel.text = flag[currency_EN[indexPath.row]]!
    var str: String
    if cashspot == 0 {
        str = "現金"
    } else {
        str = "即期"
    }
    cell.buyLabel.text = str + "買入 : " + buy[Int(indexPath.row)][cashspot]
    cell.sellLabel.text = str + "賣出 : " + sell[Int(indexPath.row)][cashspot]
    return cell
}
```

設定UITableViewCell

程式架構

```
override func prepare(for segue: UIStoryboardSegue, sender: Any?) {  
    let page2 = segue.destination as! WebViewController  
    chartURL = _chartURL + currency_EN[tableView.indexPathForSelectedRow!.row] + "?Lang=zh-TW"  
    tableView.deselectRow(at: (tableView.indexPathForSelectedRow)!, animated: false)  
    page2.url = chartURL  
}
```

換頁傳值

程式架構

```
var url : String = ""
var mytitle = "歷史匯率"

@IBOutlet weak var webView: WKWebView!
@IBOutlet var label: UILabel!

override func viewDidLoad() {
    super.viewDidLoad()

    title = mytitle

    let request = URLRequest(url: URL(string: url)!)
    webView?.load(request)
    // Do any additional setup after loading the view.
}
```

新頁面瀏覽歷史匯率

成果

執行畫面

