中華航空股份有限公司

iOS App 開發教育訓練課程

Part 5

Zencher Co. Ltd 2018 March

XML & HTML Parser

Michael Pan

So 什麼是 HTML

```
<html>
  <body>
    <h1> Hi Michael </h1>
  </body>
</html>
```

存成 .html

由 Browser 讀取

Hi Michael

On line html editor

http://rendera.heroku.com

http://htmlparser.appspot.com/samples/RenderToText.jsp

http://scratchpad.io/forgetful-push-8078

tag 標籤 element 元素

<html>

<body>

<h1>

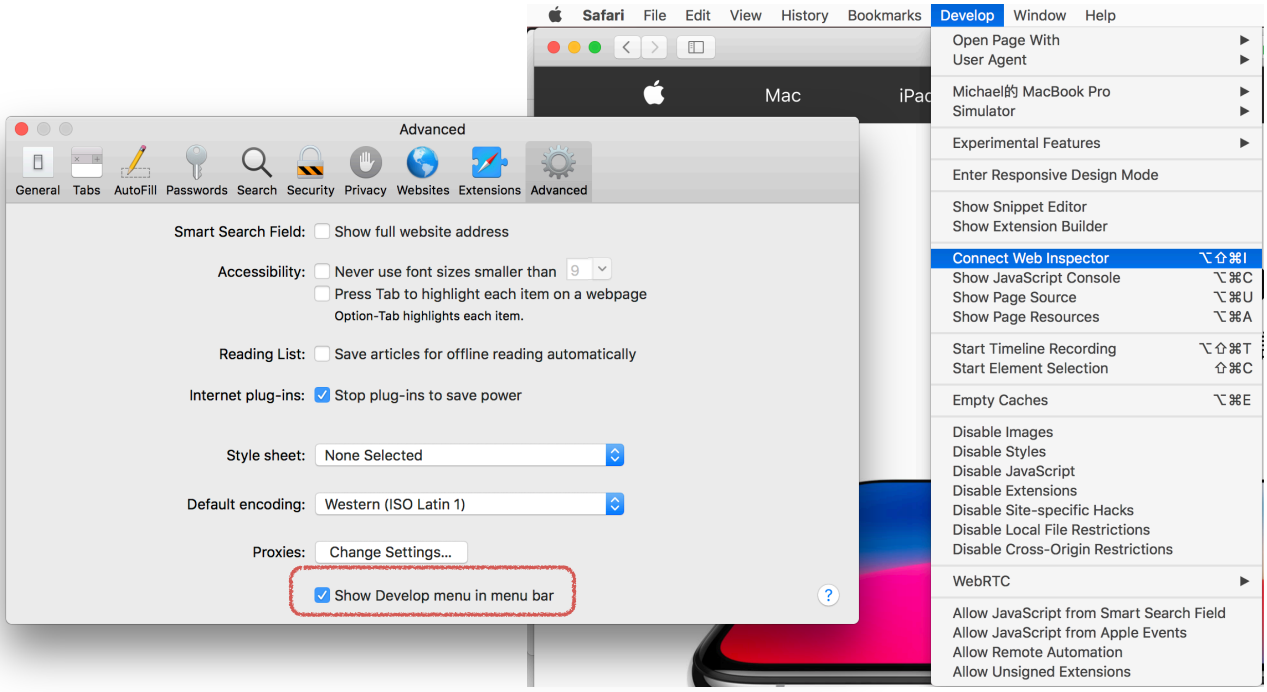
每個 tag 都有特別的用意，用來呈現網頁的內容

Table

```
<table style="width:300px">
<tr>
  <td>Jill</td>
  <td>Smith</td>
  <td>50</td>
</tr>
<tr>
  <td>Eve</td>
  <td>Jackson</td>
  <td>94</td>
</tr>
</table>
```

Jill	Smith	50
Eve	Jackson	94

開發者工具 - Safari



Inspector

開發者工具 - Chrome

線上學習

Codecademy

http://www.codecademy.com/tracks/web?jump_to=5024844597a4040002069e67

書本

http://www.books.com.tw/products/0010359684

XML & HTML

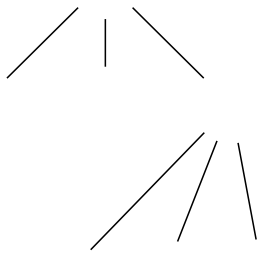
XML 是自訂的 Tag

可以任意寫，自訂意義

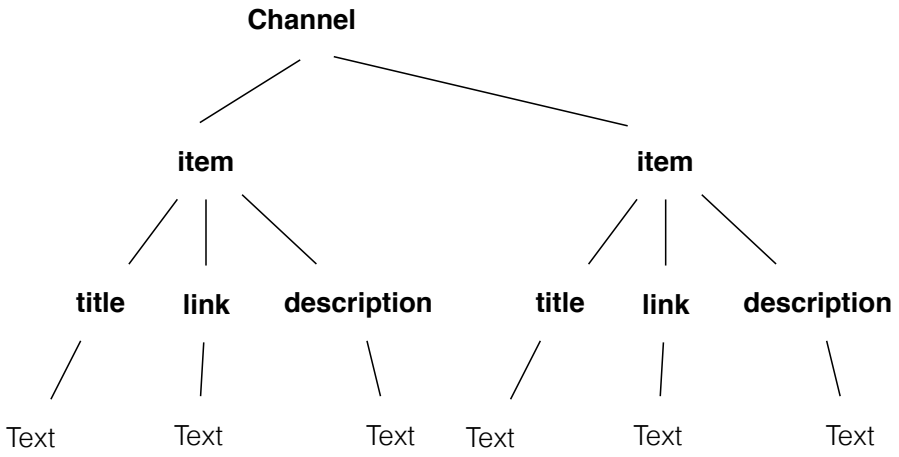
廣義的說 HTML 是 XML 的子集合

XML

```
<?xml version="1.0" encoding="UTF-8" ?>
<rss version="2.0">
<channel>
  <item>
    <title>W3Schools Home Page</title>
    <link>http://www.w3schools.com</link>
    <description>Free web building tutorials</description>
  </item>
  <item>
    <title>RSS Tutorial</title>
    <link>http://www.w3schools.com/rss</link>
    <description>New RSS tutorial on W3Schools</description>
  </item>
  <item>
    <title>XML Tutorial</title>
    <link>http://www.w3schools.com/xml</link>
    <description>New XML tutorial on W3Schools</description>
  </item>
</channel>
</rss>
```



More details - tree



如何找到某一個元素

XPath

For example

/rss/channel/item

```
<?xml version="1.0" encoding="UTF-8" ?>
<rss version="2.0">
 <channel>
  <item>
    <title>W3Schools Home Page</title>
    <link>http://www.w3schools.com</link>
    <description>Free web building tutorials</description>
  </item>
  <item>
    <title>RSS Tutorial</title>
    <link>http://www.w3schools.com/rss</link>
    <description>New RSS tutorial on W3Schools</description>
  </item>
  <item>
    <title>XML Tutorial</title>
    <link>http://www.w3schools.com/xml</link>
    <description>New XML tutorial on W3Schools</description>
  </item>
 </channel>
</rss>
```

如何找到某一個元素 - More

//item/title

找到多個 **title**

```
<?xml version="1.0" encoding="UTF-8" ?>
<rss version="2.0">
 <channel>
   <item>
     <title>W3Schools Home Page</title>
     <link>http://www.w3schools.com</link>
     <description>Free web building tutorials</description>
   </item>
   <item>
     <title>RSS Tutorial</title>
     <link>http://www.w3schools.com/rss</link>
     <description>New RSS tutorial on W3Schools</description>
   </item>
   <item>
     <title>XML Tutorial</title>
     <link>http://www.w3schools.com/xml</link>
     <description>New XML tutorial on W3Schools</description>
   </item>
 </channel>
</rss>
```

XML or HTML Parse

將文字轉成記憶體裡的資料結構

```
<item>
    <title>W3Schools Home Page</title>
    <link>http://www.w3schools.com</link>
    <description>Free web building tutorials</description>
</item>
```

```
{ "item": {"title":"W3Schools Home Page",
           "link":"http://www.w3schools.com",
           "description":"Free web building tutorials"
         }
}
```

We need 3rd party Parser

for XML and HTML

HTML 是很鬆散的 XML

Browser 可以看，但一般 Parser 不能解析也不要感覺意外

我覺得**這個**很厲害

Kanna

http://tid-kijyun.github.io/Kanna/

MIT LICENSE

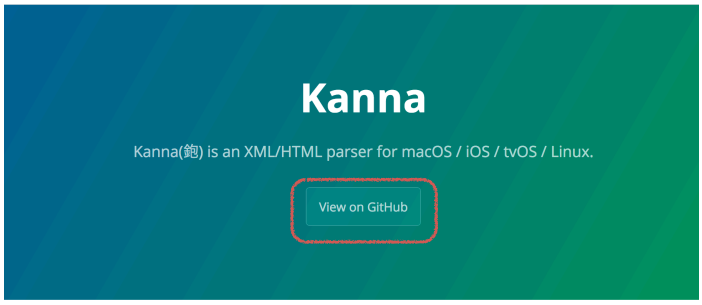
Swift 4

CocoaPods

⚠️ **CocoaPods (`1.1.0 or later`) is required.**

Adding it to your `Podfile` :

```
use_frameworks!
pod 'Kanna', '~> 4.0.0'
```



Tutorials for Swift 3.x

1. Installing Kanna
2. Parsing an HTML/XML document
3. Searching a XML/HTML document
4. Modifying an HTML/XML document

Example - HTML

```
import Kanna
let html = "<html>...</html>"
if let doc = try? HTML(html: html, encoding: .utf8) {
    print(doc.title)

    // Search for nodes by CSS
    for link in doc.css("a, link") {
        print(link.text)
        print(link["href"])
    }
    // Search for nodes by XPath
    for link in doc.xpath("//a | //link") {
        print(link.text)
        print(link["href"])
    }
}
```

Example - XML

```
let xml = "..."
if let doc = try? Kanna.XML(xml: xml, encoding: .utf8) {
    let namespaces = [
        "o":  "urn:schemas-microsoft-com:office:office",
        "ss": "urn:schemas-microsoft-com:office:spreadsheet"
    ]
    if let author = doc.at_xpath("//o:Author", namespaces: namespaces) {
        print(author.text)
    }
}
```

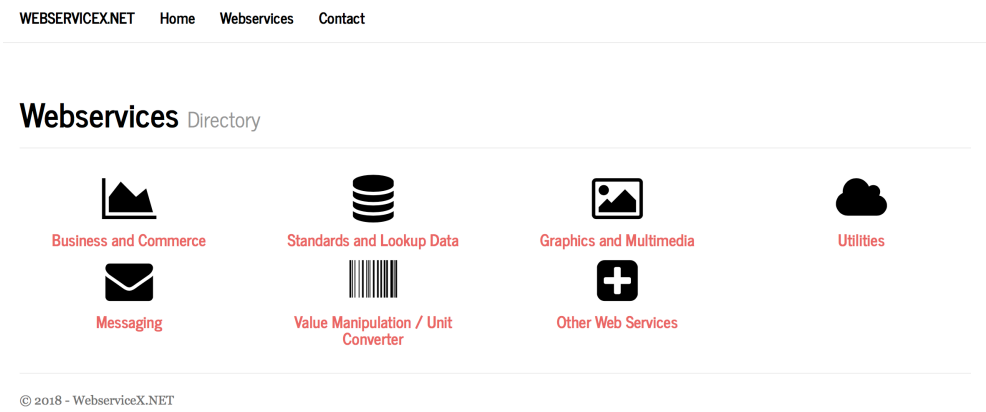
Run

Web Service

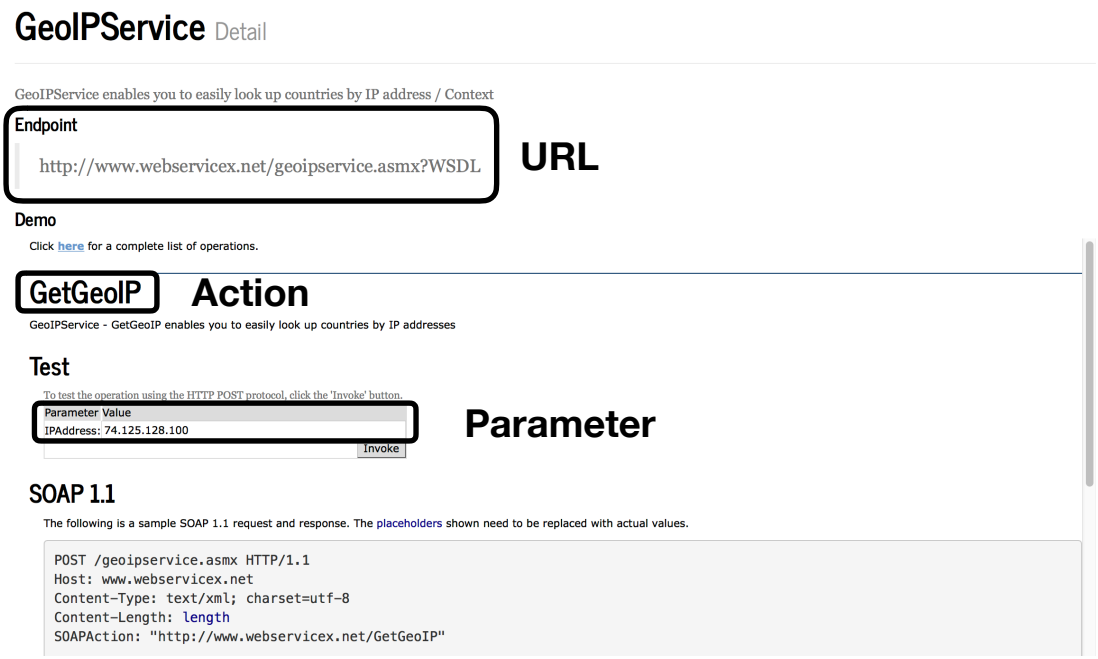
Michael Pan

Web Service

http://www.webservicex.net/New/Home/Directory



Web Service



Web Service

Test

IPAddress: 74.125.128.100

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
▼<GeoIP xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://www.webservicex.net/">
   <ReturnCode>1</ReturnCode>
   <IP>74.125.128.100</IP>
   <ReturnCodeDetails>Success</ReturnCodeDetails>
   <CountryName>United States</CountryName>
   <CountryCode>USA</CountryCode>
 </GeoIP>
```

Web Service

Request

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```
POST /geoipservice.asmx HTTP/1.1
Host: www.webservicex.net
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap1
2="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <GetGeoIP xmlns="http://www.webservicex.net/">
      <IPAddress>string</IPAddress>
    </GetGeoIP>
  </soap12:Body>
</soap12:Envelope>
```

Namespace

Web Service

Response

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap1
2="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <GetGeoIPResponse xmlns="http://www.webservicex.net/">
      <GetGeoIPResult>
        <ReturnCode>int</ReturnCode>
        <IP>string</IP>
        <ReturnCodeDetails>string</ReturnCodeDetails>
        <CountryName>string</CountryName>
        <CountryCode>string</CountryCode>
      </GetGeoIPResult>
    </GetGeoIPResponse>
  </soap12:Body>
</soap12:Envelope>
```

Alamofire

https://github.com/Alamofire/Alamofire

Alamofire - cocoapods

pod init

Edit podfile

```
pod 'Alamofire', '~> 4.7'
```

Pod install

Import Alamofire

In ViewController.swift

```
import UIKit
import Alamofire

class ViewController: UIViewController {

……
```

toSoapMessage

In ViewController.swift

```
func toSoapMessage12(methodName: String, paramValues: String, namespace: String) -> String
{
        var message: String = String()
        message += "<?xml version=\"1.0\" encoding=\"utf-8\"?>"
        message += "<soap12:Envelope xmlns:xsi=\"http://www.w3.org/2001/XMLSchema-
instance\" xmlns:xsd=\"http://www.w3.org/2001/XMLSchema\" xmlns:soap12=\"http://
www.w3.org/2003/05/soap-envelope\">"
        message += "<soap12:Body>"
        message += "<\(methodName) xmlns=\"\(namespace)\">"
        message += "\(paramValues)"
        message += "</\(methodName)>"
        message += "</soap12:Body>"
        message += "</soap12:Envelope>"
        return message
    }
```

```
POST /geoipservice.asmx HTTP/1.1
Host: www.webservicex.net
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <GetGeoIP xmlns="http://www.webservicex.net/">
      <IPAddress>string</IPAddress>
    </GetGeoIP>
  </soap12:Body>
</soap12:Envelope>
```

soapRequest

In ViewController.swift

```
func soapRequest12(url:String, parameter:Dictionary<String, String>, methodName:String , namespace:String) -> URLRequest {
        let URL = NSURL(string: url)!
        var urlRequest: URLRequest = URLRequest(url:URL as URL)
        var rootCodeStr = ""

        if parameter != ["":""] {
            for dic in parameter {
                rootCodeStr += "<"
                rootCodeStr += dic.key
                rootCodeStr += ">"
                rootCodeStr += dic.value
                rootCodeStr += "</"
                rootCodeStr += dic.key
                rootCodeStr += ">"
            }
        }
        let soapMsg: String = toSoapMessage12(methodName: methodName, paramValues: rootCodeStr, namespace: namespace)
        urlRequest.setValue("application/soap+xml; charset=utf-8", forHTTPHeaderField: "Content-Type")
        //        mutableURLRequest.setValue(action, forHTTPHeaderField: "SOAPAction")
        urlRequest.setValue(String(soapMsg), forHTTPHeaderField: "Content-Length")
        urlRequest.httpMethod = "POST"
        urlRequest.httpBody = soapMsg.data(using: String.Encoding.utf8)
        print("soapMsg: ", soapMsg)
        return urlRequest
    }
```

parameter

```
POST /geoipservice.asmx HTTP/1.1
Host: www.webservicex.net
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <GetGeoIP xmlns="http://www.webservicex.net/">
      <IPAddress>string</IPAddress>
    </GetGeoIP>
  </soap12:Body>
</soap12:Envelope>
```

getCountryByIP

In ViewController.swift

```
func getCountryByIP() {

    let parameters = [“IPAddress":"74.125.128.100"]

    let urlRequest = soapRequest12(url: "http://www.webservicex.net/geoipservice.asmx",
                                   parameter: parameters,
                                   methodName: "GetGeoIP",
                                   namespace: "http://www.webservicex.net/")

    Alamofire.request(urlRequest).response { (response) in

        if let data = response.data, let utf8Text = String(data: data,
                                                           encoding: .utf8) {
            print("Data: \(utf8Text)")
        }
    }
}
```

getCountryByIP

In ViewController.swift

```
override func viewDidLoad() {
    super.viewDidLoad()

    getCountryByIP()

}
```

App Transport Security

In Info.plist (Source code)

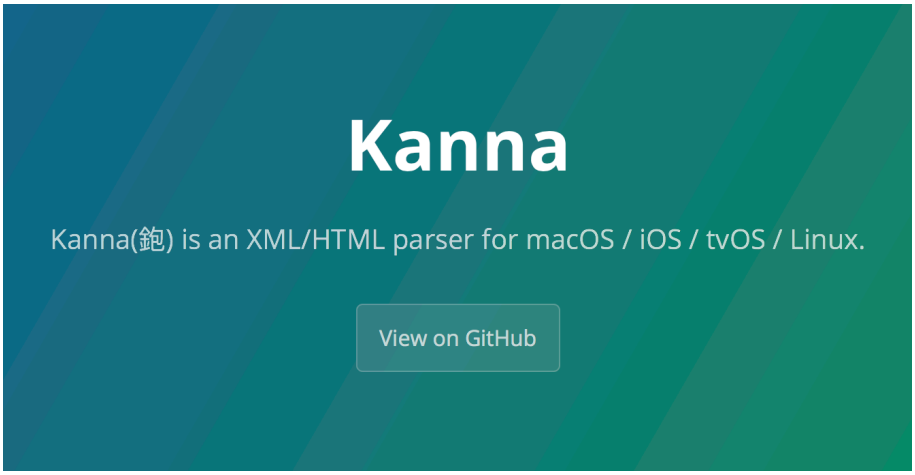
<key>NSAppTransportSecurity</key>
<dict>
<key>NSAllowsArbitraryLoads</key><true/>
</dict>

Run

```
Data: <?xml version="1.0" encoding="utf-8"?><soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/
soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/
2001/XMLSchema"><soap:Body><GetGeoIPResponse xmlns="http://
www.webservicex.net/"><GetGeoIPResult><ReturnCode>1</ReturnCode><IP>74.125.128.100</
IP><ReturnCodeDetails>Success</ReturnCodeDetails><CountryName>United States</
CountryName><CountryCode>USA</CountryCode></GetGeoIPResult></GetGeoIPResponse></soap:Body></
soap:Envelope>
```

XML Parser- Kanna

Kanna



Cocoapods

pod init

Modify Podfile

```
pod 'Kanna', '~> 4.0.0'
```

pod install

Import Kanna

In ViewController.swift

```
import UIKit
import Alamofire
import Kanna

class ViewController: UIViewController {

……
```

WebService- getRankCode

http://tpeweb02.china-airlines.com/webfz/services/ListbyFdateMainService?wsdl

```
<xs:element name="getRankCode">
  <xs:complexType>
    <xs:sequence/>
  </xs:complexType>
</xs:element>
<xs:element name="getRankCodeResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element maxOccurs="unbounded" minOccurs="0" name="return" nillable="true" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Action

WebService- getRankCode

http://192.168.10.127/GetBinaryFile.asmx

```
<wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:ns1="http://org.apache.axis2/xsd" xmlns:ns="http://MsgFromGrd.ws"
xmlns:ax27="http://MsgFromGrd.ws/xsd"
xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
targetNamespace="http://MsgFromGrd.ws">
```

Namespace

WebService- getRankCode

http://192.168.10.127/GetBinaryFile.asmx

```
▼<wsdl:service name="ListbyFdateMainService">
 ▼<wsdl:port name="ListbyFdateMainServiceHttpSoap11Endpoint"
  binding="ns:ListbyFdateMainServiceSoap11Binding">
    <soap:address location="http://tpeweb02.china-
    airlines.com/webfz/services/ListbyFdateMainService.ListbyFdateMainServiceHttpSoap11Endpoint/"/>
  </wsdl:port>
 ▼<wsdl:port name="ListbyFdateMainServiceHttpSoap12Endpoint"
  binding="ns:ListbyFdateMainServiceSoap12Binding">
    <soap12:address location="http://tpeweb02.china-
    airlines.com/webfz/services/ListbyFdateMainService.ListbyFdateMainServiceHttpSoap12Endpoint/"/>
  </wsdl:port>
 ▼<wsdl:port name="ListbyFdateMainServiceHttpEndpoint" binding="ns:ListbyFdateMainServiceHttpBinding">
    <http:address location="http://tpeweb02.china-
    airlines.com/webfz/services/ListbyFdateMainService.ListbyFdateMainServiceHttpEndpoint/"/>
  </wsdl:port>
</wsdl:service>
```

URL

WebService- getRankCode

In ViewController.swift

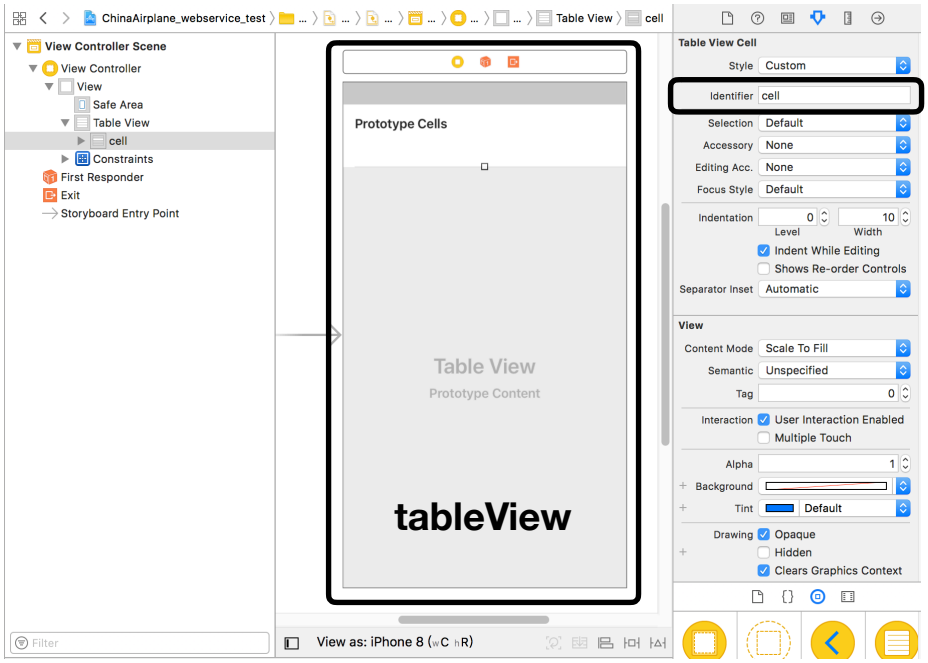
```
func getRankCode() {

    let parameters = ["":""]

    let urlRequest = soapRequest12(url: "http://tpeweb02.china-airlines.com/webfz/services/
                        ListbyFdateMainService.ListbyFdateMainServiceHttpSoap12Endpoint/",
                           parameter: parameters,
                          methodName: "getRankCode",
                           namespace: "http://MsgFromGrd.ws")

    Alamofire.request(urlRequest).response { (response) in

        if let data = response.data, let utf8Text = String(data: data, encoding: .utf8) {

            print("Data: \(utf8Text)")

            self.xml = utf8Text
        }
    }
}
```

WebService- getRankCode

In ViewController.swift

```
override func viewDidLoad() {
    super.viewDidLoad()

    getRankCode()

}
```

WebService- tableView



Identifier

WebService- tableView

In ViewController.swift

```
class ViewController: UIViewController {

    @IBOutlet weak var tableView: UITableView!

    var xml = “"

    var rankCodeArr:[String] = []

    …
}
```

WebService- tableView

In ViewController.swift

```
extension ViewController: UITableViewDelegate, UITableViewDataSource {

    func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {

        let cell:UITableViewCell = tableView.dequeueReusableCell(withIdentifier: "cell",
                                                                 for: indexPath)

        cell.textLabel?.text = rankCodeArr[indexPath.row]

        return cell
    }

    public func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {

        return rankCodeArr.count
    }
}
```

WebService- getRankCode

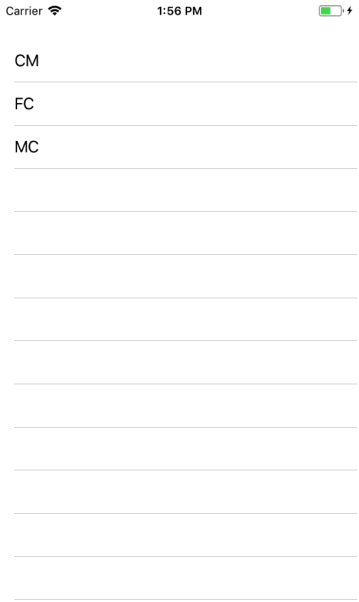
In ViewController.swift

```
func getRankCode() {

    ……

    Alamofire.request(urlRequest).response { (response) in

        if let data = response.data, let utf8Text = String(data: data, encoding: .utf8) {

            print("Data: \(utf8Text)")

            self.xml = utf8Text

            do {
                let doc = try Kanna.XML(xml: self.xml, encoding: .utf8)
                print("Doc: \(doc)")

                for node in doc.xpath("//w:return", namespaces: ["w": "http://MsgFromGrd.ws"]) {
                    print(node.text!)

                    self.rankCodeArr.append(node.text!)
                }
                print(self.rankCodeArr)
                self.tableView.reloadData()

            } catch {
                print("Error")
            }
        }
    }
}
```

Run

["CM", "FC", "MC"]

Run



Upload zip

Michael

Zip & Alamofire

Zip

Alamofire

Zip & Alamofire - cocoapods

pod init

Edit podfile

```
pod 'Zip', '~> 1.1'
pod 'Alamofire', '~> 4.7'
```

pod install

Import Zip & Alamofire

In ViewController.swift

```
import UIKit
import Zip
import Alamofire

class ViewController: UIViewController {

……
```

App Transport Security

In Info.plist (Source code)

```
<key>NSAppTransportSecurity</key>
<dict>
<key>NSAllowsArbitraryLoads</key><true/>
</dict>
```

Zip

In ViewController.swift

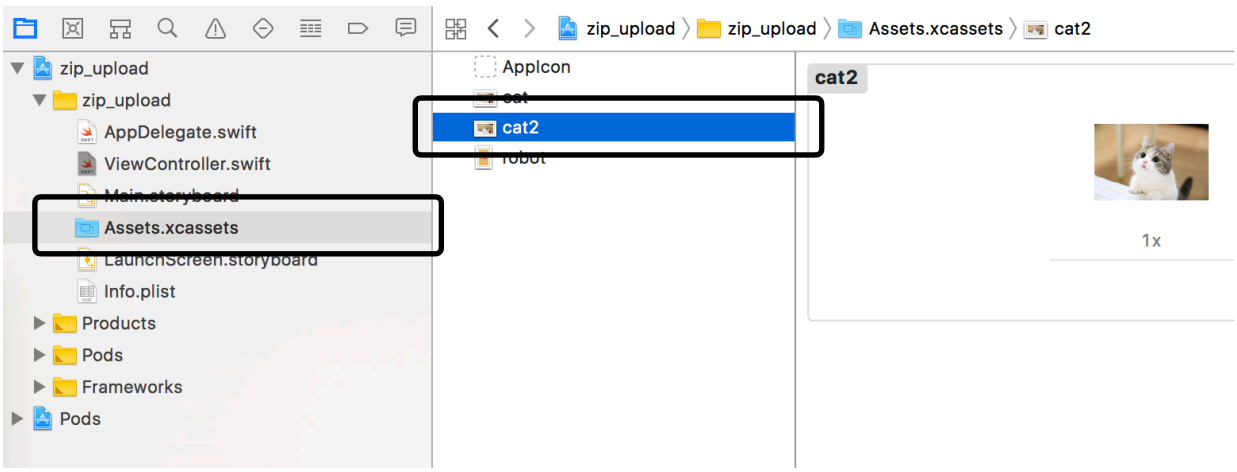
```
func zipTheCatPhoto(imgName: String, zipName: String) {

        let image = UIImage(named:imgName)
        let imageData: Data = UIImageJPEGRepresentation(image!, 1.0)!

        let docUrl = fileManager.urls(for: .documentDirectory,
                in: .userDomainMask)[0].appendingPathComponent(imgName)

        do {
            try imageData.write(to: docUrl)
        } catch {
            print("Write Data Error")
            print(error.localizedDescription)
        }
    }
```

Zip

In ViewController.swift

```
func zipTheCatPhoto(imgName: String, zipName: String) {

    let image = UIImage(named:imgName)
    let imageData: Data = UIImageJPEGRepresentation(image!, 1.0)!
    let docUrl = fileManager.urls(for: .documentDirectory, in: .userDomainMask)[0].appendingPathComponent(imgName)

    do {
        try imageData.write(to: docUrl)
    } catch {
        print("Write Data Error")
        print(error.localizedDescription)
    }

    do {
        zipFilePath = fileManager.urls(for: .documentDirectory, in: .userDomainMask)[0].appendingPathComponent(zipName)
        try Zip.zipFiles(paths: [docUrl], zipFilePath: zipFilePath, password: nil, progress: { (progress) -> () in
            print(progress)
        }) //Zip
    }
    catch {
        print("Something went wrong")
    }
}
```

Zip

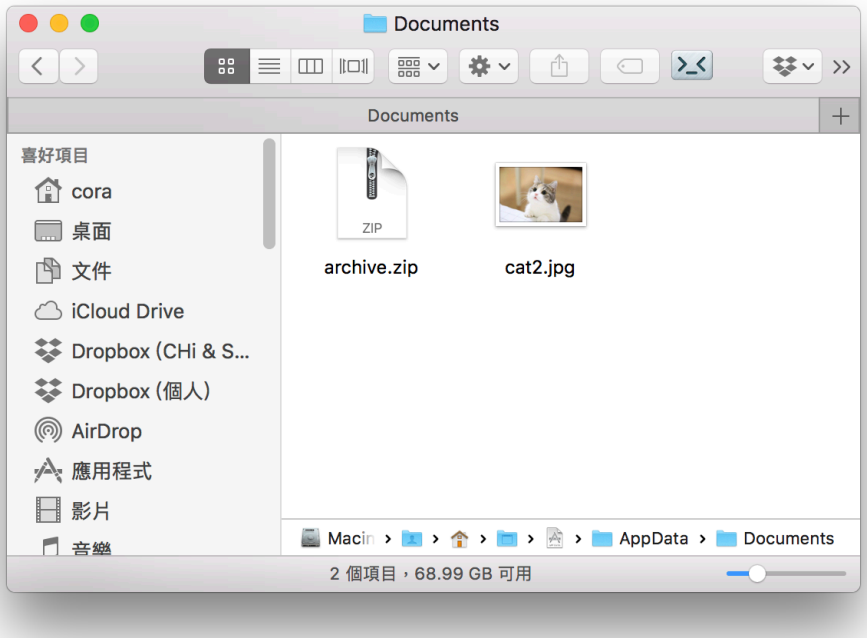


Zip

In ViewController.swift

```
class ViewController: UIViewController {

    let fileManager = FileManager.default
    var imageName = "cat2.jpg"
    var zippedName = "archive.zip"
    var zipFilePath:URL!

    override func viewDidLoad() {
        super.viewDidLoad()

        zipTheCatPhoto(imgName: imageName, zipName: zippedName)

    }
……
```

Run



WebService- SaveFile

http://192.168.10.127/GetBinaryFile.asmx

SaveFile **Action**

測試

測試表單只適用於來自本機電腦的要求。

SOAP 1.1

下列是 SOAP 1.1 要求與回應的範例。預留位置顯示之處必須代入實際的值。

```
POST /GetBinaryFile.asmx HTTP/1.1
Host: 192.168.10.127
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://192.168.10.127/SaveFile"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.
  <soap:Body>
    <SaveFile xmlns="http://192.168.10.127/">
      <file>
        <filedsc>string</filedsc>
        <filename>string</filename>
        <fltd>string</fltd>
        <fltno>string</fltno>
        <sect>string</sect>
        <upddate>string</upddate>
        <upduser>string</upduser>
        <zipfile>base64Binary</zipfile>
      </file>
      <sysPwd>string</sysPwd>
    </SaveFile>
  </soap:Body>
</soap:Envelope>
```

WebService- SaveFile

http://192.168.10.127/GetBinaryFile.asmx

```
POST /GetBinaryFile.asmx HTTP/1.1
Host: 192.168.10.127
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchem
  <soap12:Body>
    <SaveFile xmlns="http://192.168.10.127/">
      <file>
        <filedsc>string</filedsc>
        <filename>string</filename>
        <fltd>string</fltd>
        <fltno>string</fltno>
        <sect>string</sect>
        <upddate>string</upddate>
        <upduser>string</upduser>
        <zipfile>base64Binary</zipfile>
      </file>
      <sysPwd>string</sysPwd>
    </SaveFile>
  </soap12:Body>
</soap12:Envelope>
```

Namespace

Parameter

toSoapMessage

In ViewController.swift

```
func toSoapMessage12(methodName: String, paramValues: String, namespace: String) -> String
{
        var message: String = String()
        message += "<?xml version=\"1.0\" encoding=\"utf-8\"?>"
        message += "<soap12:Envelope xmlns:xsi=\"http://www.w3.org/2001/XMLSchema-
instance\" xmlns:xsd=\"http://www.w3.org/2001/XMLSchema\" xmlns:soap12=\"http://
www.w3.org/2003/05/soap-envelope\">"
        message += "<soap12:Body>"
        message += "<\(methodName) xmlns=\"\(namespace)\">"
        message += "\(paramValues)"
        message += "</\(methodName)>"
        message += "</soap12:Body>"
        message += "</soap12:Envelope>"
        return message
    }
```

```
POST /GetBinaryFile.asmx HTTP/1.1
Host: 192.168.10.127
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  <soap12:Body>
    <SaveFile xmlns="http://192.168.10.127/">
      <file>
        <filedsc>string</filedsc>
        <filename>string</filename>
        <fltd>string</fltd>
        <fltno>string</fltno>
        <sect>string</sect>
        <upddate>string</upddate>
        <upduser>string</upduser>
        <zipfile>base64Binary</zipfile>
      </file>
      <sysPwd>string</sysPwd>
    </SaveFile>
  </soap12:Body>
</soap12:Envelope>
```

soapRequest

In ViewController.swift

```
func soapRequest12(url:String, parameter:Dictionary<String, String>, methodName:String , namespace:String) -> URLRequest {
        let URL = NSURL(string: url)!
        var urlRequest: URLRequest = URLRequest(url:URL as URL)
        var rootCodeStr = ""

        if parameter != ["":""] {
            for dic in parameter {
                rootCodeStr += "<"
                rootCodeStr += dic.key
                rootCodeStr += ">"
                rootCodeStr += dic.value
                rootCodeStr += "</"
                rootCodeStr += dic.key
                rootCodeStr += ">"
            }
        }
        let soapMsg: String = toSoapMessage12(methodName: methodName, paramValues: rootCodeStr, namespace: namespace)
        urlRequest.setValue("application/soap+xml; charset=utf-8", forHTTPHeaderField: "Content-Type")
        //        mutableURLRequest.setValue(action, forHTTPHeaderField: "SOAPAction")
        urlRequest.setValue(String(soapMsg), forHTTPHeaderField: "Content-Length")
        urlRequest.httpMethod = "POST"
        urlRequest.httpBody = soapMsg.data(using: String.Encoding.utf8)
        print("soapMsg: ", soapMsg)
        return urlRequest
    }
```

parameter

```
POST /GetBinaryFile.asmx HTTP/1.1
Host: 192.168.10.127
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd
  <soap12:Body>
    <SaveFile xmlns="http://192.168.10.127/">
      <file>
        <filedsc>string</filedsc>
        <filename>string</filename>
        <fltd>string</fltd>
        <fltno>string</fltno>
        <sect>string</sect>
        <upddate>string</upddate>
        <upduser>string</upduser>
        <zipfile>base64Binary</zipfile>
      </file>
      <sysPwd>string</sysPwd>
    </SaveFile>
  </soap12:Body>
</soap12:Envelope>
```

WebService- SaveFile

In ViewController.swift

```
func requestSaveFile(filename:String, filedsc:String, zipPath:URL) {

        var data:Data = Data()
        do {
            data = try Data(contentsOf: zipPath)
        } catch {
            print("Error")
        }

        let caFile = ["filedsc":filedsc,
                      "filename":filename,
                      "fltd":"",
                      "sect":"",
                      "upddate":"",
                      "upduser":"",
                      "zipfile":data.base64EncodedString()]

        var caFileStr = ""

        for dic in caFile {
            caFileStr += "<"
            caFileStr += dic.key
            caFileStr += ">"
            caFileStr += dic.value
            caFileStr += "</"
            caFileStr += dic.key
            caFileStr += ">"
        }

        let parameters = ["file":caFileStr,
                          "sysPwd":""]

        let urlRequest = soapRequest12(url: "http://192.168.10.127/GetBinaryFile.asmx", parameter: parameters, methodName: "SaveFile", namespace: "http://192.168.10.127/")

        Alamofire.request(urlRequest).response { (response) in

            if let data = response.data, let utf8Text = String(data: data, encoding: .utf8) {

                print("Data: \(utf8Text)")
            }
        }
    }
```

WebService- SaveFile

In ViewController.swift

```
override func viewDidLoad() {
    super.viewDidLoad()

    zipTheCatPhoto(imgName: imageName, zipName: zippedName)

    requestSaveFile(filename:zippedName, filedsc:"", zipPath:zipFilePath)

}
```

Run

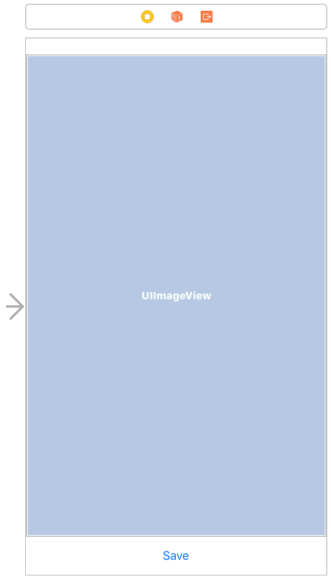
Draw

Michael Pan

Storyboard

UIImageView

“Save” Button



ViewController

```
class ViewController: UIViewController {

    @IBOutlet weak var imageView: UIImageView!

    var lastPoint = CGPoint.zero
    var brushWidth: CGFloat = 10.0
    var opacity: CGFloat = 1.0
    var swiped = false
……
```

DrawLine

In ViewController.swift

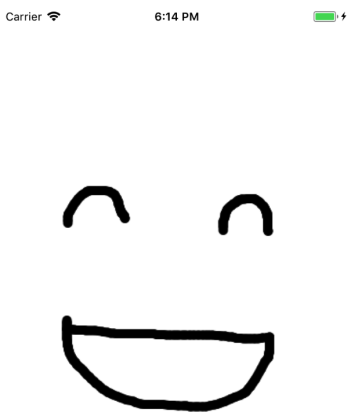
```
func drawLine(fromPoint: CGPoint, toPoint: CGPoint) {

    UIGraphicsBeginImageContext(view.frame.size)
    let context = UIGraphicsGetCurrentContext()
    imageView.image?.draw(in: CGRect(x: 0, y: 0, width: view.frame.size.width,
                                                 height: view.frame.size.height))

    context?.move(to: fromPoint)
    context?.addLine(to: toPoint)

    context?.setLineCap(.round)
    context?.setLineWidth(brushWidth)
    context?.setStrokeColor(UIColor.black.cgColor)
    context?.setBlendMode(.normal)

    context?.strokePath()

    imageView.image = UIGraphicsGetImageFromCurrentImageContext()
    imageView.alpha = opacity
    UIGraphicsEndImageContext()
}
```

DrawLine

In ViewController.swift

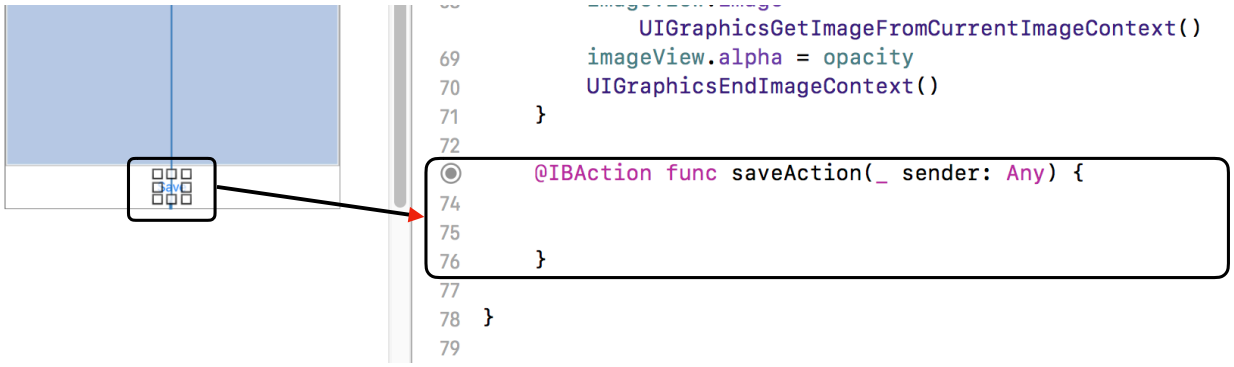
```
override func touchesBegan(_ touches: Set<UITouch>, with event: UIEvent?) {
    swiped = false
    if let touch = touches.first {
        lastPoint = touch.location(in: self.view)
    }
}

override func touchesMoved(_ touches: Set<UITouch>, with event: UIEvent?) {
    swiped = true
    if let touch = touches.first {
        let currentPoint = touch.location(in: view)
        drawLine(fromPoint: lastPoint, toPoint: currentPoint)

        lastPoint = currentPoint
    }
}

override func touchesEnded(_ touches: Set<UITouch>, with event: UIEvent?) {
    if !swiped {
        drawLine(fromPoint: lastPoint, toPoint: lastPoint)
    }
}
```

Run



Save our Image

In ViewController.swift



Save our Image

In ViewController.swift

```
@IBAction func saveAction(_ sender: Any) {

    UIGraphicsBeginImageContextWithOptions(imageView.layer.frame.size, false, 1.0)

    imageView.layer.render(in: UIGraphicsGetCurrentContext()!)
    let viewImage = UIGraphicsGetImageFromCurrentImageContext()!

    UIGraphicsEndImageContext()

    let data = UIImagePNGRepresentation(viewImage)

}
```

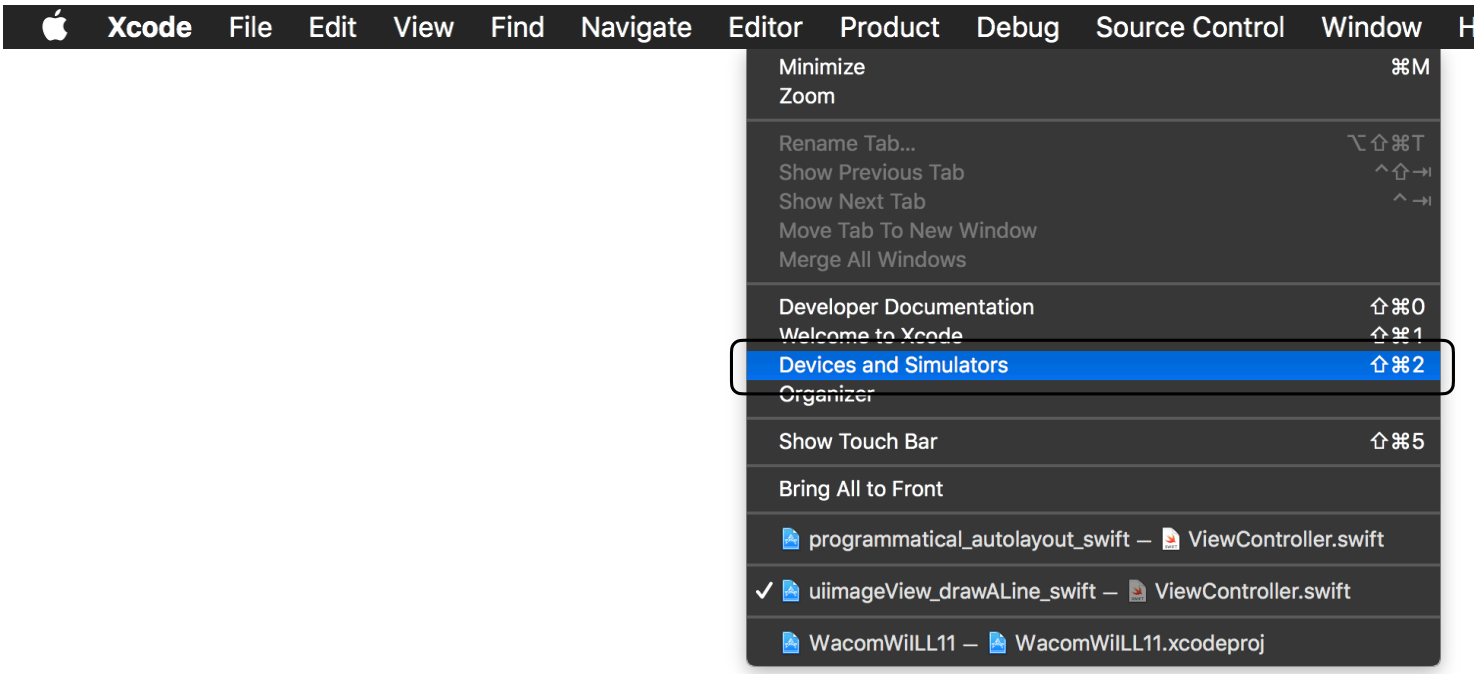
Save our Image

In ViewController.swift

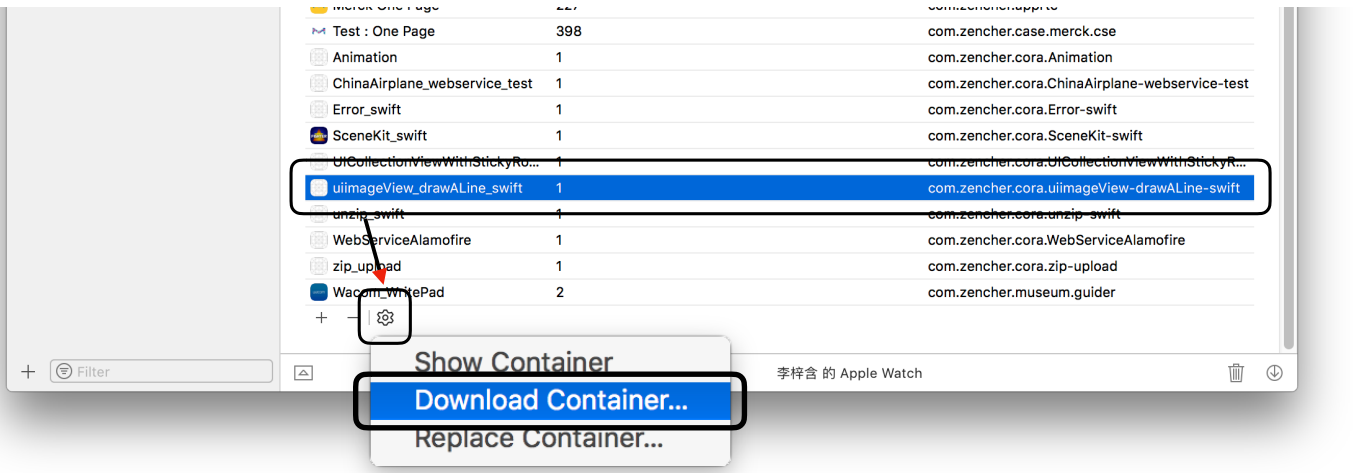
```
@IBAction func saveAction(_ sender: Any) {
    UIGraphicsBeginImageContextWithOptions(imageView.layer.frame.size, false, 1.0)
    imageView.layer.render(in: UIGraphicsGetCurrentContext()!)
    let viewImage = UIGraphicsGetImageFromCurrentImageContext()!
    UIGraphicsEndImageContext()
    let data = UIImagePNGRepresentation(viewImage)

    let documentsDir = FileManager.default.urls(for: .documentDirectory,
                                                in: .userDomainMask)[0]
    let writePath = documentsDir.appendingPathComponent("myImage.png")

    do {
        try data?.write(to: writePath)
    } catch {
        print("Write Error")
    }
}
```

Run

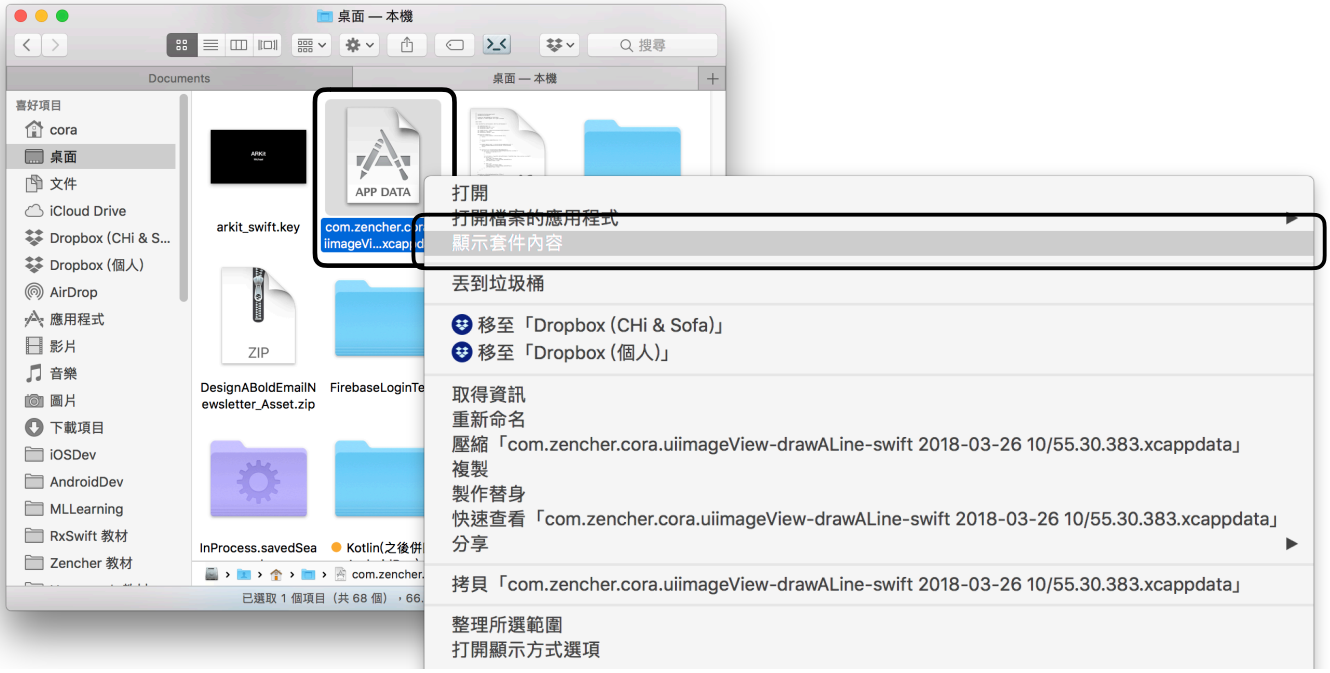
Check our Saved Image



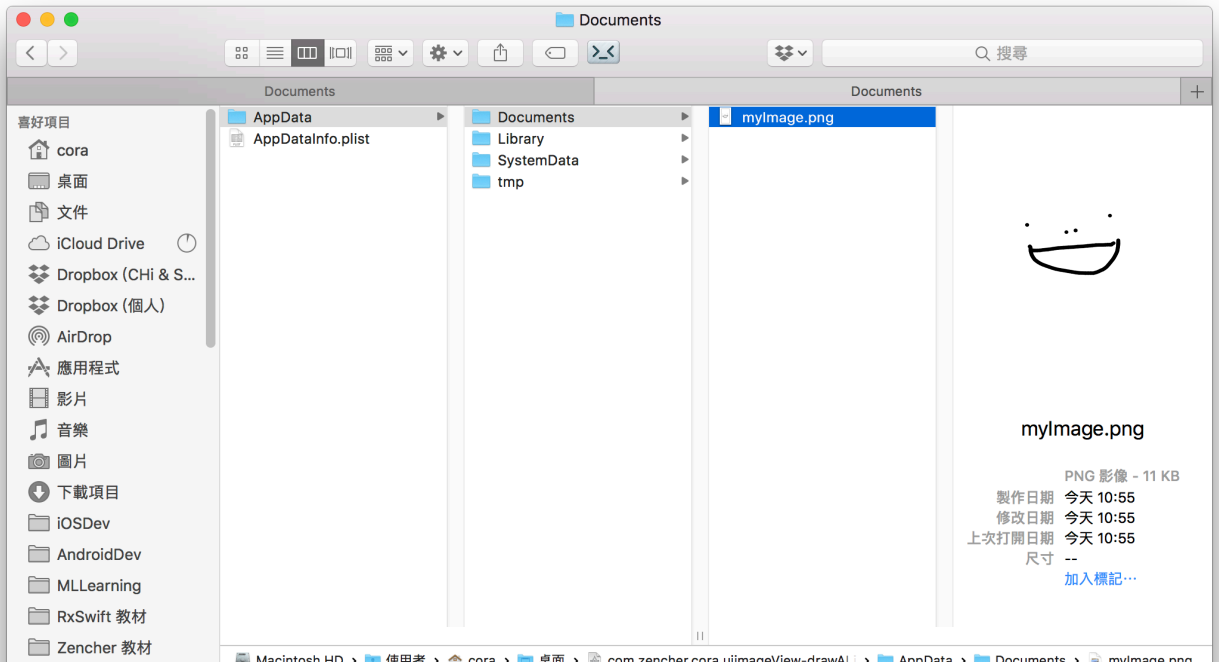
Check our Saved Image



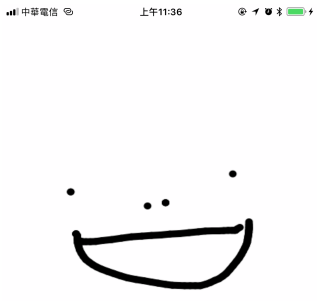
Check our Saved Image



Check our Saved Image



Run

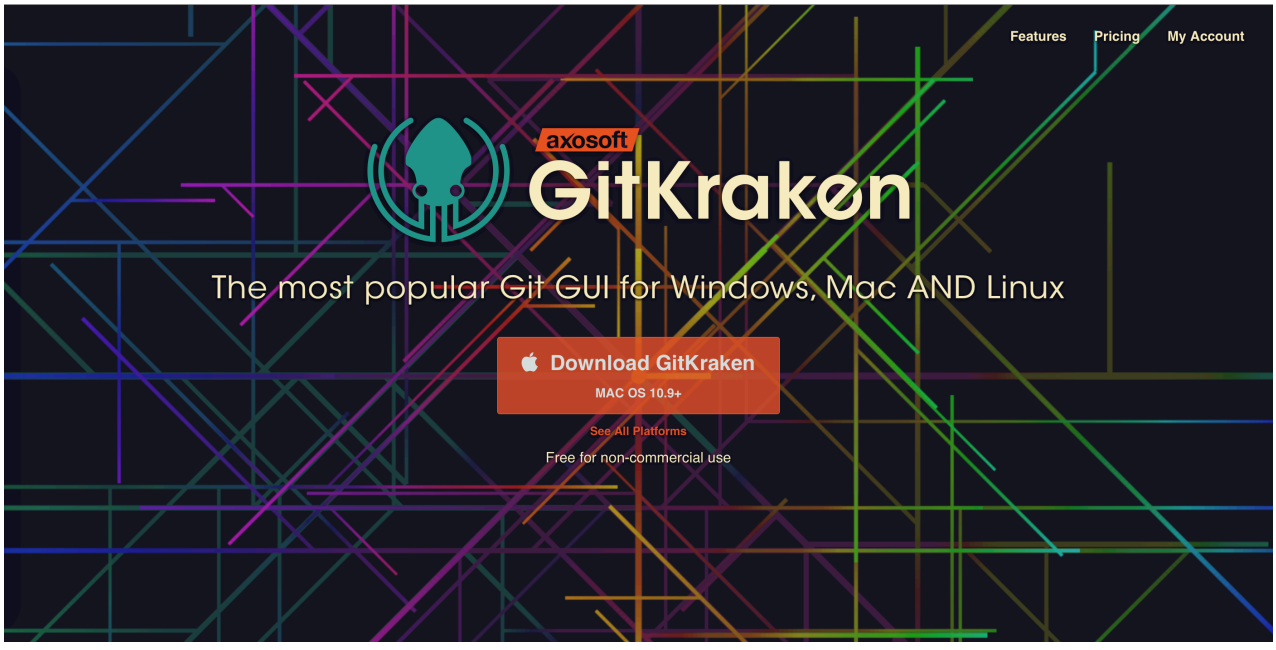


Save

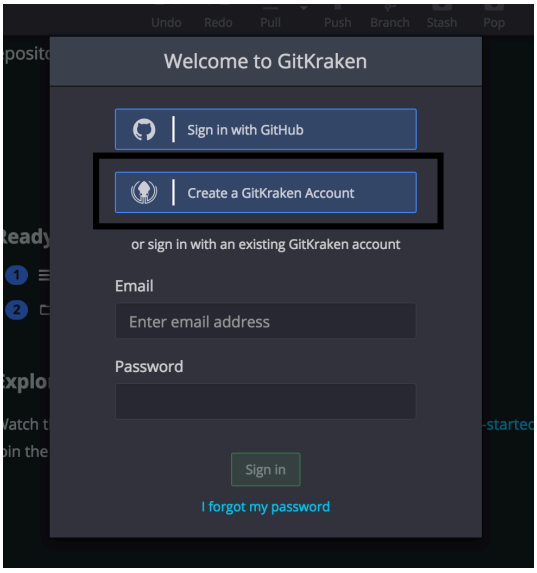
GitKraken

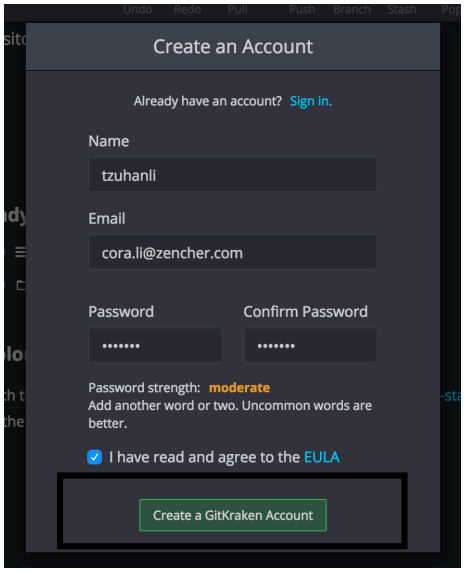
Michael Pan

Download GitKraken

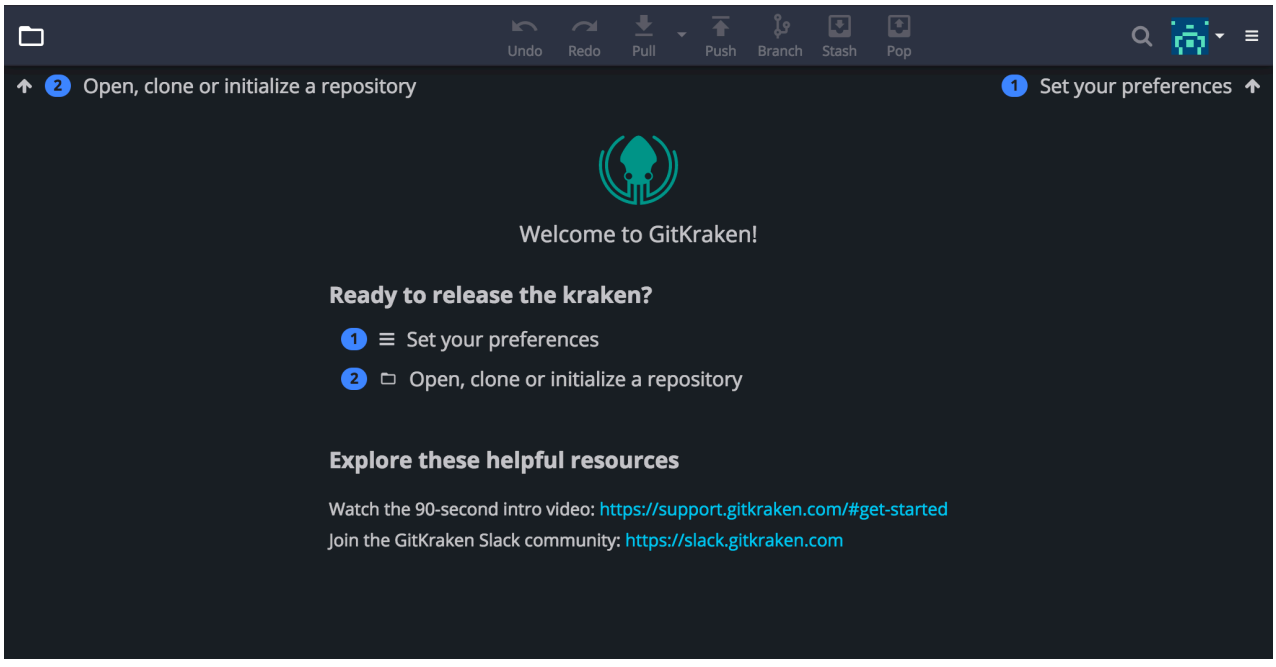


註冊帳號

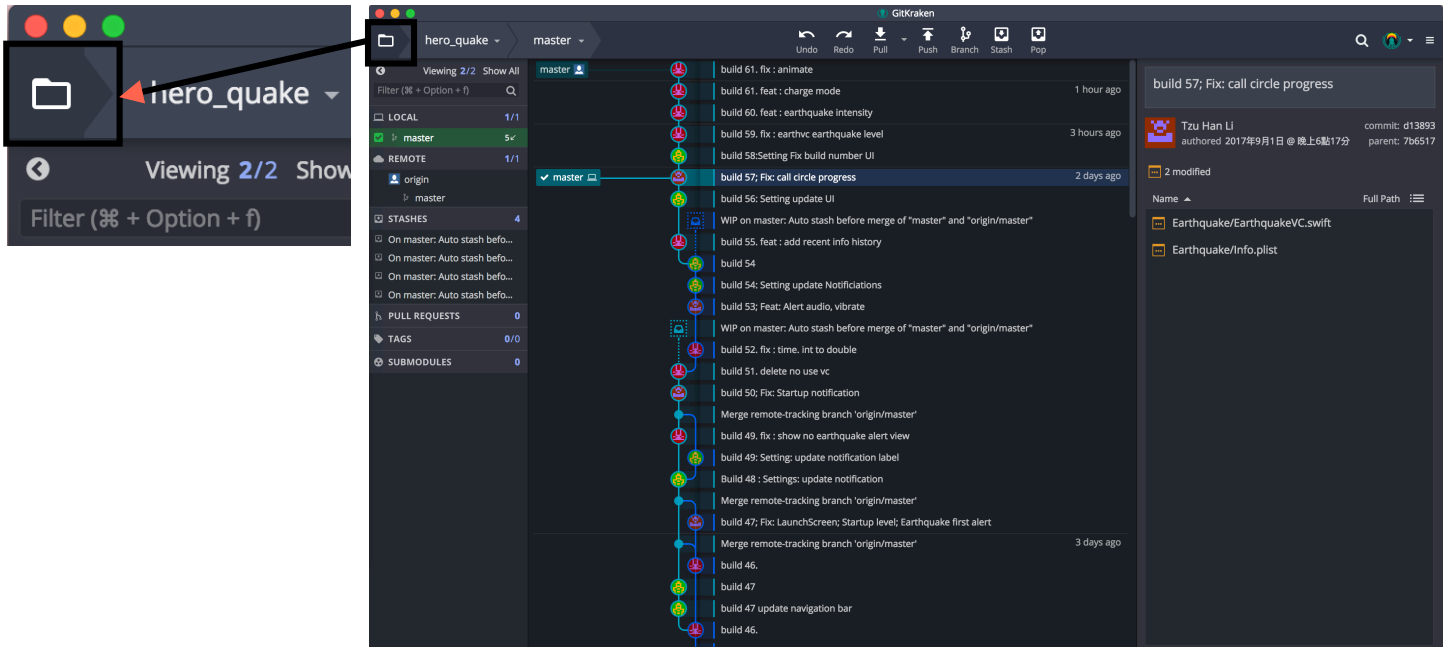




成功畫面



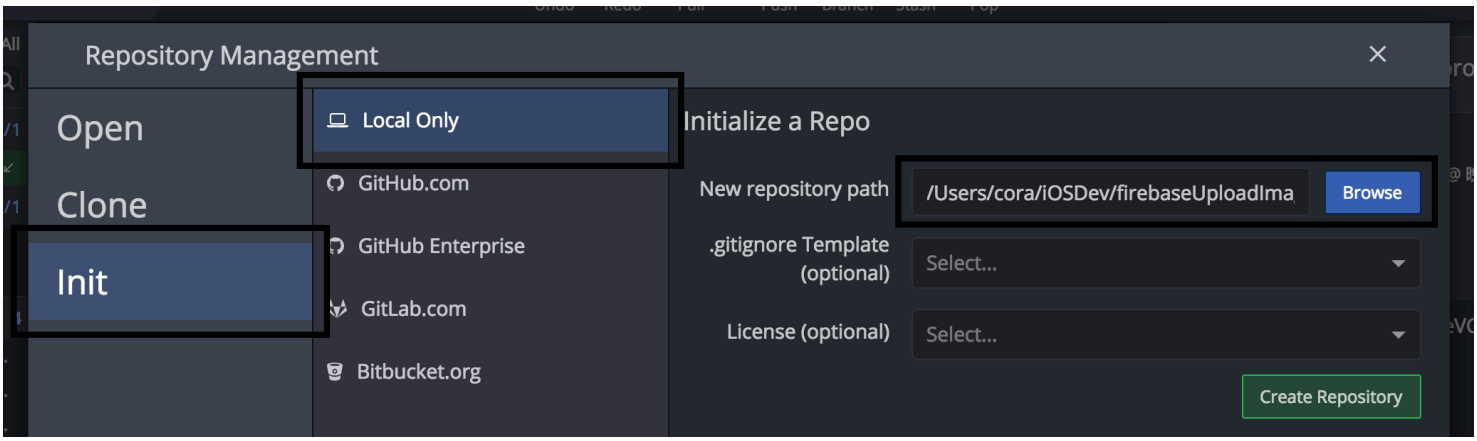
設定 Repository



設定 Repository

Init repository

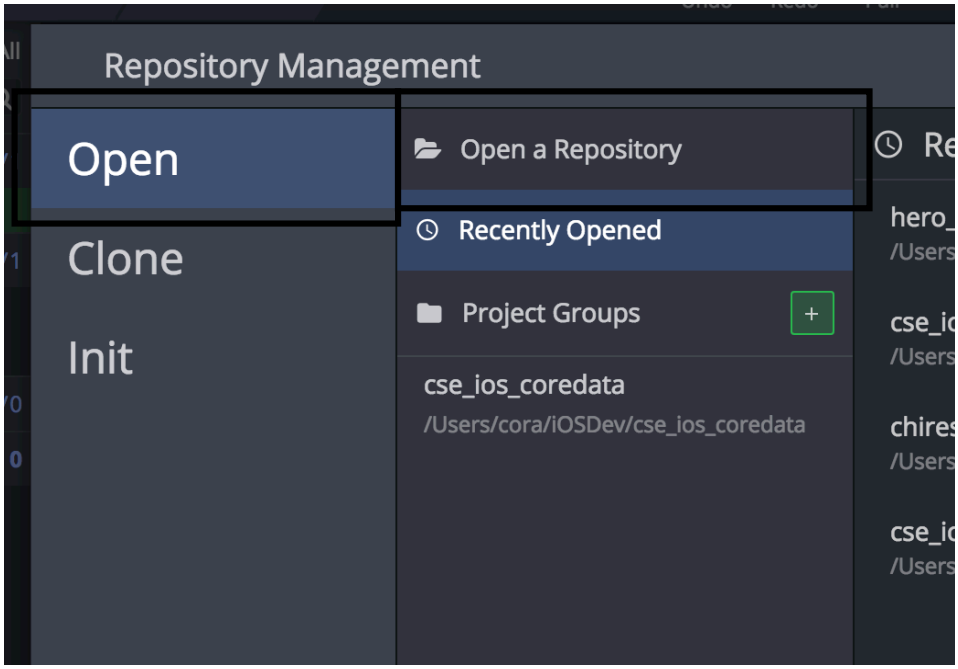
選擇專案資料夾



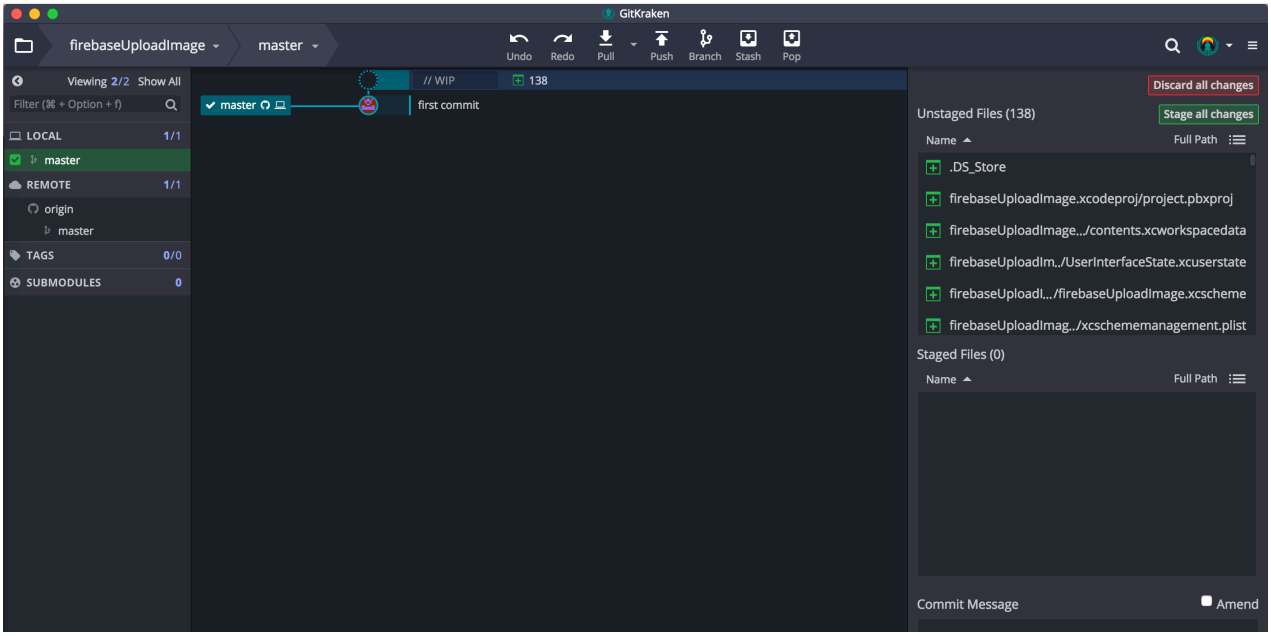
開啟 Repository

打開 Repository

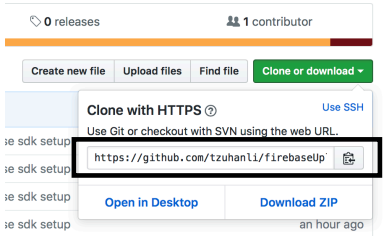
選擇專案資料夾

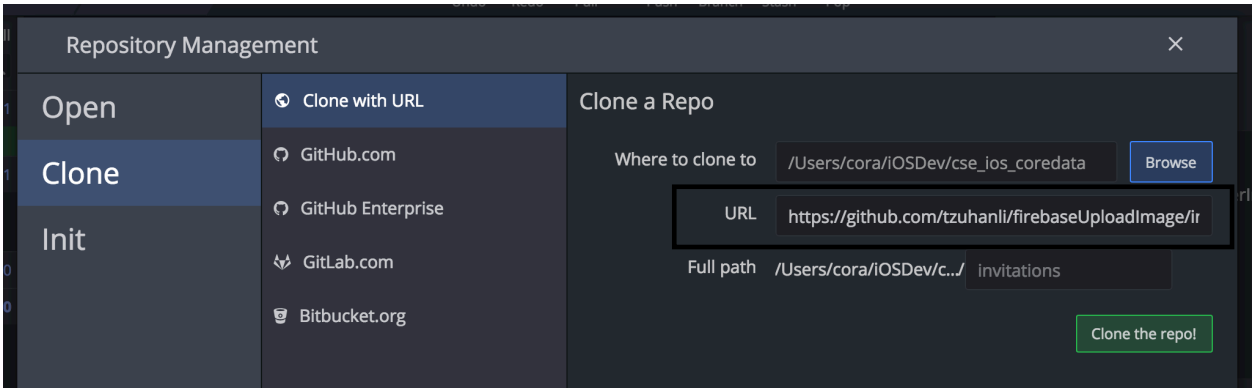


開啟 Repository



使用URL加入Repository





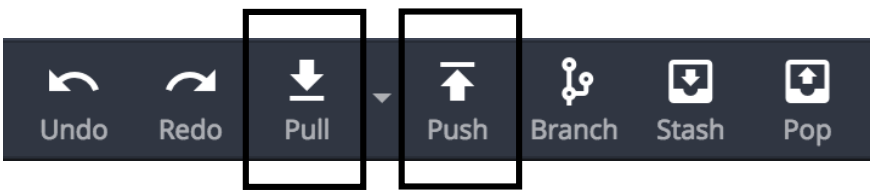
GitKraken 操作

Stage

Commit

Pull

Push

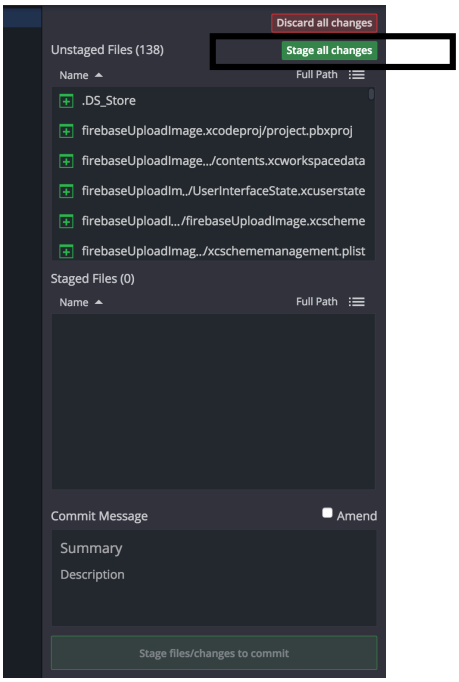


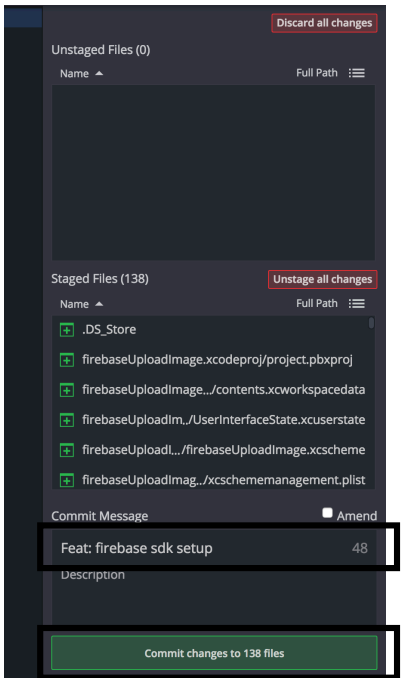
GitKraken 操作- Stage, Commit

Stage all changes

Commit Message

Commit



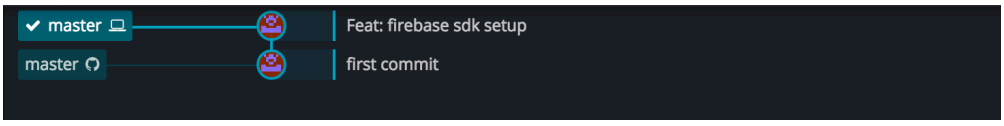


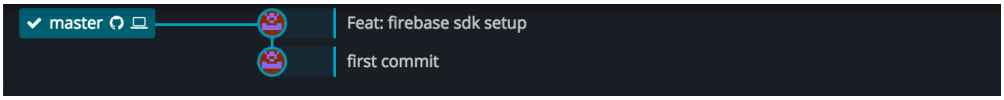
GitKraken 操作- Push

Push

Log In

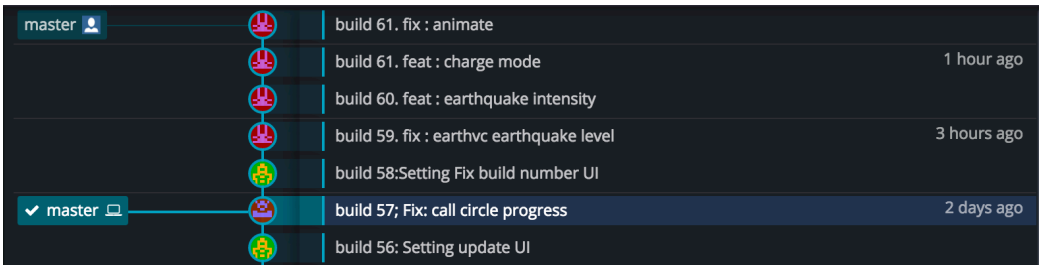
Push 成功



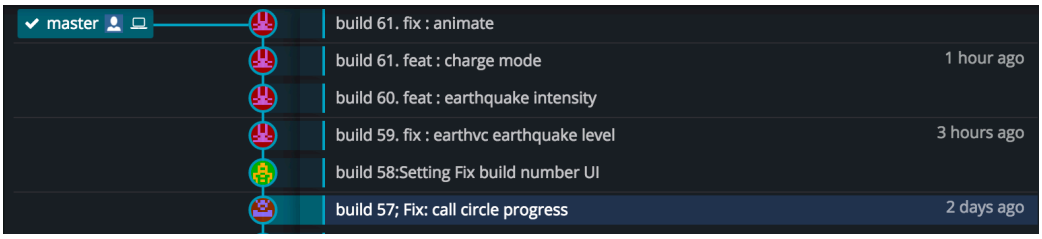


GitKraken 操作- Pull

Pull

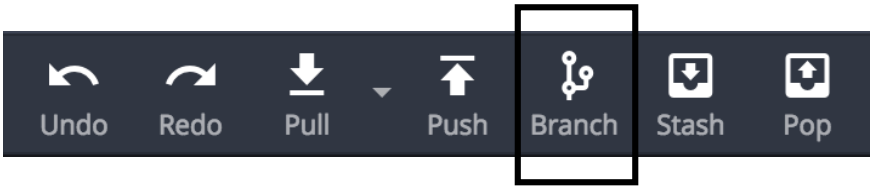


Pull 成功

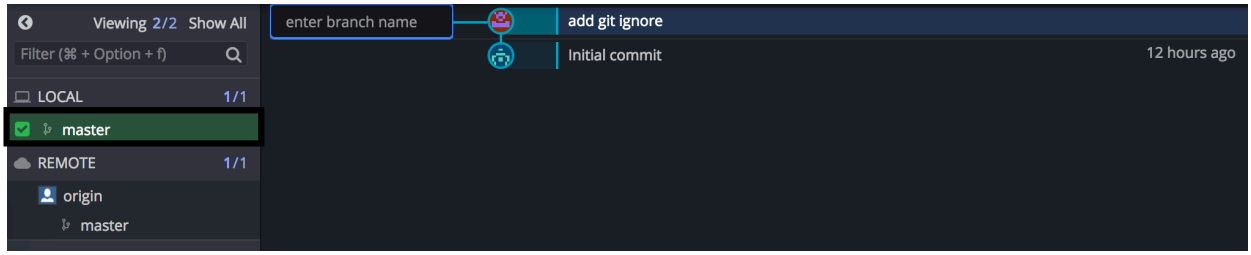


GitKraken 操作

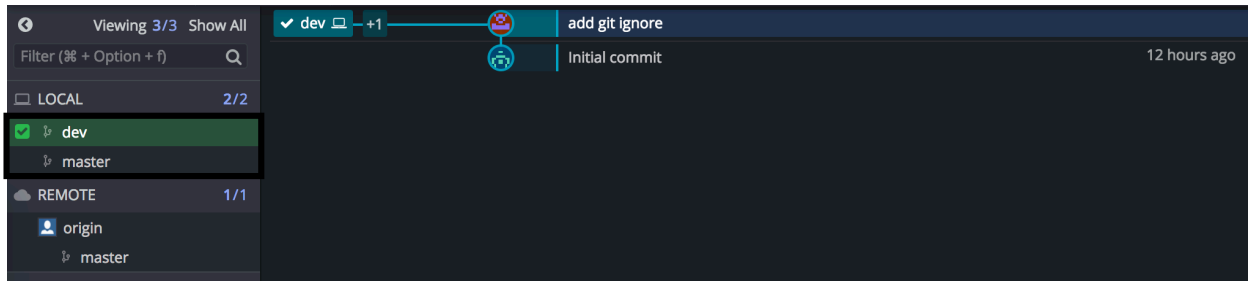
Branch



GitKraken 操作- Branch

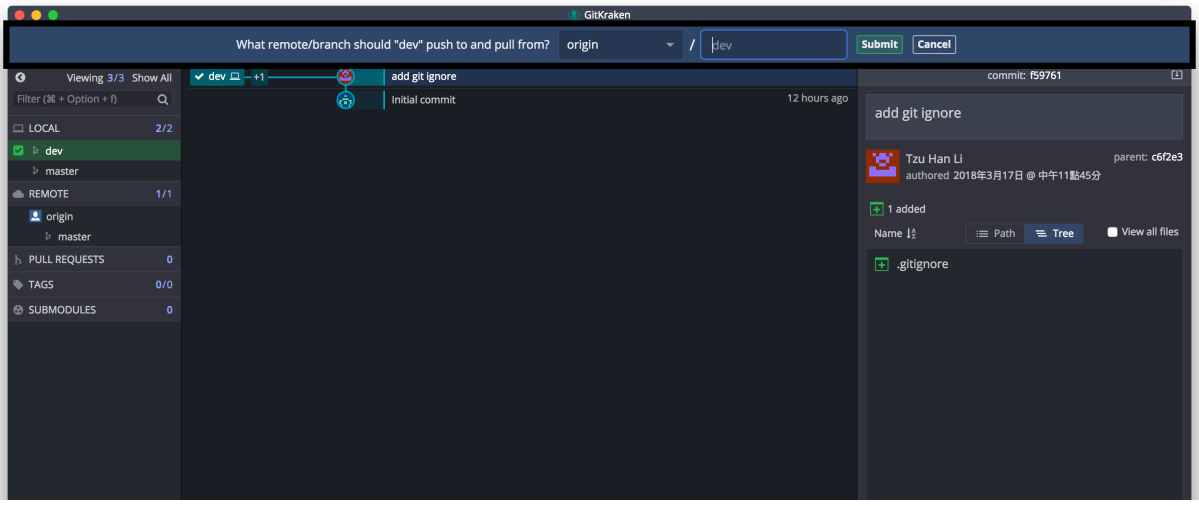


▪

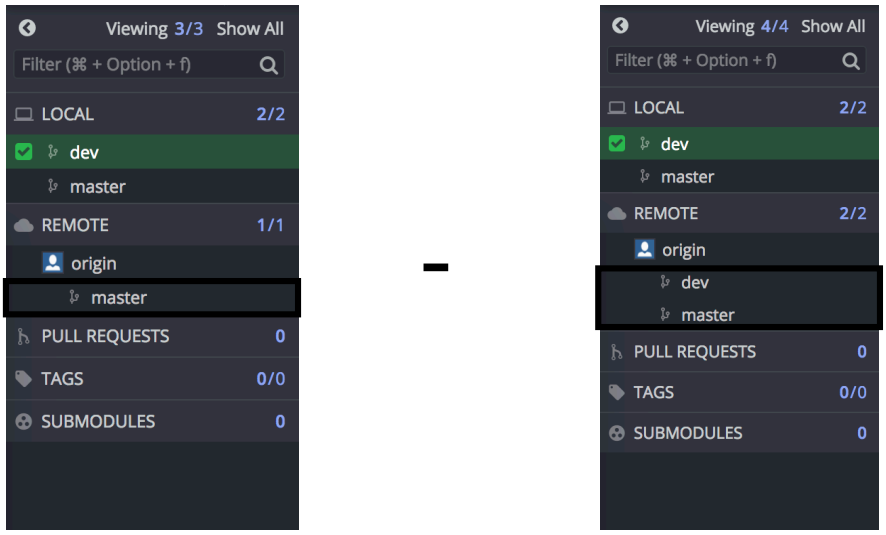


GitKraken 操作- Branch

Push



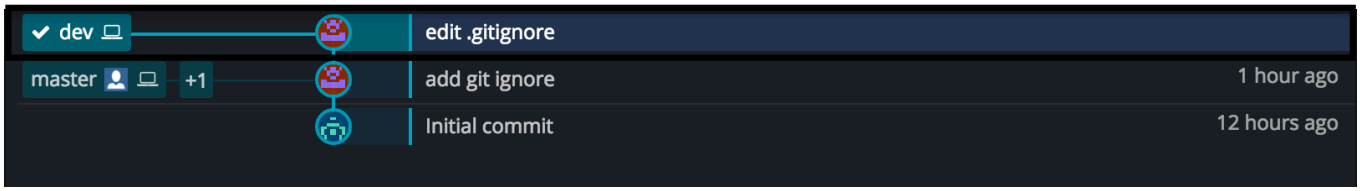
GitKraken 操作- Branch



GitKraken 操作- Branch

修改任意檔案

Push



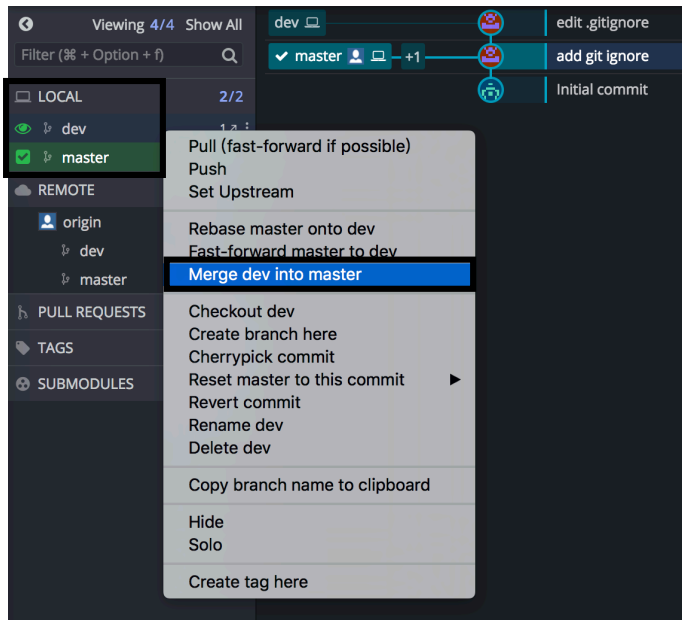
GitKraken 操作- Merge

把分支 (Branch) 合併回 master 上

選擇 master

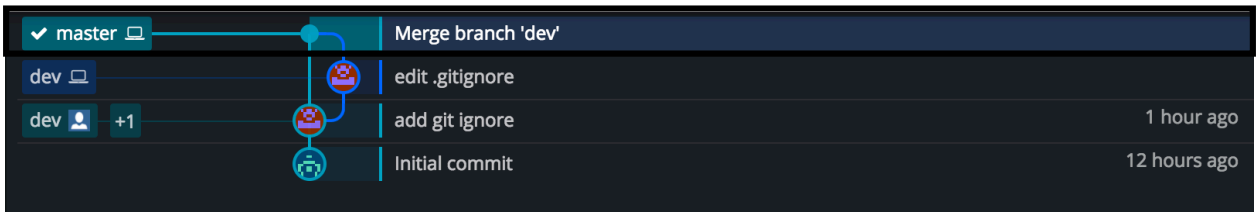
對要合併的分支按右鍵

選擇 “Merge dev into master”



GitKraken 操作- Merge

觀察 master 的變化

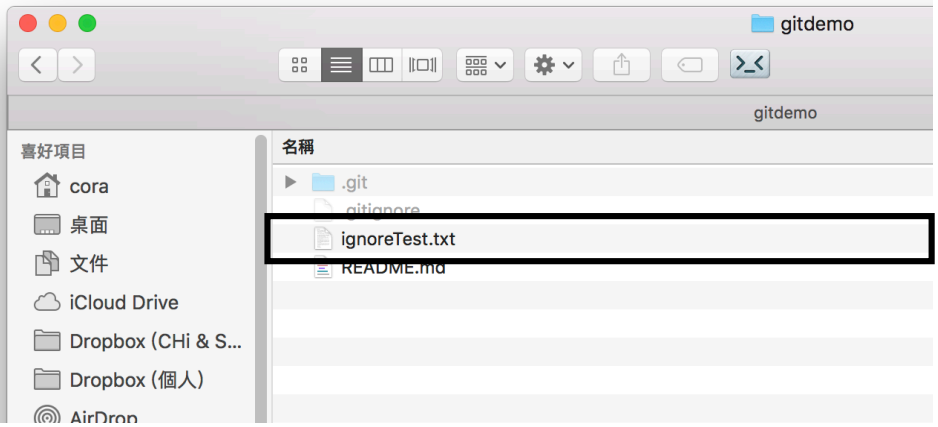


編輯 .gitignore

避免上傳一些不需要的資料、檔案

練習：

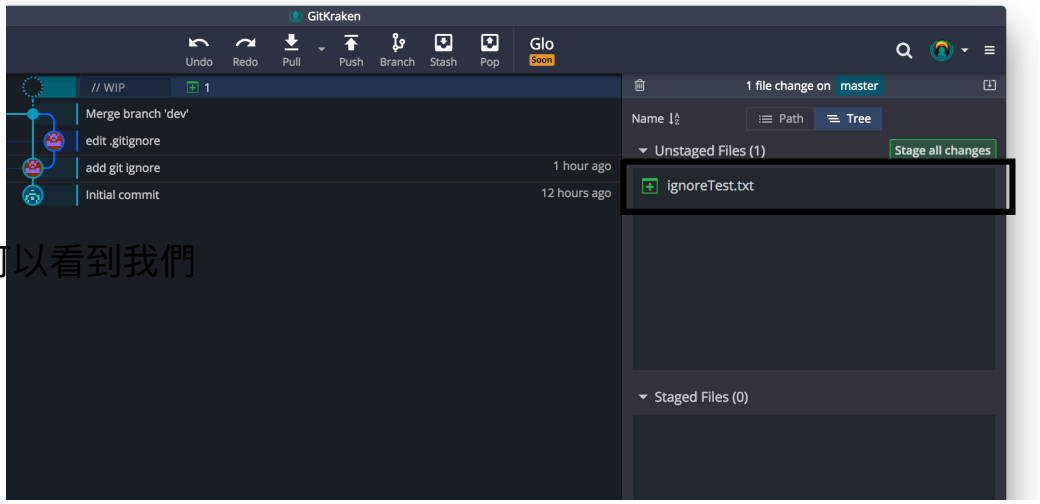
新增一個 txt 檔



編輯 .gitignore

觀察 GitKraken

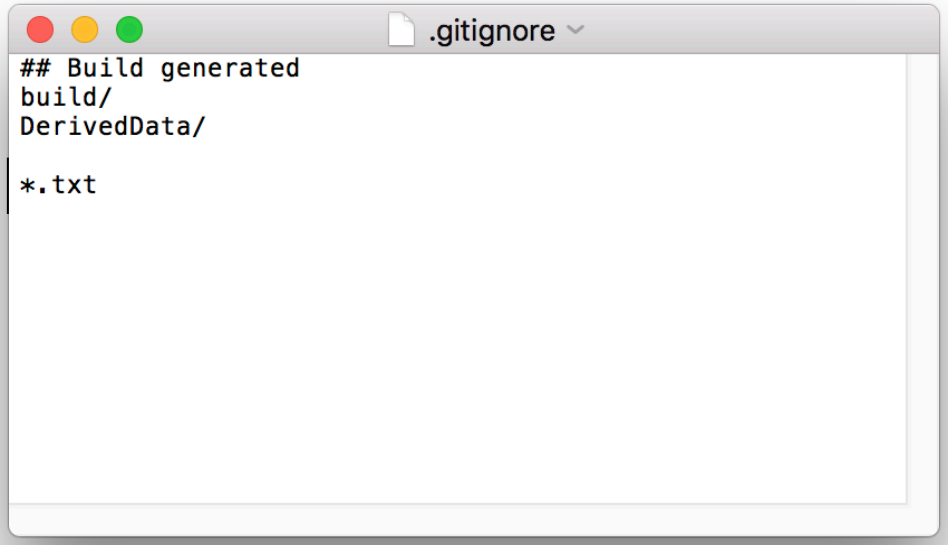
Unstaged 的區塊可以看到我們剛剛新增的 txt 檔



編輯 .gitignore

修改 .gitignore 檔

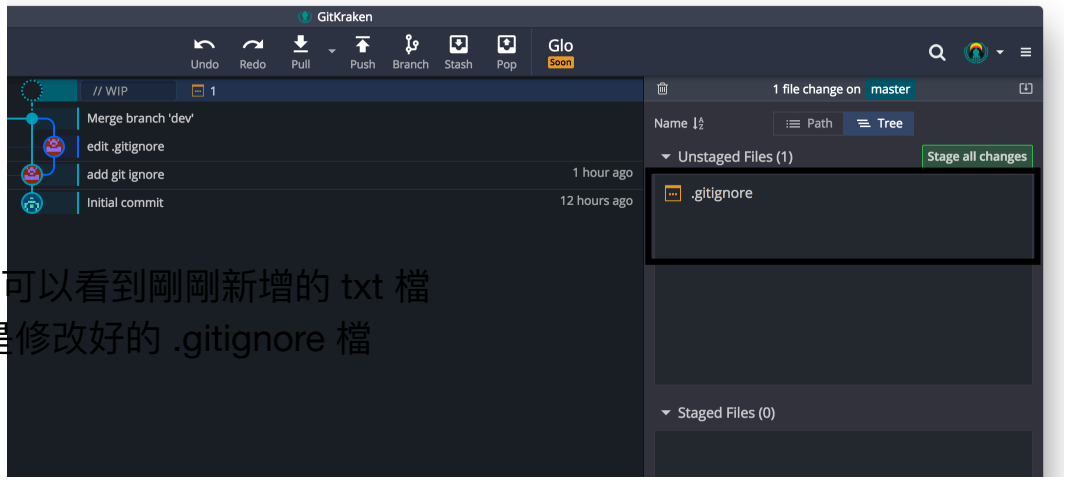
加入 *.txt
（排除所有 txt 檔）



編輯 .gitignore

觀察 GitKraken

Unstaged 的區塊可以看到剛剛新增的 txt 檔
消失了，取代的是修改好的 .gitignore 檔



編輯 .gitignore

建議要排除上傳的檔案類型：

build/

DerivedData/

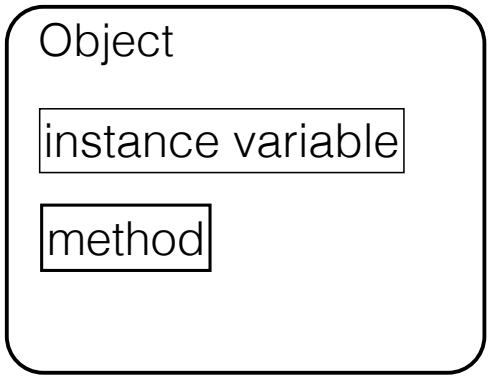
*.DS_Store

Object

Michael Pan

Object Oriented Language

Object 的觀念和 Struct 類似



member

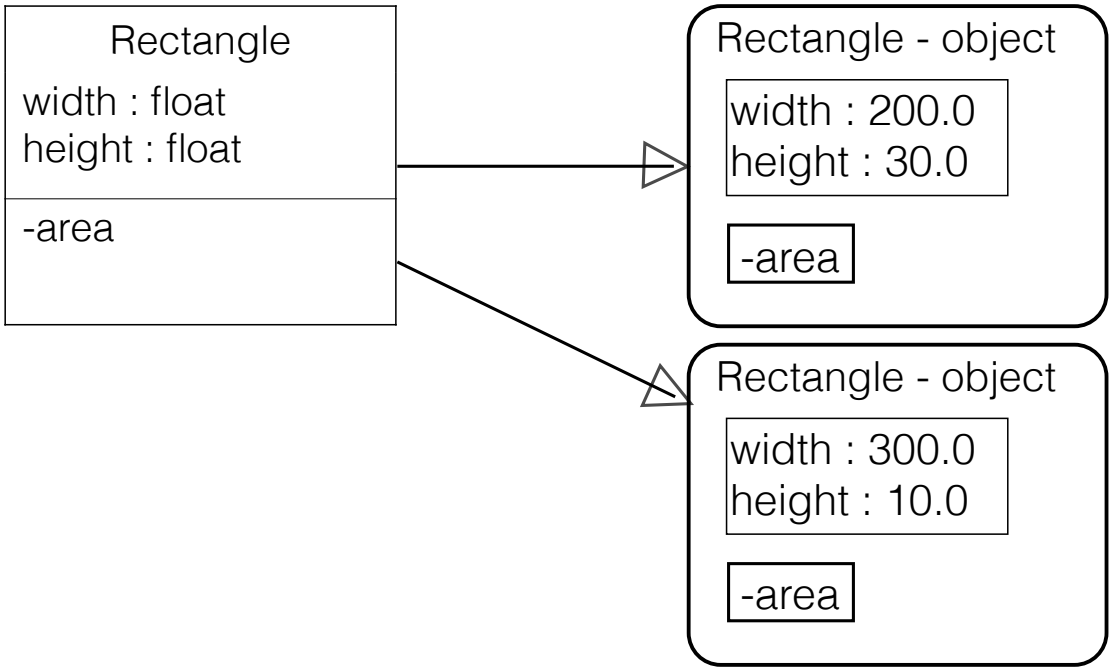
```
struct Rectangle{
    float width;
    float height;
};
```

Like struct, Object need a type

```
struct Rectangle{
    float width;
    float height;
};

int main(int argc, const char * argv[]){
  struct Rectangle r1 ;
}
```

操作的變數

Class - the type of object



建立專案

建立一個 Command Line Tool 專案

命名為 GenerateID

NSUUID

```
#import <Foundation/Foundation.h>

int main(int argc, const char * argv[]){

    @autoreleasepool {

        NSUUID * myid = [NSUUID UUID];
    }
    return 0;
}
```

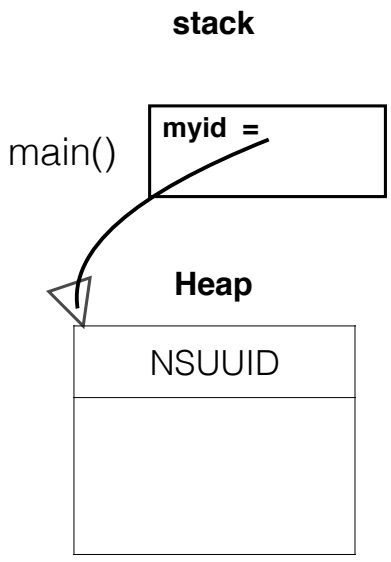
What does UUID do

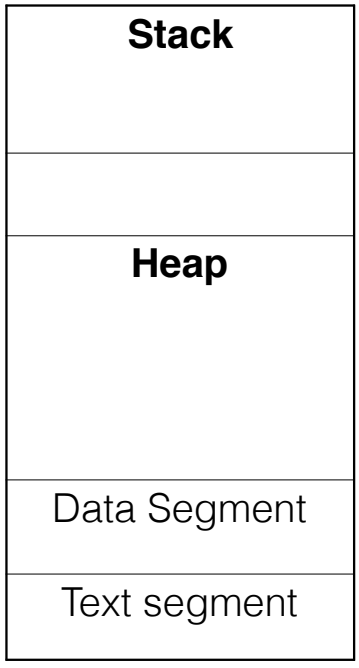
Allocate a memory for NSUUID - we call it **instance**

Initialise the instance

return the most front address of the instance

Memory View





address of Object

```
#import <Foundation/Foundation.h>

int main(int argc, const char * argv[]){
    @autoreleasepool {
        NSUUID * myid = [NSUUID UUID];
        NSLog(@"address of id is %p", myid);
    }
    return 0;
}
```

Message & method

[NSUUID UUID] ; // send message to NSUUID

NSUUID 是接收者

接收者可以是 class 或是 instance，必需具有可以被呼叫的 method

NSUUID 是 class

UUID 是屬於 NSUUID 可以被呼叫的 method 名稱，又稱 **selector**

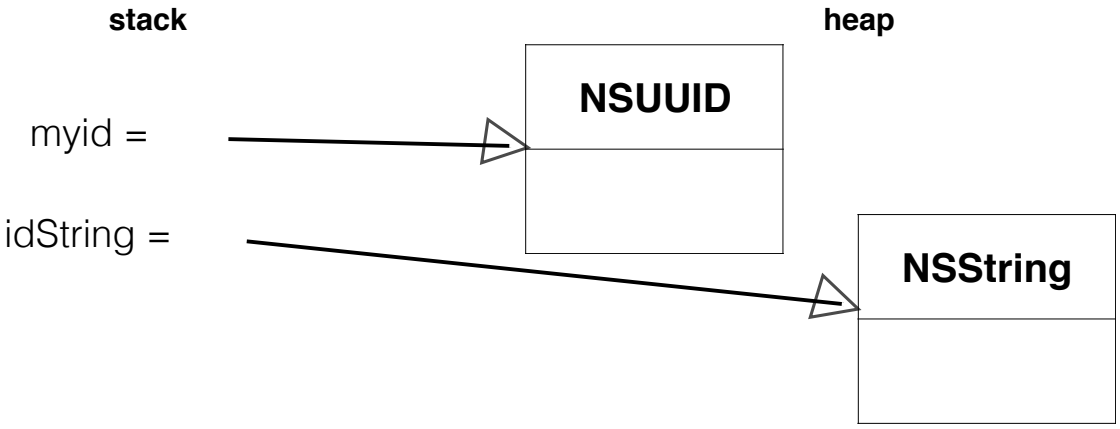
NSUUID

```
int main(int argc, const char * argv[]){

    @autoreleasepool {

        NSUUID * myid = [NSUUID UUID];
        NSLog(@"%@", myid);
    }
    return 0;
}
```

More method

```
#import <Foundation/Foundation.h>

int main(int argc, const char * argv[]){

    @autoreleasepool {
        NSUUID * myid = [NSUUID UUID];
        NSString * idString = [myid UUIDString];
        NSLog(@"UUID String is %@",idString);
    }
    return 0;
}
```

Memory View



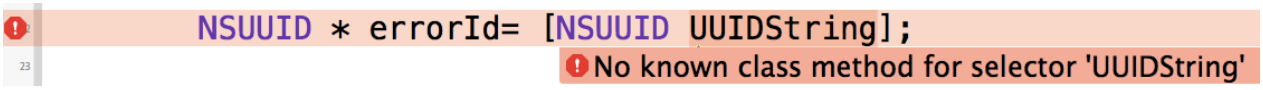
Try & error

UUID not uuid

Capitalization does matter

```
NSUUID * errorId= [NSUUID uuid];
                  No known class method for selector 'uuid'
NSString * idString = [myid UUIDstring];
                  No visible @interface for 'NSUUID' declares the selector 'UUIDstring'
```

instance method is not class method



命名習慣 - Naming Convention

Camel Case

兩個 term 在一起第一個 term 的第一字小寫後接字母第一字大寫

string**B**y**A**ppending**S**tring

UUID 是專有名詞所以都大寫

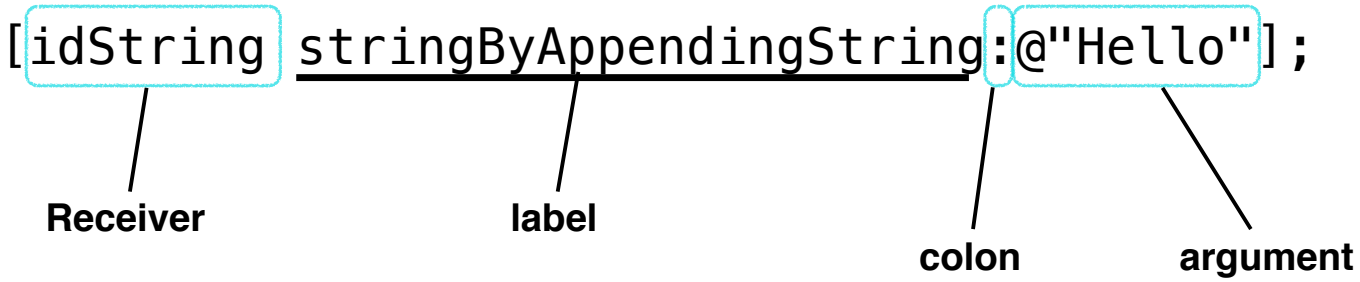
UUIDString - String 的 S 大寫

NS 開頭是歷史的記錄代表 NEXTSTEP。Objective-C 這個語言最早由 NeXT.inc 這家公司創造，用在 NEXTSTEP 這個作業系統上。

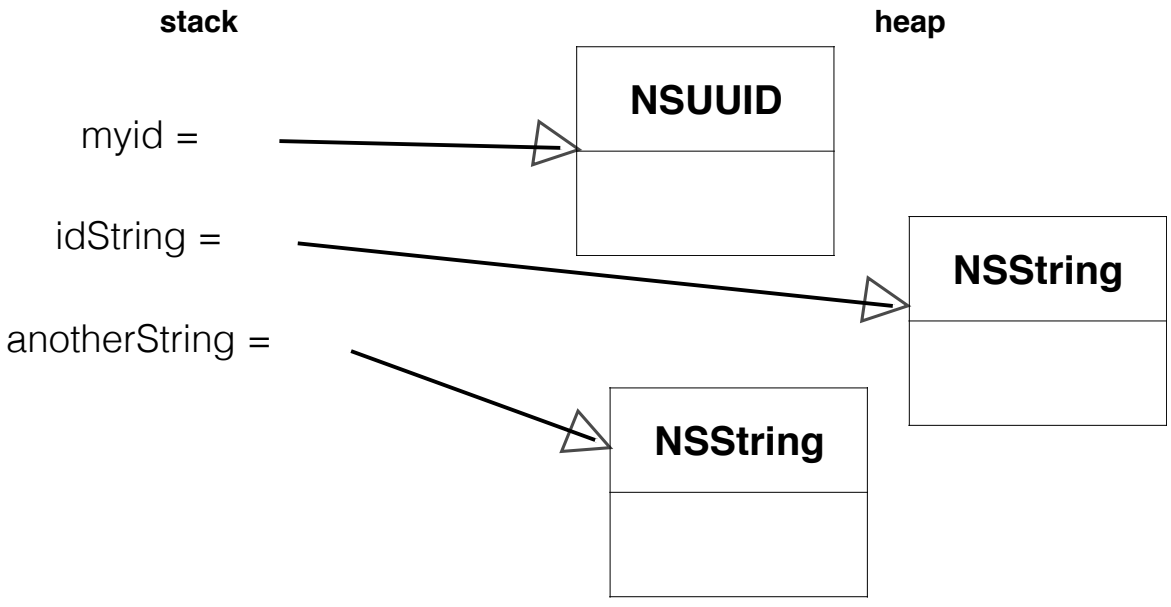
Argument in method

```
NSString * idString = [myid UUIDString];
NSLog(@"UUID String is %@",idString);

NSString * anotherString = [idString stringByAppendingString:@"Hello"];
```

```
2014-02-10 17:15:33.157 GenerateID[1328:303] UUID String is B7B0FEA2-49E8-42C8-8092-582BE4C57FDE
2014-02-10 17:15:33.158 GenerateID[1328:303] B7B0FEA2-49E8-42C8-8092-582BE4C57FDEHello
```



Memory View



More arguments

```
NSString * anotherString = [idString stringByAppendingString:@"Hello"];
NSLog(@"%@",anotherString );
NSString * changedStr = [anotherString
stringByReplacingOccurrencesOfString:@"Hello" withString:@"xxxx"];
NSLog(@"%@",changedStr);
```

```
A6AD1DD2-485C-4448-ABDE-C530A0690866Hello
A6AD1DD2-485C-4448-ABDE-C530A0690866xxxx
```

[] in []

```
NSUUID * myid = [NSUUID UUID];
NSString * idString = [myid UUIDString];
NSLog(@"UUID String is %@",idString);
```

```
NSString * idString = [[NSUUID UUID] UUIDString];
NSLog(@"UUID String is %@",idString);
```

Normal way - allocation & initialisation

```
NSUUID * myid = [NSUUID UUID];
```

```
NSUUID * myid = [[NSUUID alloc] init];
```

Send message to nil

nil is the empty value of instance

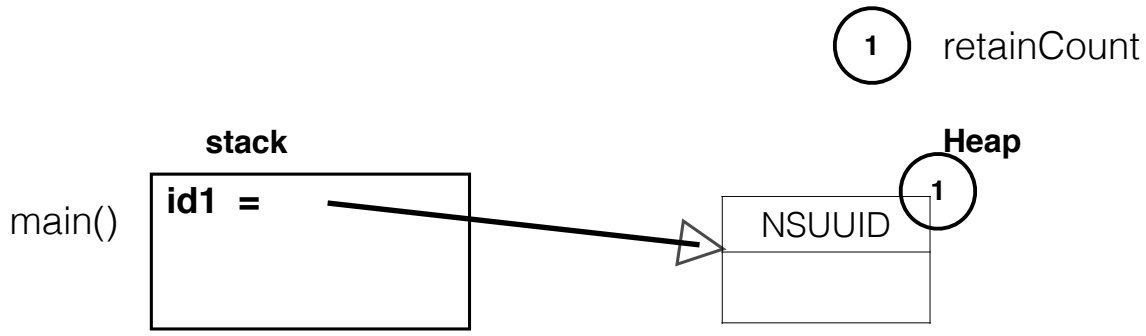
```
myid = nil;
NSLog(@"%@",[myid UUIDString]);
```

id

```
NSUUID * myid = [NSUUID UUID];
NSString * idString = [myid UUIDString];

id omnivar = myid;
omnivar = idString;
```

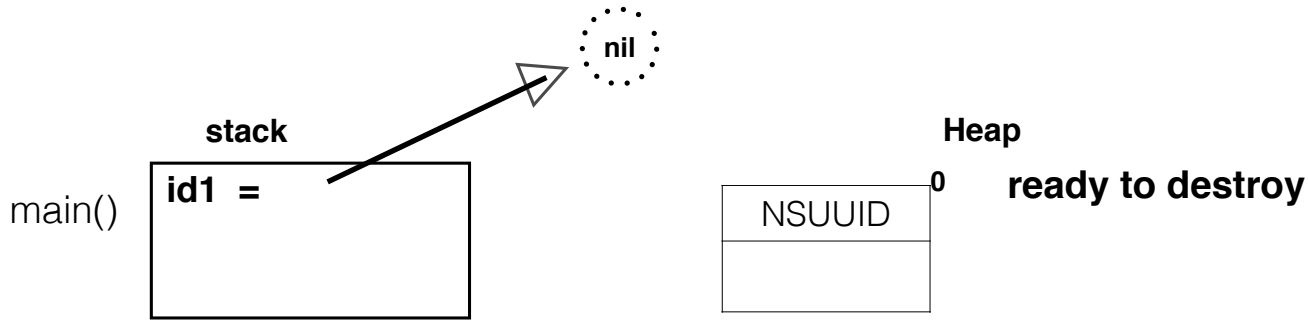
Memory Management - ARC

```
NSUUID * id1 = [NSUUID UUID];
```



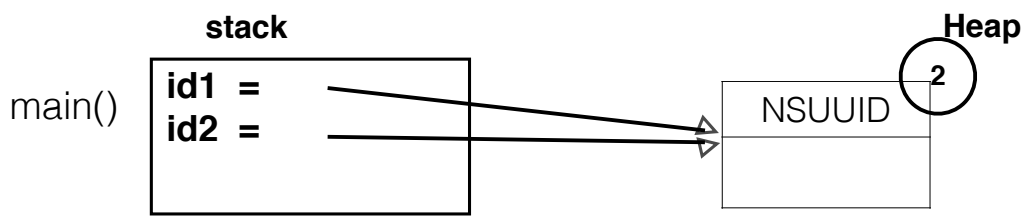
retainCount is Zero

```
NSUUID * id1 = [NSUUID UUID];
id1 = nil;
```



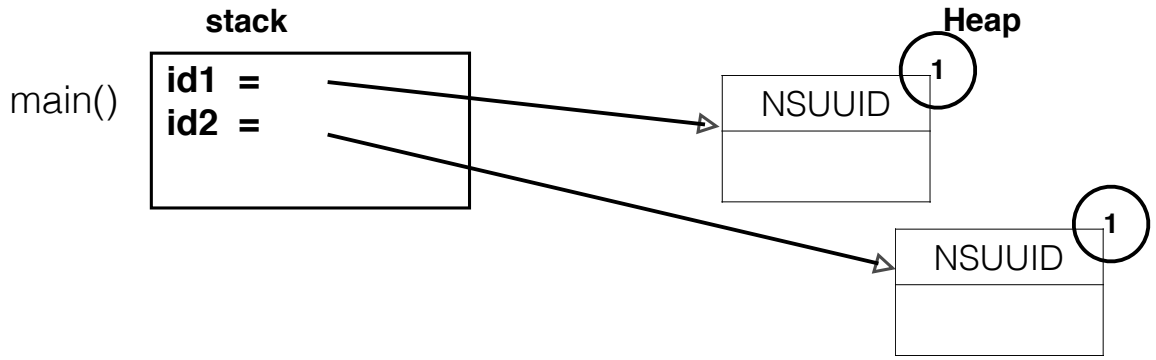
retainCount is more than 1

```
NSUUID * id1 = [NSUUID UUID];
NSUUID * id2 = id1;
```



retainCount is more than 1

```
NSUUID * id1 = [NSUUID UUID];
NSUUID * id2 = id1;
id2 = [NSUUID UUID];
```



Recap

使用 NSDate 和 NSDateFormatter

改變 date format ref: http://bit.ly/1f9tHTJ

```
NSDateFormatter * formatter = [[NSDateFormatter alloc] init];

NSDate * date = [NSDate date];

[formatter setDateFormat:@"YYYY-MM-DD"];
NSLog(@"%@",[formatter stringFromDate:date]);
```

Question

Custom Class

Michael Pan

struct 缺點

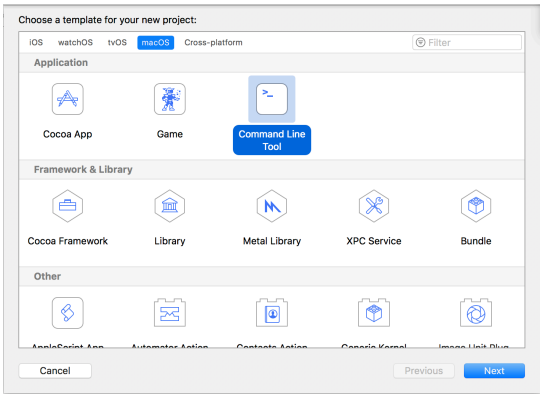
無法延伸功能

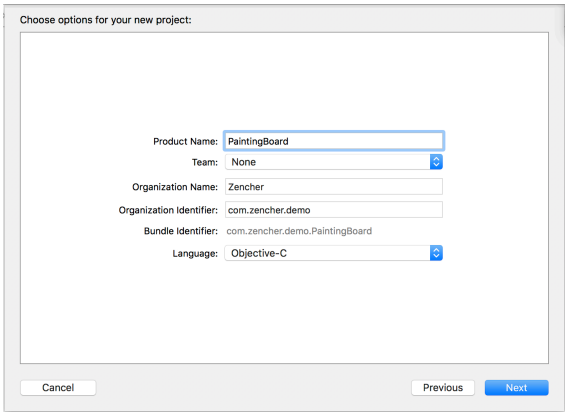
無論新增或是修改都要改寫原本的struct

建立一個 Command Line Tool

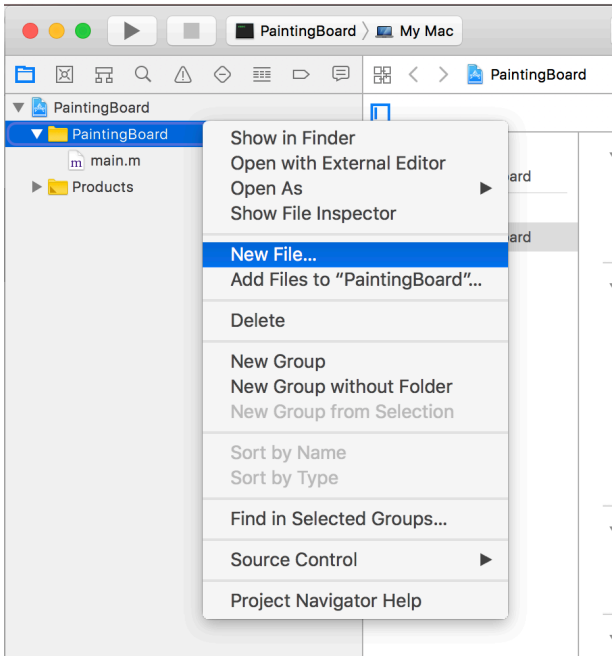
命名 PaintingBoard

選擇 Objective-C Language

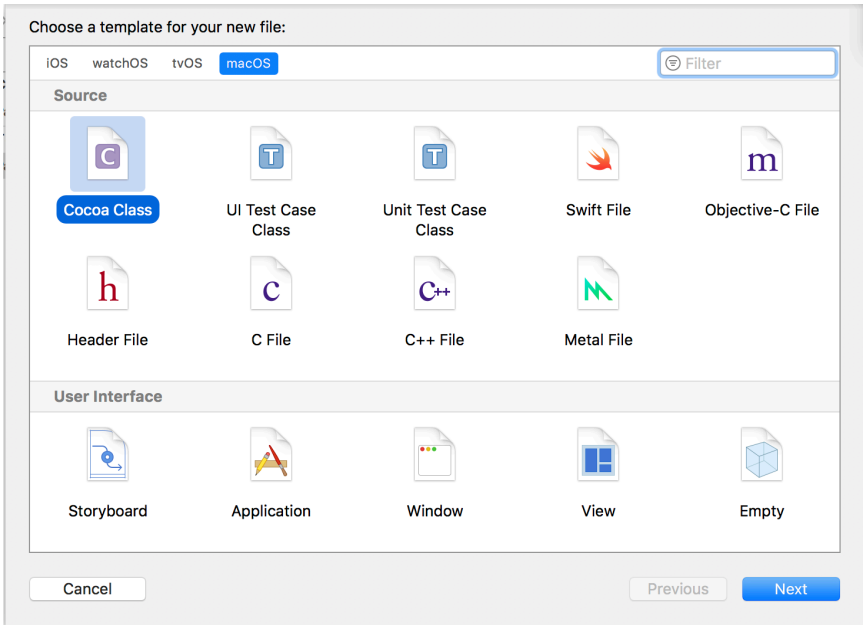




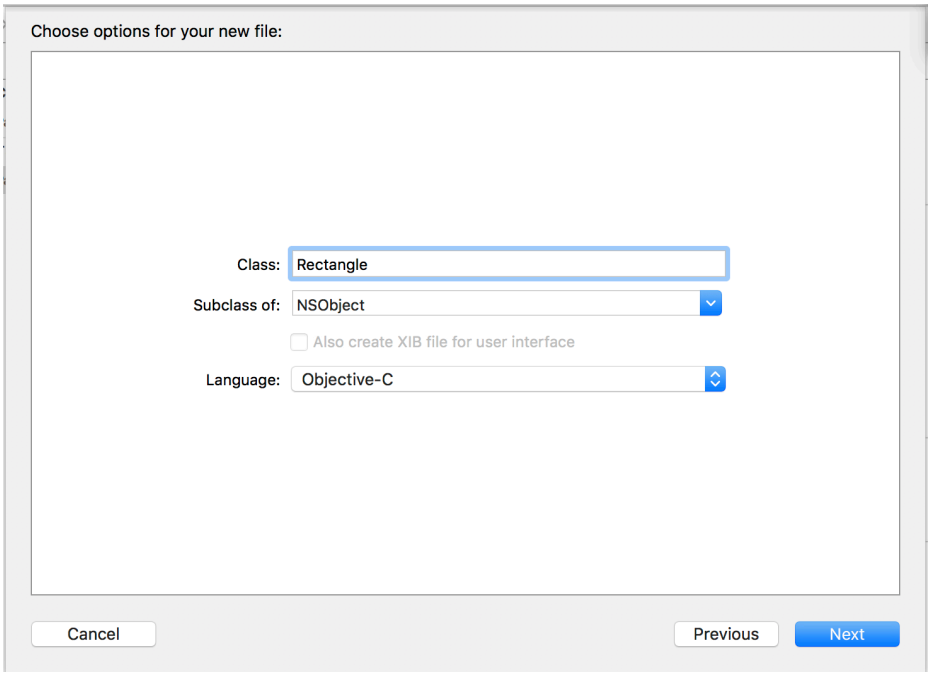
Create a new File



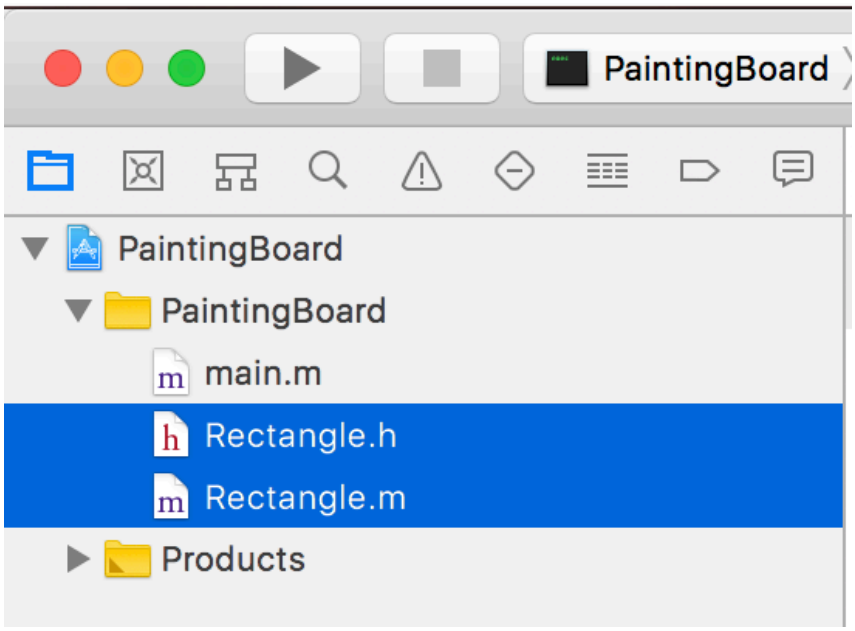
Cocoa Class - File



Name - Rectangle



New file - Rectangle.h/.m



Rectangle.h

```
@interface Rectangle : NSObject

@end
```

```
@interface

@end
```

用來 框架 class 的程式碼

Method

```
@interface Rectangle : NSObject
-(int) area;
@end
```

這是什麼 ？

直接把 area - method 宣告部分 放在 interface 裡面

記得沒有主體

C Function to method

```
int area()
```

```
-(int) area
```

Function to method - two parameters

```
int sum(int a, int b)
```

```
-(int) sum:(int) a with:(int) b;
```

```
sum:with:
```

selector => method name

Rectangle.m

```
#import "Rectangle.h"

@implementation Rectangle
-(int) area{
    return 100;
}
@end
```

一開始要 import .h 檔

implementation 就是放 area() 的主體

Use Rectangle in main.m

```
#import <Foundation/Foundation.h>

#import “Rectangle"

int main(int argc, const char * argv[]) {
    @autoreleasepool {
        Rectangle * rect = [Rectangle new];

        int r = [rect area:23];
        NSLog(@"%d",r);
    }
    return 0;
}
```

需要 import

用 new 產生實體

使用 method

Property - Rectangle.m

```
@implementation Rectangle{
    int width;
    int height;
}
-(int) area{
    return 100;
}
@end
```

Getter & Setter

如何在 main.m 存取 在 PlayGround 的 變數？

method for width and height

Getter & Setter

Getter & Setter - Interface

```
@interface Rectangle : NSObject
-(int) area;
-(int) width;
-(void) setWidth:(int) aWidth;
@end
```

Getter & Setter - implementation

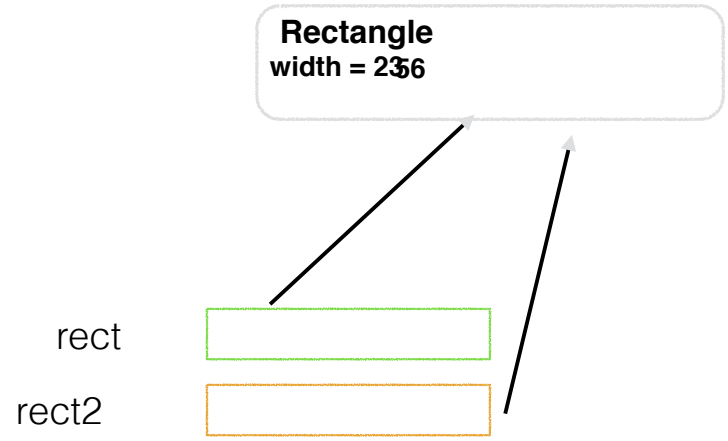
```
@implementation Rectangle{
    int width;
    int height;
}
-(int) width {
    return width;
}
-(void) setWidth:(int) aWidth {
    width = aWidth;
}
@end
```

Test Getter & Setter - main.m

```
Rectangle * rect = [Rectangle new];
rect.width = 500;
int r = rect.width;
NSLog(@"%d",r);
```

Object View

```
Rectangle * rect = [Rectangle new];

rect.width = 23;

Rectangle * rect2 = rect;

rect.question = 56;
```



@property - interface

```
@interface Rectangle : NSObject
-(int) area;
//-(int) width;
//-(void) setWidth:(int) aWidth;
@property int width;
@end
```

@property - implementation

```
@implementation Rectangle{
//    int width;
    int height;
}

-(instancetype) init {
    self = [super init];
    if (self) {
        _width = 200;
        height = 300;
    }
    return self;
}
//-(int) width {
//    return width;
//}
//-(void) setWidth:(int) aWidth {
//    width = aWidth;
//}

@end
```

Constructor - 建構子

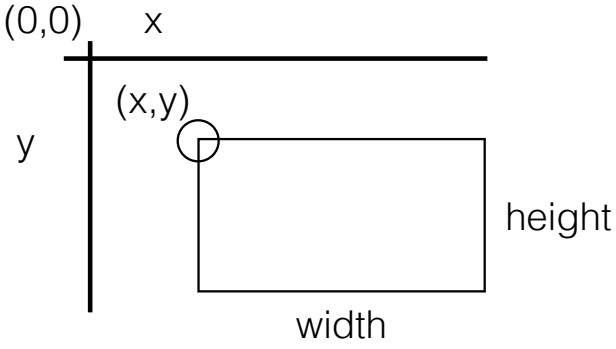
```
@implementation Rectangle{
    int width;
    int height;
}

-(instancetype) init {
    self = [super init];
    if (self) {
        width = 200;
        height = 300;
    }
    return self;
}
@end
```

Recap

在 class Rectangle 新增 4 個 ivar，分別是 x,y,width,height型別都是 float

產生一個 method 可以同時設定 x 和 y

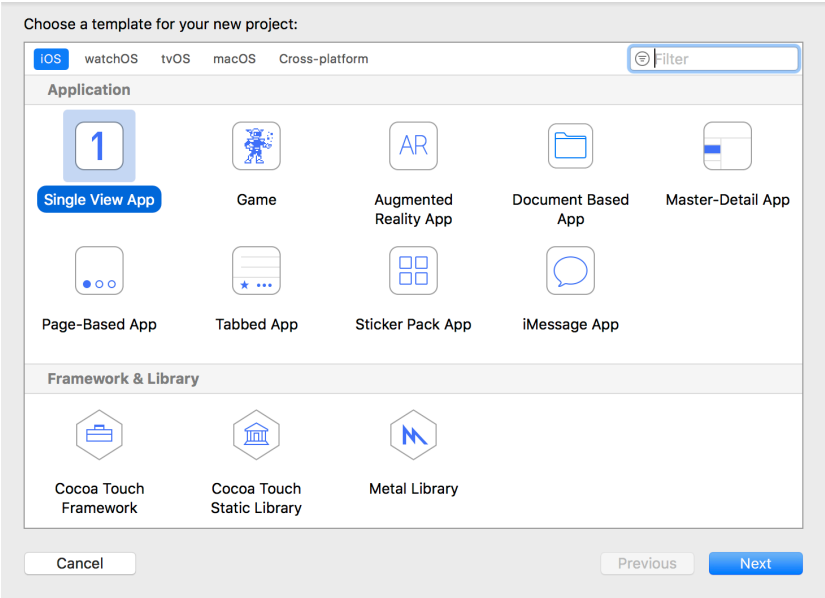
產生一個 center method 依據 x,y,width, height 的值可以用 NSLog 呈現 Rectangle 的中心位置



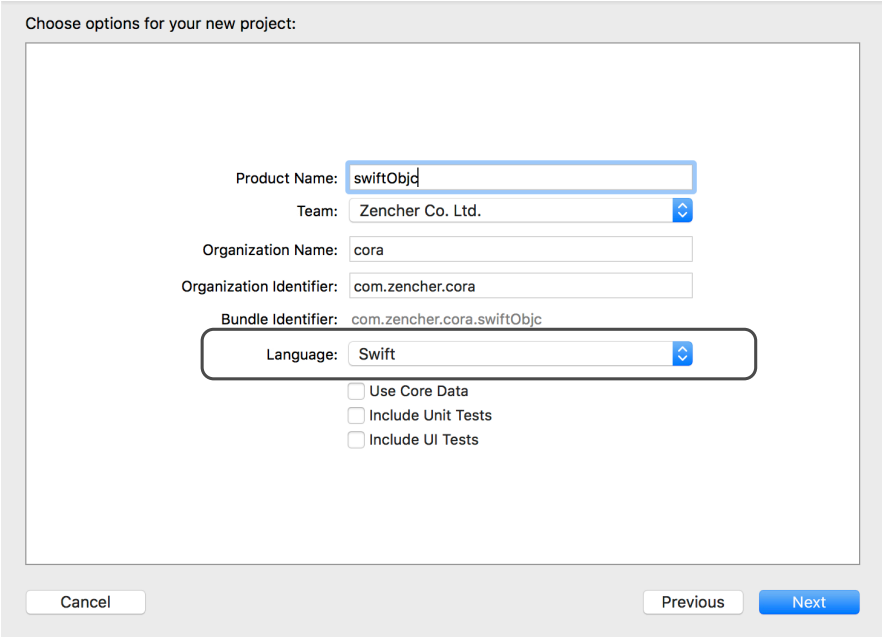
Swift Using Objective-C

Michael Pan

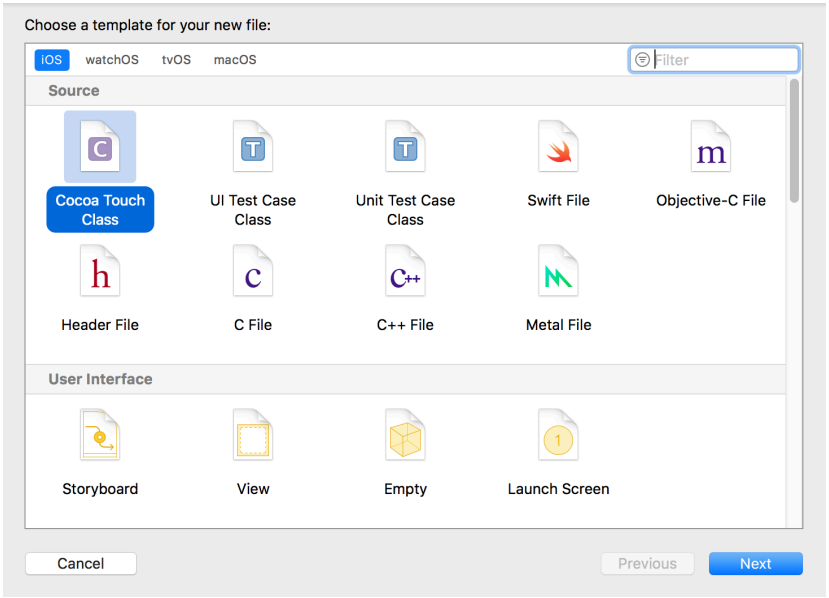
Create a Single View application



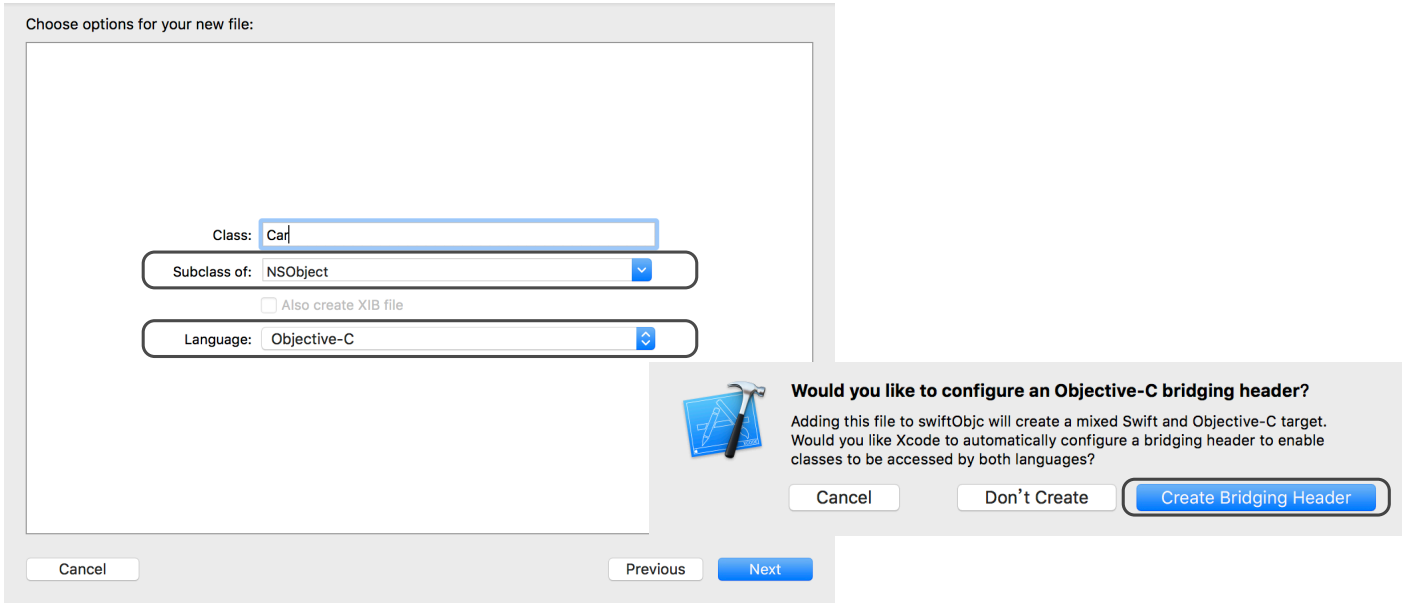
Choose Swift as Language



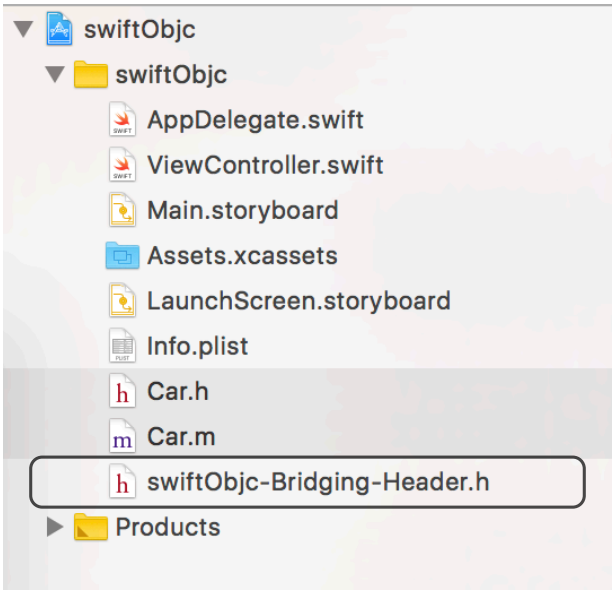
Create an Cocoa Touch class



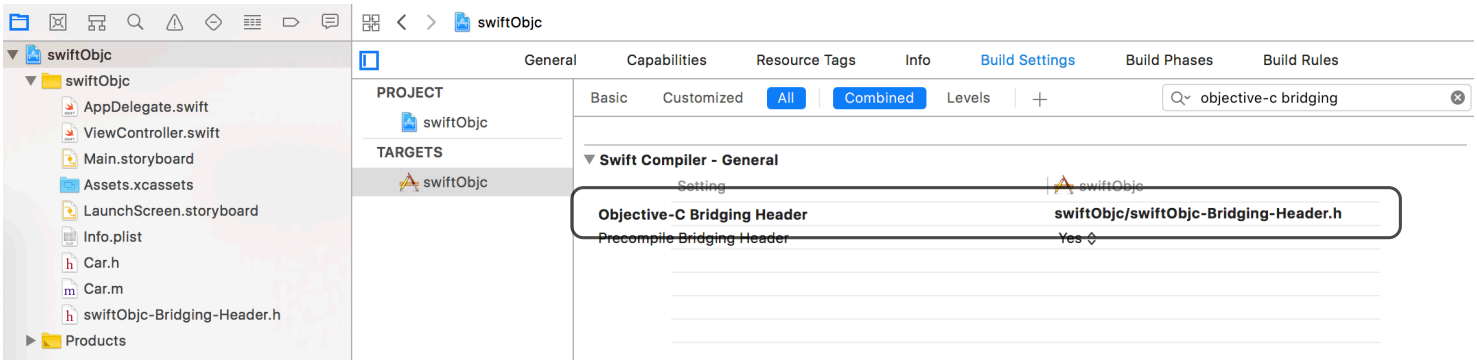
Car : NSObject



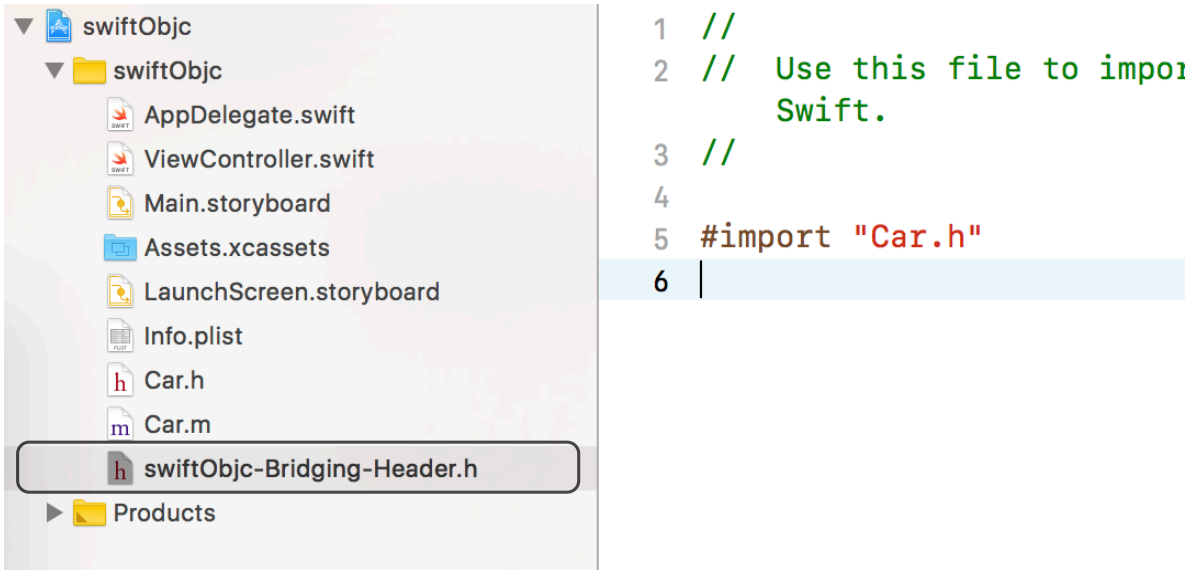
Bridge-Header.h



Manually



import Head.h in Bridge-Header.h



Use Car in ViewController

No import before using

```
import UIKit

class ViewController: UIViewController {

    override func viewDidLoad() {
        super.viewDidLoad()
        var car = Car()
    }
```

Add some method

Car.h

```
#import <Foundation/Foundation.h>

@interface Car : NSObject
-(void) setPrice:(int) aPrice andYear:(int) aYear;
@end
```

ViewController.swift

```
override func viewDidLoad() {
    super.viewDidLoad()
    var car = Car()
    car.setPrice(2000, andYear: 1990)
}
```

Run and Crash

-[Car setPrice:andYear:]: unrecognized selector sent to instance 0x60400000c980

Implementation in Car.m

```
#import "Car.h"

@implementation Car {
    int price;
    int year;
}

-(void) setPrice:(int) aPrice andYear:(int) aYear {

    price = aPrice;
    year = aYear;
}

@end
```

Selector

```
(setPrice:andYear:)
```

Objective-C

```
-(void) setPrice:(int) aPrice andYear:(int) aYear;
```

Swift

as second external name

```
car.setPrice(500, andYear: 1990)
```

Perform selector in swift

```
Timer.scheduledTimer(timeInterval: 1.0, target: self,
selector:#selector(ViewController.action(action:second:)),
userInfo: nil, repeats: false)
```

which will be called???

```
@objc func action(a:Timer, second b:Int ) {
        print("action1")
    }

    @objc func action(action a:Timer, second b:Int){
        print("action2")
    }
```

Run

Move ivar to Car.h

```
#import <Foundation/Foundation.h>

@interface Car : NSObject {
    int price;
    int year;
}

-(void) setPrice:(int) aPrice andYear:(int) aYear;

@end
```

Access ivar

```
9  import UIKit
10
11 class ViewController: UIViewController {
12
13     override func viewDidLoad() {
14         super.viewDidLoad()
15         var car = Car()
16         car.setPrice(2000, andYear: 1990)
17         car.price = 300          Value of type 'Car' has no member 'price'
18
```

Change ivar to Property

Car.h

```
11 @interface Car : NSObject {
12     int price;
13     int year;
14 }
15 @property int price;
16 @property int year;
17
18 -(void) setPrice:(int) aPrice andYear:(int) aYear;
19 @end
```

Car.m

```
@implementation Car
-(void) setPrice:(int) aPrice andYear:(int) aYear {
    self.price = aPrice;
    self.year = aYear;
}
@end
```

Run

```
override func viewDidLoad() {
    super.viewDidLoad()
    var car = Car()
    car.setPrice(2000, andYear: 1990)
    car.price = 300
    print("car price \(car.price)")
```

Class method in Objective-C

```
@interface Car : NSObject

+(instancetype) sharedInstance;
@end
```

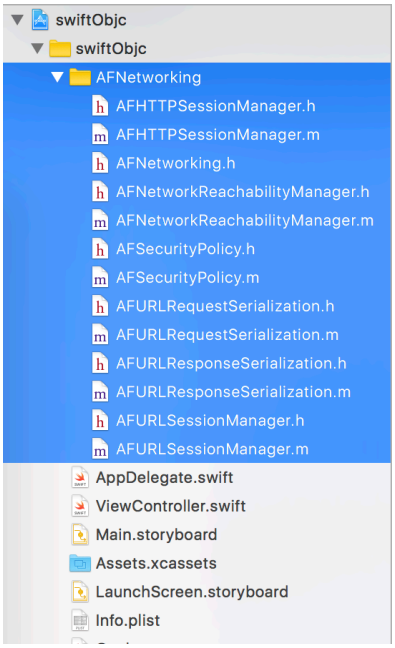
```
var myCar = Car.sharedInstance()
```

swift

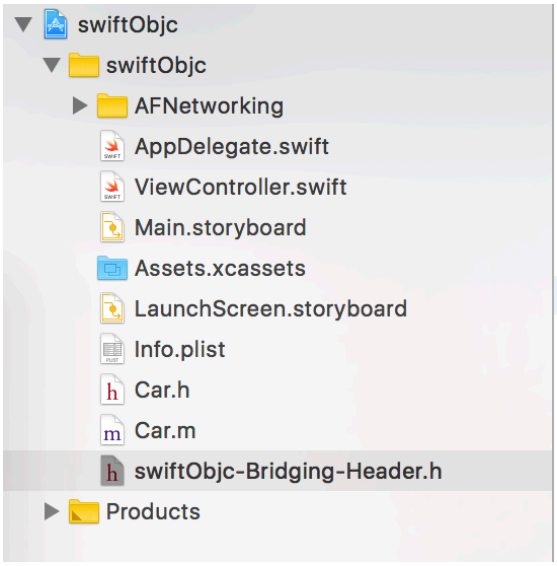
Use the famous third party project

AFNetworking

Drag Files into Project



Import AFNetworking in Bridge.h




```
1  //
2  //  Use this file to import you
       would like to expose to Swi
3  //
4
5  #import "Car.h"
6
7  #import "AFNetworking.h"
8
```

Test - AFNetworking

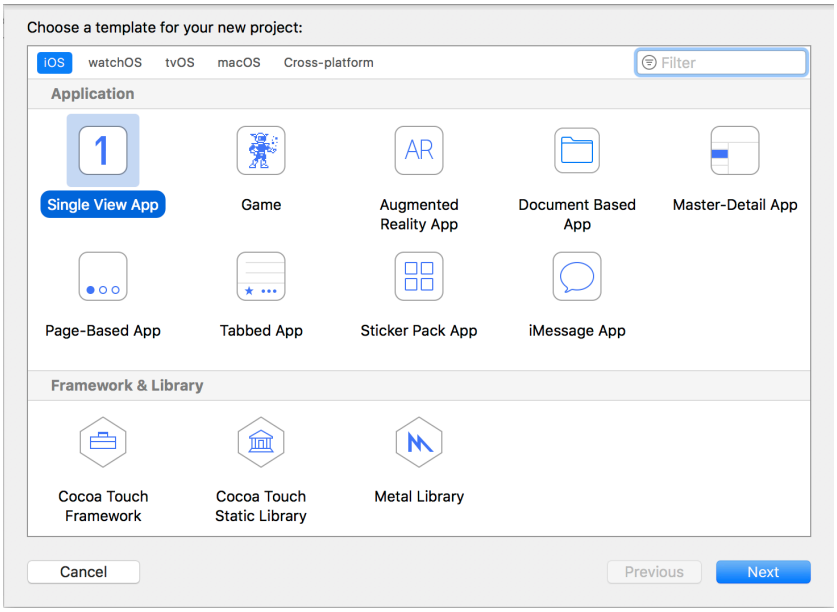
```
let configuration:URLSessionConfiguration = URLSessionConfiguration.default
        let manager:AFURLSessionManager = AFURLSessionManager(sessionConfiguration: configuration)

        let url:URL = URL(string: "https://i.pinimg.com/564x/08/57/0d/
                                   08570daccbdc9c8d5f9027a84be82b0b.jpg")!
        let request:NSURLRequest = NSURLRequest(url: url)

        let downloadTask:URLSessionDownloadTask = manager.downloadTask(with: request as URLRequest,
                                      progress: nil, destination: { (targetPath, response) -> URL in

            var documentsDirectoryURL:URL? = nil
            do {
            documentsDirectoryURL = try FileManager.default.url(for: .documentDirectory,
                                        in: .userDomainMask, appropriateFor: nil, create: false)
            } catch {
                print("Error")
            }

            return documentsDirectoryURL!.appendingPathComponent(response.suggestedFilename!)

        }) { (response, filePath, error) in
            print("File downloaded to: \(String(describing: filePath))")
        }

        downloadTask.resume()
```

Question

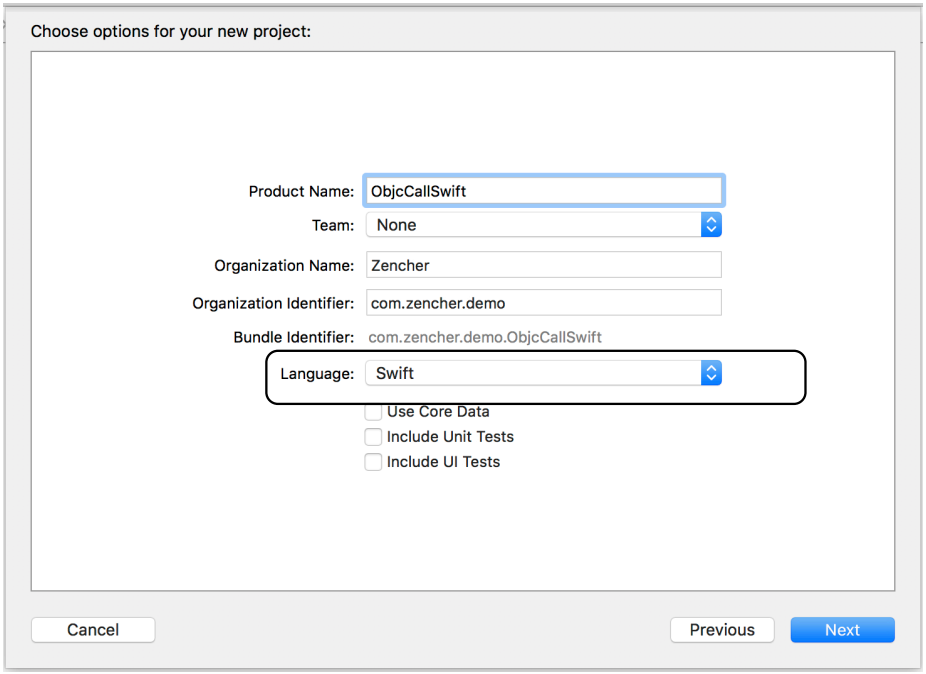
Objective-C Using Swift

Michael Pan

Create a single view application

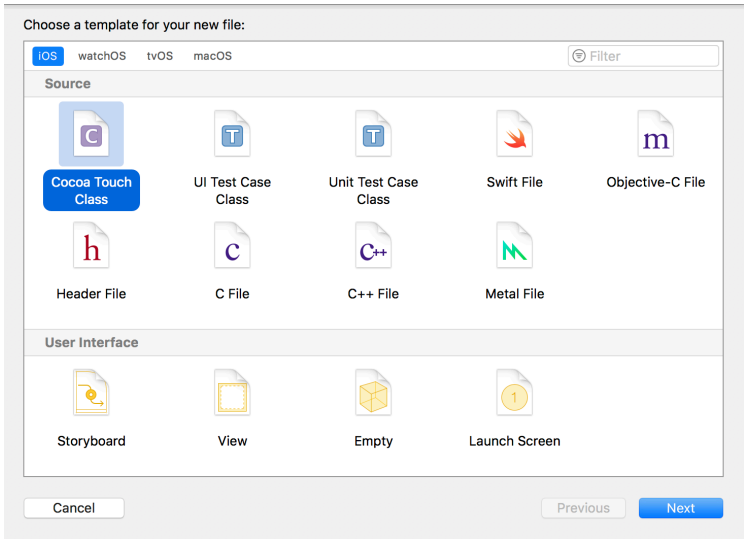


Swift Language

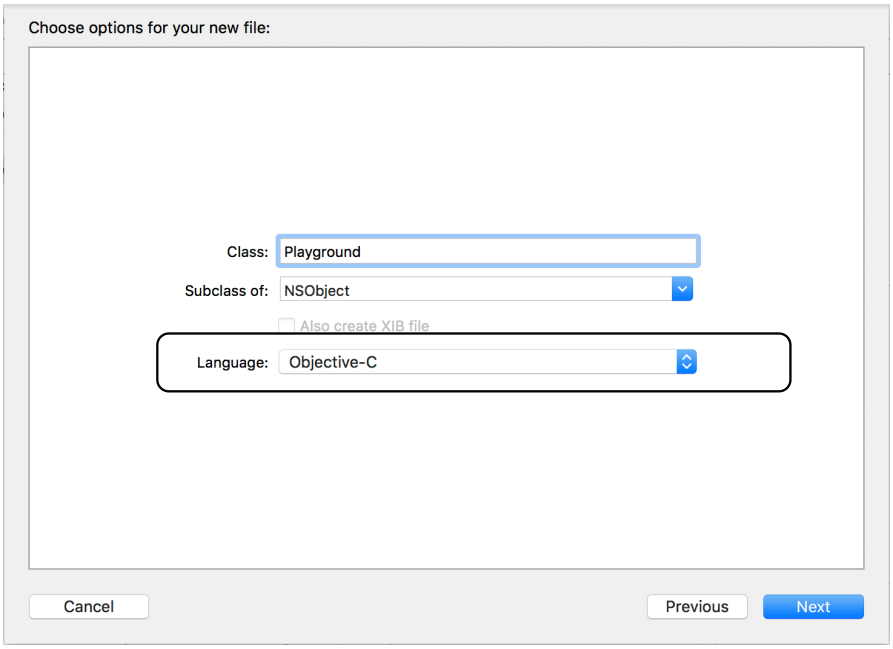


Create an Objective-C Class

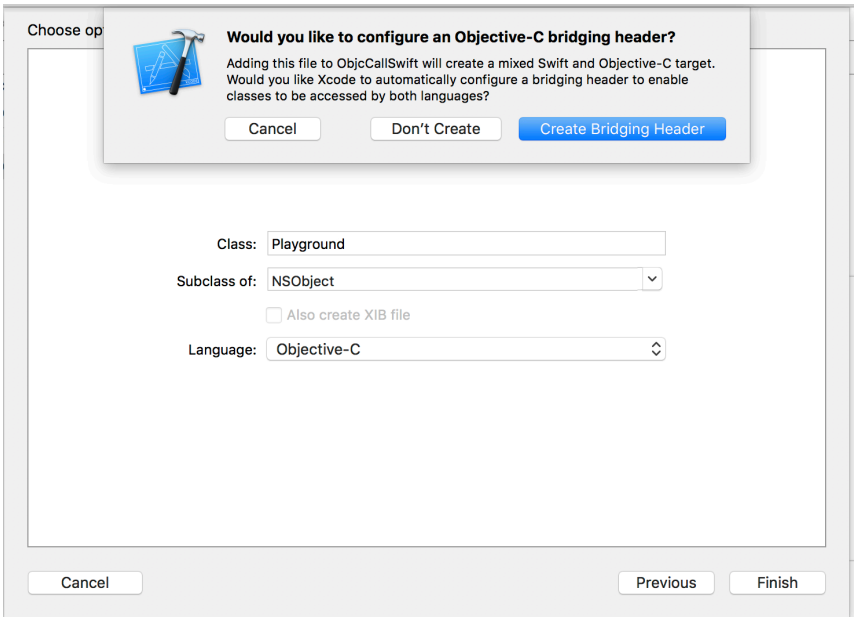
Cocoa Touch Class



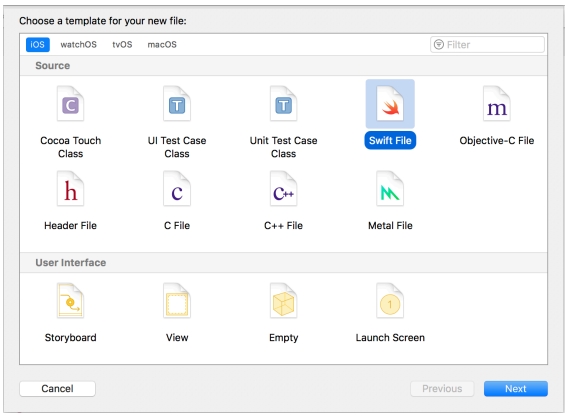
Playground : NSObject

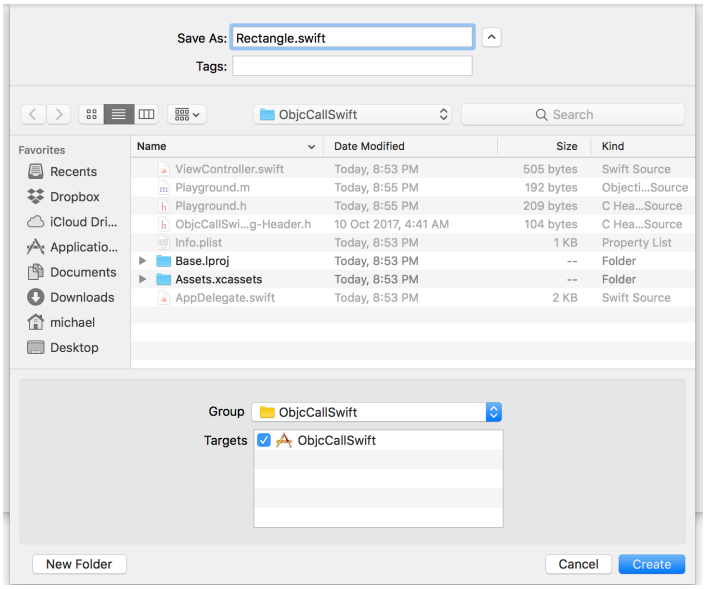


Create the bridge head file



Create a Swift Class - Rectangle





Rectangle.swift

```
import Foundation
class Rectangle : NSObject{
    var width = 300
    var height = 600
}
```

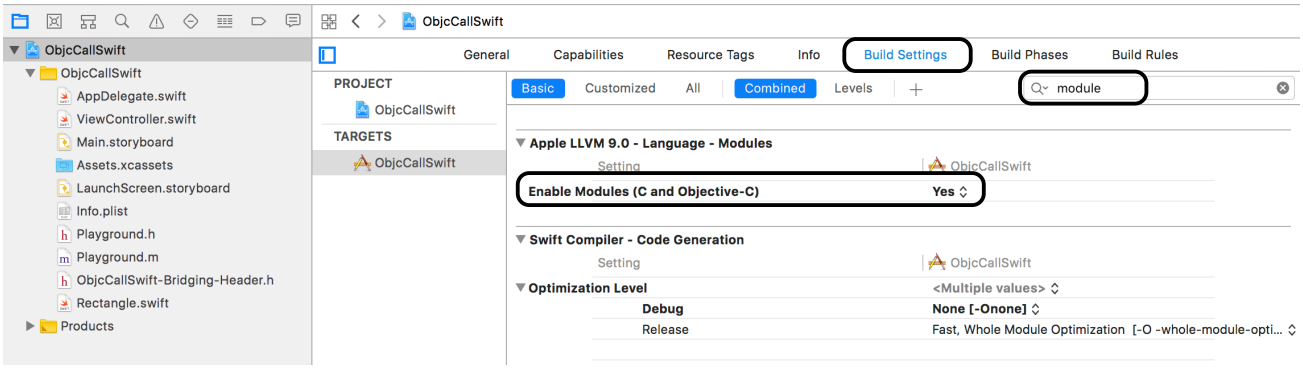
Make sure Swift class extends to NSObject

Or you can not alloc - init the Swift Class object

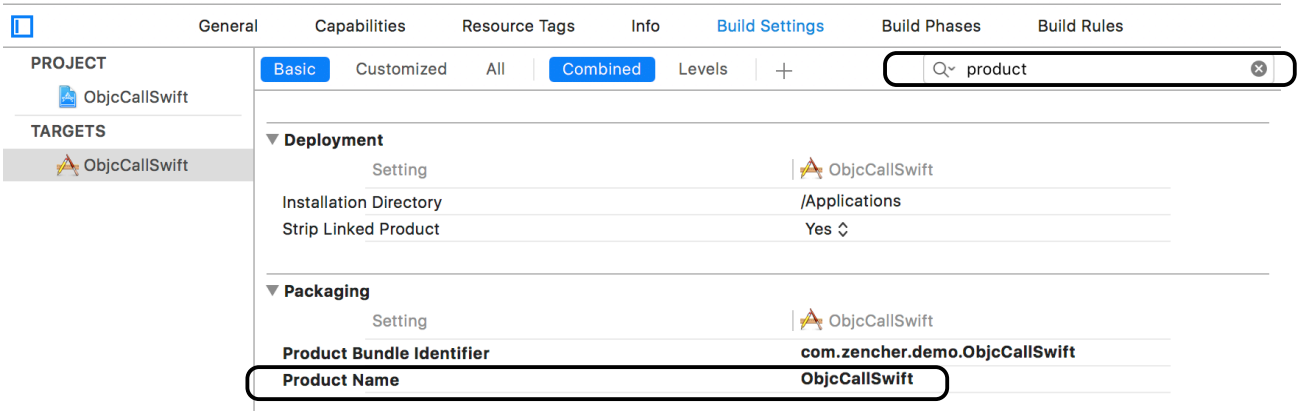
```
import Foundation
class Rectangle : NSObject{
    var width = 300
    var height = 600
}
```

Check settings

Enable Modules - Yes



Product name - ObjcCallSwift



Import Swift Class in Objective-C Implementation

you can not see but it is there

```
#import "Playground.h"
#import "ObjcCallSwift-Swift.h"
@implementation Playground
-(void) makeRect {
    Rectangle * rect = [Rectangle new];



}
@end
```

Access swift property

```
#import "Playground.h"
#import "ObjcCallSwift-Swift.h"
@implementation Playground
-(void) makeRect {
    Rectangle * rect = [Rectangle new];
    rect.width = 300;
    NSLog(@"%d",rect.width);
}
@end
```

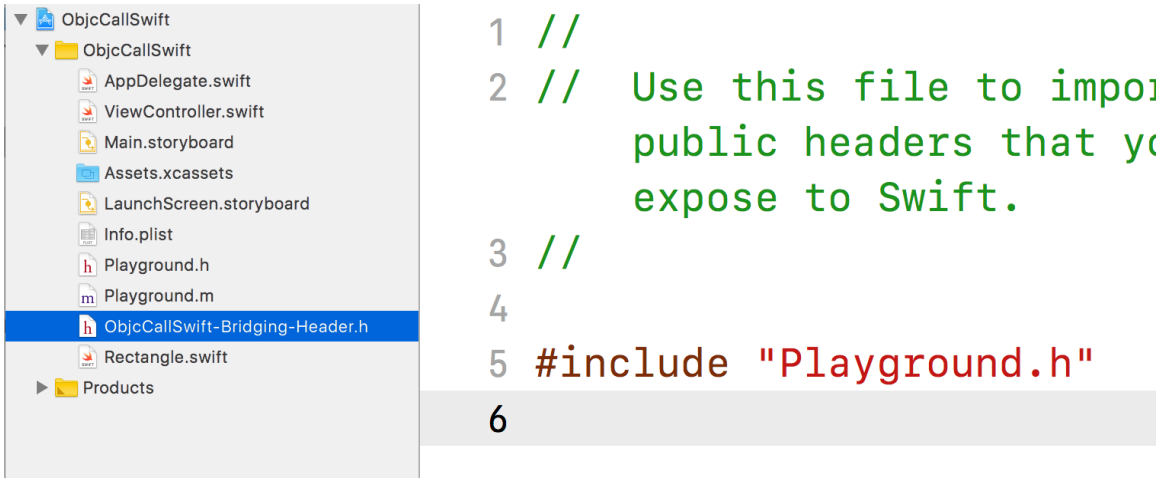
Error

@objc

```
class Rectangle : NSObject{
    @objc var width = 300
    @objc var height = 600
}
```

Use the Objective-C Class

Remember that - Bridge Header of Objective-C



Make a public method in Playground.h

```
@interface Playground : NSObject
-(void) makeRect;
@end
```

Test Playground object

```
class ViewController: UIViewController {

    override func viewDidLoad() {
        super.viewDidLoad()
        let playground = Playground()
        playground.makeRect()
    }
}
```

Run

Gesture

Tutor : Michael

UIGestureRecognizer

UI**Tap**GestureRecognizer

UI**Pinch**GestureRecognizer

UI**Rotation**GestureRecognizer

UI**Swipe**GestureRecognizer

UI**Pan**GestureRecognizer

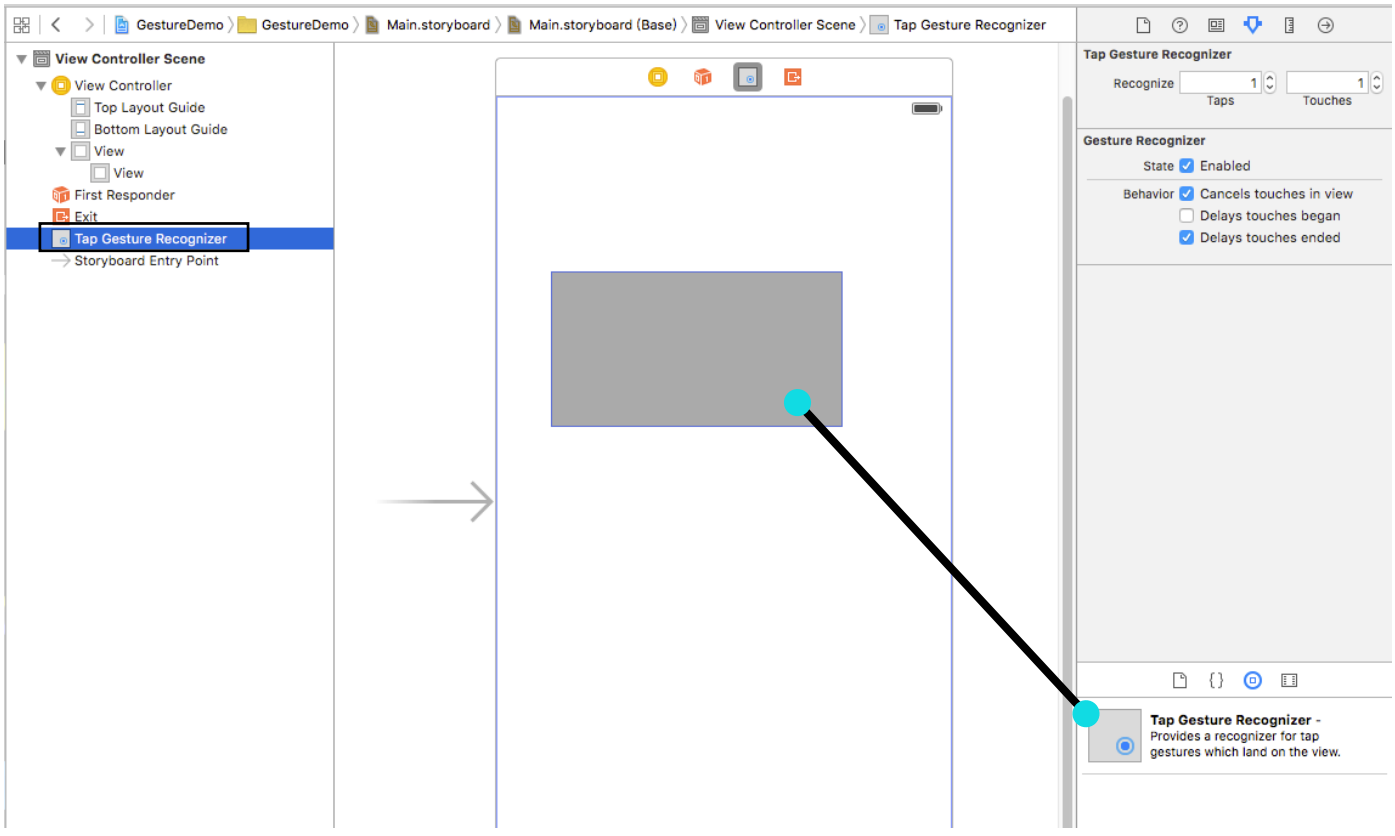
UI**LongPress**GestureRecognizer

Explain

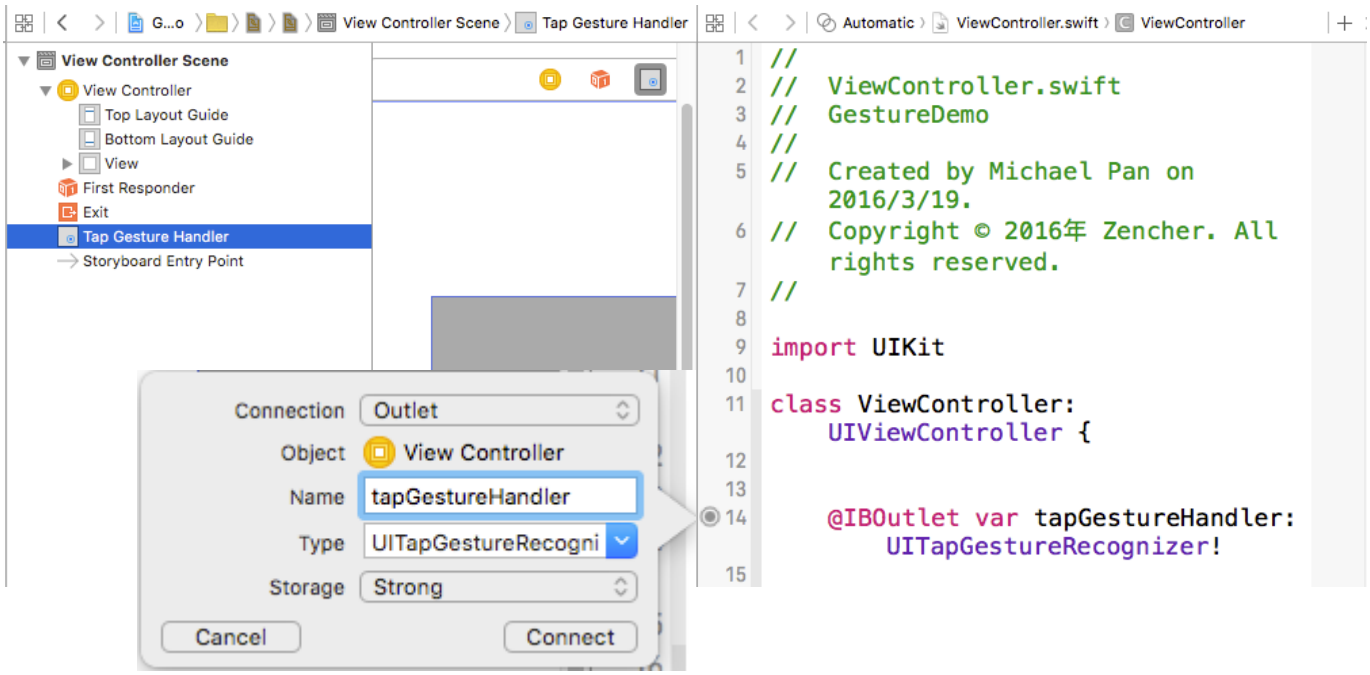
Touch - 名詞，接觸點

Tap - 動作，點下去

Visual Component



IBOutlet



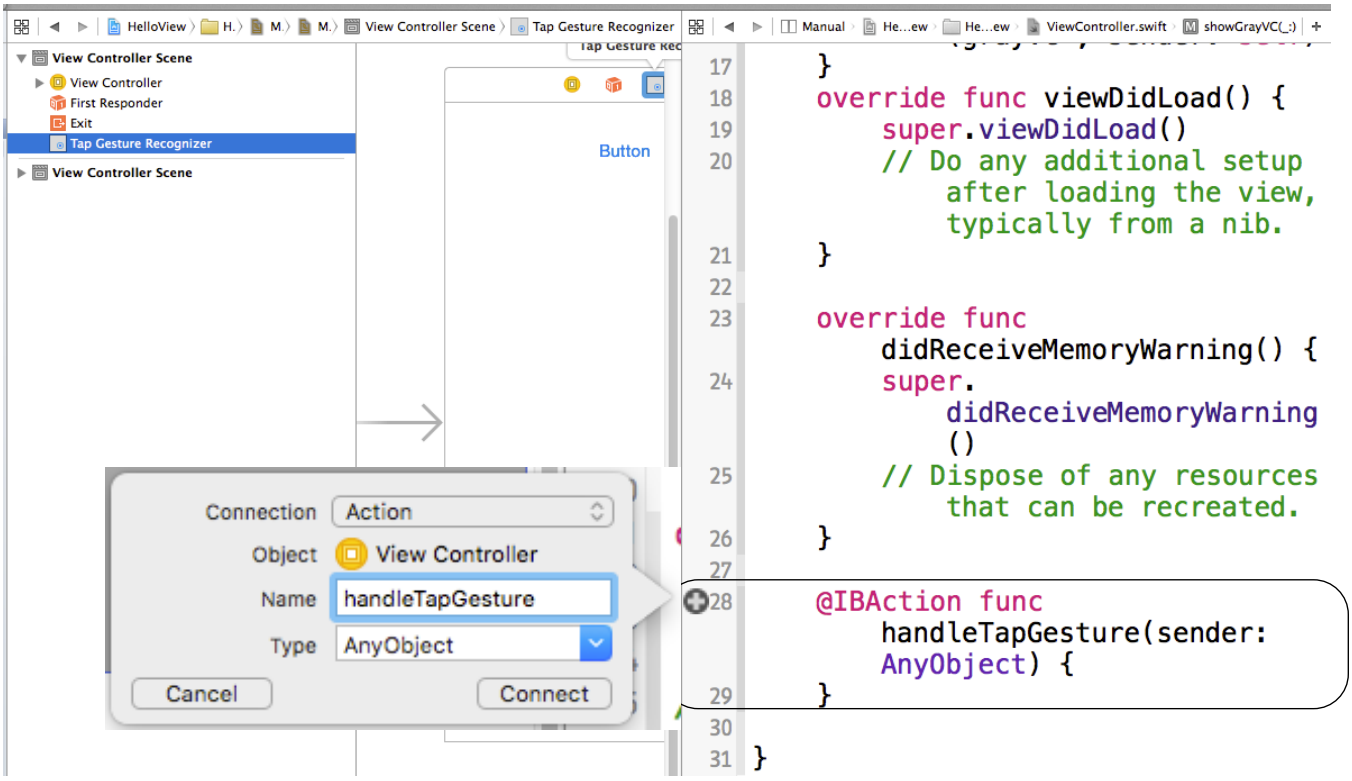
Tap

numberOfTapsRequired

numberOfTouchesRequired

```
override func viewDidLoad() {
 super.viewDidLoad()
 tapGestureHandler.numberOfTapsRequired = 2
 tapGestureHandler.numberOfTouchesRequired = 2


}
```

Gesture Connection



Tap - handleTapFrom:

```
@IBAction func handleTapGesture(sender:
UITapGestureRecognizer) {
        print("\(sender.locationInView(self.view))")

    }
```

Demo

-Tap Gesture

State

```
public enum UIGestureRecognizerState : Int {

    case Possible // the recognizer has not yet recognized its gesture, but may be
evaluating touch events. this is the default state

    case Began // the recognizer has received touches recognized as the gesture. the
action method will be called at the next turn of the run loop
    case Changed // the recognizer has received touches recognized as a change to
the gesture. the action method will be called at the next turn of the run loop
    case Ended // the recognizer has received touches recognized as the end of the
gesture. the action method will be called at the next turn of the run loop and the
recognizer will be reset to UIGestureRecognizerStatePossible
    case Cancelled // the recognizer has received touches resulting in the
cancellation of the gesture. the action method will be called at the next turn of
the run loop. the recognizer will be reset to UIGestureRecognizerStatePossible

    case Failed // the recognizer has received a touch sequence that can not be
recognized as the gesture. the action method will not be called and the recognizer
will be reset to UIGestureRecognizerStatePossible

    // Discrete Gestures – gesture recognizers that recognize a discrete event but
do not report changes (for example, a tap) do not transition through the Began and
Changed states and can not fail or be cancelled
    public static var Recognized: UIGestureRecognizerState { get } // the recognizer
has received touches recognized as the gesture. the action method will be called at
the next turn of the run loop and the recognizer will be reset to
UIGestureRecognizerStatePossible
}
```

State - Swipe

Began Changed Ended

Long Press

minimumPressDuration

```
var longPressGesture = UILongPressGestureRecognizer()
longPressGesture.minimumPressDuration = 3.0

self.grayView.addGestureRecognizer(longPressGesture)
longPressGesture.addTarget(self, action:
Selector("handleLongPress:"))
```

states

```
func handleLongPress(sender:UILongPressGestureRecognizer){
        switch sender.state {
        case .began :
            print("began long press")
        case .changed :
            print("changeed long press")
        case .ended :
            print("ended long press")
        default:
            print("ignore..")
        }
    }
```

Demo

-Long Press Gesture

Swipe

direction

numberOfTouchesRequired

```
let swipeGesture = UISwipeGestureRecognizer()
swipeGesture.direction = .left
swipeGesture.numberOfTouchesRequired = 1
swipeGesture.addTarget(self, action: "handleSwipe:")
self.grayView.addGestureRecognizer(swipeGesture)
```

Handler

```
func handleSwipe(sender:UISwipeGestureRecognizer){
 switch sender.state {
 case .possible:
  print("possible")
 case .began:
  print("began")
 case .ended:
  print("ended")
  // testing
 default:
  print("ignore..")
 }

}
```

Demo

-Swipe Gesture

Rotate

CGFloat rotation

CGFloat velocity

```
let rotateGesture = UIRotationGestureRecognizer()
rotateGesture.addTarget(self, action: "handleRotate:")

self.grayView.addGestureRecognizer(rotateGesture)
```

Rotate - handleRotationFrom:

```
func handleRotate(sender:UIRotationGestureRecognizer){
 let transform =
 CGAffineTransformMakeRotation(sender.rotation);
 self.grayView.transform = transform;

}
```

Demo

-Rotate Gesture

Pinch

CGFloat scale

CGFloat velocity

```
let pinchGesture = UIPinchGestureRecognizer()
pinchGesture.addTarget(self, action: "handlePinch:")
self.grayView.addGestureRecognizer(pinchGesture)
```

Pinch - handlePinchFrom:

```
func handlePinch(sender:UIPinchGestureRecognizer){
 self.grayView.transform =
 CGAffineTransformMakeScale(sender.scale, sender.scale);
}
```

Demo

-Pinch Gesture

Pan

maximumNumberOfTouches

minimumNumberOfTouches

```
let panGesture = UIPanGestureRecognizer()
panGesture.addTarget(self, action: "handlePan:")
panGesture.minimumNumberOfTouches = 2
self.grayView.addGestureRecognizer(panGesture)
```

Pan - handlePanFrom:

```
func handlePan(sender:UIPanGestureRecognizer){
 UIView.animateWithDuration(0.1) { () -> Void in
  self.grayView.center = sender.locationInView(self.view)
 }

}
```

Demo

-Pan Gesture

Question ?