

Swift course notes

Complete iOS 10 developer course

<https://www.udemy.com/the-complete-ios-10-developer-course-beginner-to-advanced/learn/v4/questions/>

XCode

2-7 Ctrl-drag to code window to connect

Connect could be of type Outlet or Action or OutletCollection (eg for Button) - action for whatever is to happen - usually placed within ViewController class

```
import UIKit
```

2-8 XCode/Preferences - object.method/property - eg. @IBAction func
changeButton(_sender: Any) { labelPJ.text = "Hello" }

skipping some lessons on Basics of Swift

3-19 Classes -

Playground for 'new ideas'

choice of iOS, macOS or tvOS

// e.g.

```
class User {  
    var name = ""  
    var age = 0  
    var favouriteMovie = ""  
    var favouriteWeather = ""  
    var isDeveloper = false
```

```
}
```

```
var newUser = User()
```

```
newUser.name = "Dashiell Something"
```

```
newUser.age = 30
```

```
newUser.favouriteMovie = "Pulp Fiction"
```

```
newUser.favouriteWeather = "cloudy"
```

```
newUser.isDeveloper = false
```

3-20 Optionals

like nullable variables in C# , suffix '?' to avoid error if variable not set ie 'nil' -

to be unwrapped (not knowing if any contents in advance)

Alt+3 for hash symbol

//e.g.

```
//var myName : String = "James"  
var myName : String?      // optional by '?' - nullable  
myName?.uppercase()
```

<https://cocoacasts.com/what-does-an-exclamation-mark-mean-in-swift/>

TODO: optional binding

optional chaining e.g. person?.sayHello()

person?.name = "John" // not sure if property or method d'exist

Implicitly unwrapped optional - '!' - if certain that values will exist - crash otherwise

```
class Person {  
    var name: String!  
  
    func sayHello() {  
        print("hello")  
    }  
}
```

```
let person = Person()  
person.name = "John"
```

```
print(person.name)
```

3-22

XCode Interface Builder GUI libraries - File template, Code snippet, Object, Media

Labels &c in Object, mini-patterns/expressions in Code snippet

right-sidebar Utilities Area has: File Inspector, Quick Help, Identity Inspector, Attributes Inspector, Size Inspector, Connections Inspector

File Inspector: XCode project details

Quick Help,

Identity Inspector: object class, GUID

Attributes Inspector: bespoke to object eg label text, font

Size Inspector: on-screen size

Connections Inspector: outlets & actions / connections for object (event stuff)

2-7

Alt + drag to copy

Assistant Editor oo to see code

Version editor <=> to see XML

Assistant Editor should open ViewController.swift

Ctrl-drag as in 2-7 to connect Text box to class ViewController

(may need to open Main.storyboard in separate window and drag from this window to ViewController.swift, to connect from text box, eg.)

Alternatively Ctrl-drag from ViewController Scene/View/individual components listed

e.g. @IBOutlet weak var firstNumberTxt: UITextField!

```
@IBAction func plusBtn(_ sender: Any) {  
}
```

NB weak, strong, unowned - to do with memory management - Automatic Reference Counting for classes (not structs) deallocated when ARC count zero - strong is default - strong increment ARC by 1 - weak won't increment ARC, automatically set to nil on deallocation, useful for possible 2-way/circular references, must be Optional - unowned doesn't increment ARC and need not be Optional, not automatically set to nil on deallocation, should only be used if object never reset to nil - https://medium.com/@chris_dus/strong-weak-unowned-reference-counting-in-swift-5813fa454f30

viewDidLoad is default method when view has loaded

<https://www.tech-recipes.com/rx/52021/how-do-i-fix-the-issue-this-class-is-not-key-value-coding-compliant-for-the-key-in-xcode-6/>

if error e.g.

label.text assign throwing EXC_BAD_INSTRUCTION (code=exc_i386_invop
subcode=0x0)

try deleting the reference to the object (e.g. myLabel) and then (double-clicking Main storyboard to get draggable control) re-drag from object/control (e.g. label) into code to re-make Outlet

<https://stackoverflow.com/q/24091892>

my try at catching non-number inputs:

```
@IBAction func plusBtn(_ sender: Any) {  
  
    if let _ = Int(firstNumberTxt.text!) , let _ = Int(secondNumberTxt.text!)  
    { // try to catch non-number string input https://stackoverflow.com/a/38159489  
        result = Int(firstNumberTxt.text)! + Int(secondNumberTxt.text)!  
        // implicitly unwrapped optional - need to be certain that values will  
exist  
        resultLabel.text = String(result)  
    } else  
    {  
        resultLabel.text = "Please type numbers"  
    }  
  
}
```

3-23 string interpolation \(myVar)
eg "Result is \(String(result))"

sketchappsources.com

5-29 Layout resizing (in code) for devices

Cmd+R run

eg layout ok in iphone 6 & 7 but off-screen in iphone 5, e.g.

5-30 'Align' button at bottom e.g. 'horizontally in container', 'top edges x pixels' 'vertical centers' &c
click tick box and 'add 1 constraint'
red indicating error (see message) eg 'need constraint for Y position'
so...
'Add Constraint' button at bottom - values can be entered for width/height/
margins &c

'Resolve auto-Layout' button at bottom can fill in values, or option to 'Clear
Constraints'

remove constraints if clash (manually or Editor/Resolve Auto-Layout Issues...)

5-31 Project Layout can be set to a device type e.g. iphone 5

'Resolve auto-Layout' button at bottom can fill in values, as per 'Reset to Suggested Constraints'

Size Inspector control at top[-right in Utilities has an area for Constraints -

5-33. programmatic placement of objects/controls

```
override func viewDidLoad() {  
    super.viewDidLoad()  
    // Do any additional setup after loading the view, typically from a nib.  
  
    // default action on loading  
  
    let width = view.frame.size.width  
    let height = view.frame.size.height  
  
    let myLabel = UILabel() // let not var as Label will not be changed to  
    another type (immutable) https://stackoverflow.com/a/24002209 https://stackoverflow.com/a/31651715  
    myLabel.text = "My code-written label"  
    myLabel.textAlignment = NSTextAlignment.center // key-in '!' to get  
options  
    myLabel.textAlignment = .center  
    //myLabel.frame = CGRect(x:10, y:10, width: 100, height:50)  
    myLabel.frame = CGRect(x:width * 0.1, y: height * 0.3, width: width * 0.8,  
height:50)  
    view.addSubview(myLabel)  
}
```

5-34. Adding button and click event, programmatically

Alt+3 for hash symbol

Alt+drag to copy control/object

6-37 UserDefaults to persist data

```
@IBAction func saveButtonClicked(_ sender: Any) {  
    UserDefaults.standard.set(nameText.text, forKey: "name")  
    UserDefaults.standard.set(birthdayText.text, forKey: "birthday")  
    UserDefaults.standard.synchronize()  
    // persist data in key/value pairs using UserDefaults https://developer.apple.com/documentation/foundation/userdefaults
```

```
cmd+R run  
cmd+. stop  
cmd+B build
```

6-39

```
if let _ = storedDate as? String {    // underscore with let to avoid warning,  
compiler understands as temporary constant (!)  
    // if possible to create storedName, cast to String, from (hopefully non-  
nil) value retrieved...  
    UserDefaults.standard removeObject(forKey: "birthday")  
    UserDefaults.standard.synchronize()  
    storedDateLabel.text = "Stored date: "  
}
```

6-40 another view

1. drag another ViewControl onto the storyboard
2. File/New/Cocoa Touch Class (e.g.), make a subclass of 'UI View Controller'
 - place new file in same directory
3. Select second ViewController in storyboard (left-most icon at top of viewController object), in Utilities (on right) select Identity Inspector, to name storyboard object after code ViewController, and to connect object to object in code in new .swift file.

connections can then be made between the storyboard objects and their owning class (View Controller) in the code

NB - .xib - XML Interface Builder - compiled into .nib NeXT Interface Builder when built - binary - <https://www.c-sharpcorner.com/UploadFile/d49768/difference-between-xib-and-nib-in-iphone/>.

<https://developer.apple.com/library/archive/documentation/General/Conceptual/DevPedia-CocoaCore/NibFile.html>

Cocoa - [https://en.wikipedia.org/wiki/Cocoa_\(API\)](https://en.wikipedia.org/wiki/Cocoa_(API)). -

6-41 segue between ViewController screens in an app

Ctrl+drag from button to other ViewController in storyboard d'give menu 'Action Seque' with e.g. Show, Show Detail, Present Modally, Present as Popover, ...

also segue from left-most ViewController icon at top (Ctrl+drag)

using e.g. "fromFirstToSecond" name Segue in Identity Inspector

```
@IBAction func saveBtnClicked(_ sender: Any) {  
    performSegue(withIdentifier: "fromFirstToSecond"  
}
```

6-42 data between ViewControllers

in 1st ViewController:

```
override func prepare(for segue: UIStoryboardSegue, sender: Any?) {  
    if segue.identifier == "fromFirstToSecond" // NB case-sensitive  
    {  
        let destinationVC = segue.destination as! SecondViewController  
        // 'as!' forced/risky casting/conversion https://developer.apple.com/swift/blog/?id=23  
        // SecondViewController (segue's destinationVC) will be a constant  
        // ... giving access to variables & other members of segue destination  
        ViewController  
        // all via prepare  
        destinationVC.userName = name // use local var  
        print("preparing- " + name)  
    }  
}  
  
// execute segue previously defined on storyboard  
@IBAction func saveBtnClicked(_ sender: Any) {  
    name = nameText.text!  
    // record in local variable when clicked (NB had to use implicit '!')  
    performSegue(withIdentifier: "fromFirstToSecond", sender: nil)  
}
```

in 2nd ViewController:

```
nameLabel.text = "Your name is: \(userName)"
```

to return from a segue, in 2nd ViewController:
self.dismiss(animated: true, completion: nil)

6-43. alerts

```
let alert = UIAlertController(title: "Error", message: "oh dear",
preferredStyle: UIAlertControllerStyle.alert)

let okButton = UIAlertAction(title: "OK", style: UIAlertActionStyle.cancel,
handler: nil)
alert.addAction(okButton)
self.present(alert, animated: true, completion: nil)
```

6-44

6-47 ImageViews

```
imageViewActual.image = UIImage(named: "hendrix.jpeg")
@IBOutlet weak var imageViewActual: UIImageView!
```

Stack view. - easier formatting viz constraints - <https://www.raywenderlich.com/508-uistackview-tutorial-introducing-stack-views>

<https://developer.apple.com/library/archive/documentation/UserExperience/Conceptual/AutolayoutPG/LayoutUsingStackViews.html>
subviews stacked, horizontally/vertically

6-49 Gesture recognition -

```
super.viewDidLoad()
// Do any additional setup after loading the view, typically from a nib.
```

```
imageView.isUserInteractionEnabled = true
let gestureRecognizer = UITapGestureRecognizer(target: self,
    action:#selector(ViewController.changePicture))
// self is ViewController; NB hash to mark as selector for an action
imageView.addGestureRecognizer(gestureRecognizer)
```

NB for 'gesture recognition', 3 steps: (1) declare/define a UITapGestureRecognizer recogniser, (2) .addGestureRecognizer to ImageView, (3) ensure imageView.isUserInteractionEnabled = true
(4) write code for selector function ie ViewController.changePicture so func changePicture(){....}

ie gesture, recognizer & enabled interaction (1-3)

```

6-51 timers
override func viewDidLoad() {
    super.viewDidLoad()
    // Do any additional setup after loading the view, typically from a nib.

    counter = 10
    timerLabel.text = String(counter)

    timer = Timer.scheduledTimer(timeInterval: 1, target: self, selector:
#selector(ViewController.timerFunction), userInfo: nil, repeats: true)
        // 1 second, repeating, select timerFunction as event on timer 'tick'
}

func timerFunction()
{
    counter -= 1
    timerLabel.text = String(counter)
    print("Timer is running" + String(counter))
    if counter == 0
    {
        timer.invalidate()
        timerLabel.text = "Time's up!"
        // stop counting down at zero
    }
}

```

7-54 layout of game view (& connection into code)

7-55

9 images added and copy/paste/alter code for each as per steps 1-3 of 6-49
re gestures

NB TODO Instrument: profiling template, Activity monitor, File Activity, Cocoa Layout, Leaks <http://www.spotlesscode.com/blog/posts/instruments-tutorial-part-1-profiling-templates-deferred-mode-launch-instruments>

7-56 & 7-57 - timer as per 6-51, and alert as per 6-53

NB alerts have an UIAlertController, and 1 or more UIAlertAction to handle user response (eg 'ok')

NB Swift documentation comments

<https://useyourloaf.com/blog/swift-documentation-quick-guide/>

Separate timer so image hidden in intervals different from 1-second count-down (ie faster movement)

```

/// hide all kennys but 1
func hideKenny()
{
    for kenny in kennyImageArray
    {
        kenny.isHidden = true
    }

    let randomNumber =
Int(arc4random_uniform(UInt32(kennyImageArray.count - 1)))

    kennyImageArray[randomNumber].isHidden = false // un-hide 1
randomly selected image
}

```

7-58 Extra button added to dialogue (extra action added to Alert Controller) - and handler code written for this action to reset values so that user can 'replay'

```

func countDown()
{
    counter -= 1
    timeLabel.text = "\(counter)"

    // hideKenny()

    if counter == 0
    {
        timer.invalidate()
        hideTimer.invalidate()

        let alert = UIAlertController(title: "Time", message: "Time's up!",
preferredStyle: UIAlertControllerStyle.alert)
        let ok = UIAlertAction(title: "OK", style: UIAlertActionStyle.default,
handler: nil)
        alert.addAction(ok)

        let replay = UIAlertAction(title: "Replay", style:
UIAlertActionStyle.default, handler: {
            (UIAlertAction) in

            self.score = 0 // in handler block so refer back out to
ViewController class
            self.scoreLabel.text = "Score: \(self.score)"
            self.counter = 30
        })
    }
}

```

```

        self.timeLabel.text = "\(self.counter)"
        self.timer = Timer.scheduledTimer(timeInterval: 1, target: self,
selector: #selector(ViewController.countDown), userInfo: nil, repeats:
true)
        self.timer = Timer.scheduledTimer(timeInterval: 0.5, target: self,
selector: #selector(ViewController.hideKenny), userInfo: nil, repeats:
true)
    }) // NB using handler this time

    alert.addAction(replay)
    // handle replay interaction with 'replay' action handler code
    // for 'replay' button next to 'ok' button

    self.present(alert, animated: true, completion: nil)
}
}

```

7-59. 'whack-a-mole' style game
use UserDefaults to persist high score

in ViewDidLoad

```

{.....
// check for any high score set already
let highscore = UserDefaults.standard.object(forKey: "highscore")

if let newScore = highscore as? Int
{
    highscoreValueLabel.text = String(newScore)
}

```

and in countDown

```

if counter == 0
{
    timer.invalidate()
    hideTimer.invalidate()

    // checking high scores here

    if self.score > Int(highscoreValueLabel.text!)! // NB '!'
    {
        UserDefaults.standard.set(self.score, forKey: "highscore")
        highscoreValueLabel.text = String(self.score)
    }
}

```

8-61 -> 8-63 git & git hub

9-65 TableView into storyboard/ViewController
(i) stretch to whole corners of view and (ii) (Add new constraints) set all margin constraints to 0

9-66

```
class ViewController: UIViewController, UITableViewDelegate,  
UITableViewDataSource {  
    // NB manually subclass/inherit ViewController also from  
    UITableViewDelegate, UITableViewDataSource  
  
    override func viewDidLoad() {  
        super.viewDidLoad()  
        // Do any additional setup after loading the view, typically from a nib.  
  
        tableView.dataSource = self  
        tableView.delegate = self  
        // self so as to have more options when subclassing tableview functions -  
        eg numberOfRowsInSection and cellForRowAt  
  
    }  
}
```

```
// delegate functions  
func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {  
    return 10  
}
```

```
func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {  
    let cell = UITableViewCell() // instantiate a cell  
    cell.textLabel?.text = "Checking cell code"  
    return cell  
}
```

9-67

```
var landmarkNamesArray = [String]()  
var landmarkImagesArray = [UIImage]()  
  
landmarkNamesArray.append("Great Wall")  
landmarkImagesArray.append(UIImage(named:"great-wall.jpeg")!)  
&c &c  
  
func tableView(_ tableView: UITableView, numberOfRowsInSection section:
```

```

Int) -> Int {
    return landmarkNamesArray.count
    // tell table to have e.g. 6 rows for 6 images and names
}

func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {
    let cell = UITableViewCell() // instantiate a cell
    // cell.textLabel?.text = "Checking cell code" // check rendering ok

    cell.textLabel?.text = landmarkNamesArray[indexPath.row]
    // get appropriate content for *this* cell https://stackoverflow.com/a/50107220
    return cell // this cell for table row(s)
}

```

9-68 delete cell/row

NB on iphone, swipe left to reveal 'delete' button

```

// add another delegate, for deletion
func tableView(_ tableView: UITableView, commit editingStyle: UITableViewCellStyle, forRowAt indexPath: IndexPath) {

    if editingStyle == .delete
    {
        landmarkNamesArray.remove(at: indexPath.row)
        landmarkImagesArray.remove(at: indexPath.row)
        tableView.deleteRows(at: [indexPath], with:
            UITableViewRowAnimation.fade)
    }
}

```

9-69 ViewController for images (on swipe right)

as per 6-41 & 6-42 with segues,

steps 1-4

(1) add ViewController and (2) add a new Cocoa Touch Class file (3) (subclass UIViewController as default) of class named e.g. say ImageViewController (4) on storyboard, select new ViewController, in Identity Inspector , set 'Class' to new .swift filename from 3 eg ImageViewController (view now connected to code)

add & connect imageView and label

select (1st) ViewController, go to Edit/Embed In.../Navigation Controller (also tab controller available) - navigation pane will provide 'back' button between 2nd and 1st ViewControllers

if 1st-to-2nd segue popup choice is 'Show', navigation pane will be applied to 2nd ViewController (ImageViewController) also; if 'Present Modally' selected, the 1st ViewController's navigation pane won't be applied to 2nd ViewController.

(1) Highlight left-most button 'ViewController' at top of View Controller; (2) Ctrl+drag to 2nd View Controller to segue (3) remember to name segue in Attribute Inspector (utilities, top-right) eg 'tolmageVCSegue' (be ready to copy this string into *performSegue* and into *prepareForSegue*(below))

go for show (add navigation pane to 2nd viewcontroller)
(seen at top of 2nd ViewController)

9-70 set up segues between selection in 1st ViewController and display in 2nd ViewController

```
var chosenLandmarkName = ""  
var chosenLandmarkImage = UIImage()  
  
// 2 variables corresponding with similar variables in other ViewController
```

NB typing in part of many-fold function templates eg tableView eg didSelectRow in XCode will bring up desired parameter combination as auto-complete suggestion (easier than finding from typing in tableView(... and wading through all possible suggested options)

```
// add yet another delegate, to pass over data of selection in segue to other  
ViewController, if row selected by user  
func tableView(_ tableView: UITableView, didSelectRowAt indexPath:  
IndexPath) {  
    self.chosenLandmarkName = landmarkNamesArray[indexPath.row] // get  
selected row data, ready to pass  
    self.chosenLandmarkImage = landmarkImagesArray[indexPath.row] // get  
selected row data, ready to pass  
    performSegue(withIdentifier: "tolmageVCSegue", sender: nil)  
}
```

10-73

2 view controllers with a navigation pane embedded-in, tableview in 2nd viewcontroller - named DetailsViewController
segue and new file
as per 9-69 but also

(1) rename ViewController.swift to 'tableView.swift' (persistent click & hold?)&
(2) change ViewController class name in new detailsVC.swift to
tableViewController and (3) in Identity Inspector, change 'Class' from generic
'ViewController' to renamed 'tableViewController'

NB save, close project & reopen project if 'cannot find information about
suchaclass'

NB git commit —amend
to add 1 more file to commit
(or git command —amend -m "xyzzzz"
for message)

https://medium.com/@igor_marques/git-basics-adding-more-changes-to-your-last-commit-1629344cb9a8

NB vi default git editor <https://www.tutorialspoint.com/unix/unix-vi-editor.htm>
ughhh

:q to quit vi
Ctrl+x Ctrl+c to quit emacs

<https://help.github.com/en/articles/associating-text-editors-with-git>

atom was installed, as it happened, so use atom:
git config —global core.editor "atom —wait"

PNJ-dev-Mac-09:Simpsons book peterjenkin\$ git tag -a v0.5 -m "my version
0.5 just starting"

git push
git push —tags
git push —follow-tags
git push && git push —tags
git config --global push.followTags true

<https://stackoverflow.com/a/5195913>

<https://stackoverflow.com/questions/3745135/push-git-commits-tags-simultaneously/3745250>

PNJ-dev-Mac-09:Simpsons book peterjenkin\$ **git tag**
v0.5

```
PNJ-dev-Mac-09:Simpsons book peterjenkin$ git push origin v0.5
Counting objects: 1, done.
Writing objects: 100% (1/1), 177 bytes | 177.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0)
To github.com:pjenkin/iOSSwiftTutorial10SimpsonsBookTableClass.git
 * [new tag]      v0.5 -> v0.5
```

<https://github.com/pjenkin/iOSSwiftTutorial10SimpsonsBookTableClass/tags>

- 1
- Star
0
- Fork
0

pjenkin/iOSSwiftTutorial10SimpsonsBookTableClass

Code
Issues 0
Pull requests 0
Projects 0
Wiki
Insights
Settings
Releases
Tags
Tags
v0.5 ...

- 8 minutes ago c31756d zip tar.gz

-

10-74 drag in images
10-75 new class - Simpson
add new file - swift file - Simpson.swift

```
//  
// Simpson.swift  
// Simpsons book  
//
```

```
// Created by Peter Jenkin on 06/04/2019.  
// Copyright © 2019 Peter Jenkin. All rights reserved.  
//
```

```
import Foundation  
import UIKit // for UIImage
```

```
/// class to represent a Simpson character
```

```
class Simpson  
{  
    var name = ""  
    var occupation = ""  
    var image = UIImage()
```

```
    init() // constructor aka init in Swift  
    {  
        name = "default Simpsons name"  
    }  
}
```

10-76 as per 9-66 & 9-67, set up TableView delegates and datasources (extra subclassing &c)

```
class tableViewController: UIViewController, UITableViewDelegate,  
UITableViewDataSource {
```

```
to match (in ViewDidLoad)
```

```
    // table view setup  
    tableView.delegate = self  
    tableView.dataSource = self
```

```
// adding delegates to display rows and their actions
```

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

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

```

let cell = UITableViewCell()
cell.textLabel?.text = simpsons[indexPath.row].name
return cell
}

```

10-77

NB deleting erroneously connected/named label (age instead of occupation):
 select label, go to Connections Inspector, delete entry in Referencing Outlets
 (ageLabel)-(x DetailsViewController) by clicking 'x'
 Label will no longer be named other than 'Label' in assets
 Variable (no longer connected) can be deleted from code
 (check ok, by cmd+B to build)

// in DetailsViewController (2nd)

```

selectedSimpson = Simpson()

override func viewDidLoad() {
    super.viewDidLoad()

    nameLabel.text = selectedSimpson.name
    occupationLabel.text = selectedSimpson.occupation
    imageView.image = selectedSimpson.image
}

```

// in tableViewController (1st)

```

var simpsons = [Simpson]()      // array of Simpson objects
var chosenSimpson = Simpson()

// add another delegate for if row chosen (to segue)
func tableView(_ tableView: UITableView, didSelectRowAt indexPath: IndexPath) {
    self.chosenSimpson = simpsons[indexPath.row] // pass over Simpsons
    object according to row picked by user
    // performSegue(withIdentifier: <#T##String#>, sender: <#T##Any?#>)
    self.performSegue(withIdentifier: "toDetailsViewController", sender:
    nil) // segue name string copy/pasted carefully from Attributes Inspector
    off of main storyboard
    // don't bother with sender info - only 1 route
}

// define bespoke prepare for segue function for this app (override)
override func prepare(for segue: UIStoryboardSegue, sender: Any?) {

```

```

if segue.identifier == "toDetailsViewController"
{
    let destinationVC = segue.destination as! detailsVC
    destinationVC.selectedSimpson = self.chosenSimpson
}

}

```

11-80 Core Data
tick 'Use Core Data' in XCode new project

In place of UserDefaults (key/value pairs), Core Data will persist lots of data - serialised via interface into XML, binary or SQLLite

<https://developer.apple.com/library/archive/documentation/Cocoa/Conceptual/CoreData/index.html>
<https://www.raywenderlich.com/7569-getting-started-with-core-data-tutorial>
<https://medium.com/xcblog/core-data-with-swift-4-for-beginners-1fc067cca707>

Core data model in file e.g. Artbook.xcdatamodeld

Entities / Fetch Requests / Configurations

'Add Entity' button at bottom left

Entities should be named (in plural, e.g. 'Paintings' as a db entity containing several records) can have Attributes, like fields or properties

eg Paintings; artist (string) name (string) year(integer), image (Binary Data) ...

11-81

drag TableView and Embed into... a navigation pane
 – drag Bar Button into navigation pane (right hand end)

Select Bar Button - in Attribute Inspector in 'System Item' list, change from 'Custom' to 'Add' (changes to + symbol)

connect TableView into code as Collect

connect Bar Button into code as Action - addButtonClicked

drag a 2nd ViewController into Main.storyboard, and
 Ctrl-drag from icon to make a segue - in Attributes Inspector, change Identifier 'toCreateVC' (needed for *performSegue* and similar methods)

File/new/Cocoa Touch Class - 'createVC.swift', then

select 2nd VC and in Identity Inspector, change Class to newly appeared 'createVC' at end of list.

Drag onto 2nd VC (createVC) 3 Text Fields (Enter Painting Name, Enter Artist, Enter Year of the Painting) and a button (Save)

constraints - to centre. (NB alter constraint to match canvas where stretched to suitable length bigger than text currently in text field)

use < and > in Assistant editor window bar to see previous and next file
select wanted file from path in bar and use manual instead of automatic to help select if needed

11-82

tableView set up with delegates, subclassing as per 9-66 & 10-76

11-83

make up 'tap here' png, simple 200x100 - drag into project
in ImageView Attributes Inspector - 'Image' to that image
then

imageView.isUserInteractionEnabled = true
and add gesture recogniser as per 6-49

```
override func viewDidLoad() {  
    super.viewDidLoad()  
  
    imageView.isUserInteractionEnabled = true  
    // (3)  
    // as per 6-49 setup & add gesture recogniser for clicking this image  
  
    let gestureRecognizer = UITapGestureRecognizer(target: self, action:  
#selector(createVC.selectImage))  
    // (1)  
    imageView.addGestureRecognizer(gestureRecognizer)  
    // (2)  
    // NB not imageView.addGestureRecognizer(gestureRecognizer) sans Tap!  
    // Do any additional setup after loading the view.  
}  
// (4)  
func selectImage()  
{  
    // selecting image from library  
    let picker = UIImagePickerController()
```

```

    picker.delegate = self // required ViewController to subclass
    UIImagePickerControllerDelegate, UINavigationControllerDelegate
    picker.sourceType = .photoLibrary
    // source could have been from camera; library for demonstration
    picker.allowsEditing = true
    //present(<#T##viewControllerToPresent:
    UIViewController##UIViewController#, animated: <#T##Bool#>, completion:
    <#T##(() -> Void)?##(() -> Void)?##() -> Void#>)
        present(picker, animated: true, completion: nil)
        // show ImagePicker, animated, no handler function on completion
    }

// type in didfini ... & auto-complete d'pick right code

func imagePickerController(_ picker: UIImagePickerController,
didFinishPickingMediaWithInfo info: [String : Any]) {
    imageView.image = info[UIImagePickerControllerEditedImage] as?
    UIImage
    // colossal flag name - typed in nearly all before auto-complete
    // then UIImage auto-completed to huge name!
    // (happening on video too !)
    // get (hopefully) Image from picker cast as UIImage
    //self.dismiss(animated: <#T##Bool#>, completion: <#T##(() -> Void)?
    ##(() -> Void)?##(() -> Void#>)

    self.dismiss(animated: true, completion: nil)

    // info is a dictionary
}

```

then in Info.plist - like manifest? - authorisation ? - new role, so Add Row before first > at bottom ('+' button): list starting 'Application Category' - from long list of privileges, select 'Privacy - Photo Library Usage Description' giving app permission to use the Photo Library as done in the code - in Value, type in a reason for usage eg 'So that we can use a photo to show as art'

11-84 Core Data

Shift-Cmd-H on simulator to get start menu (**Home** - see simulator menus)

press & hold to save image/copy

AppDelegate.swift - code generated
eg NB lazy var persistentContainer: NSPersistentContainer = {

```

/*
The persistent container for the application. This implementation
creates and returns a container, having loaded the store for the
application to it. This property is optional since there are legitimate
error conditions that could cause the creation of the store to fail.
*/



@IBAction func saveButtonClicked(_ sender: Any) {
    let appDelegate = UIApplication.shared.delegate as! AppDelegate // NB
AppDelegate.swift
    let context = appDelegate.persistentContainer.viewContext // use this
context (in this app) for saving data into model

    let newArt = NSEntityDescription.insertNewObject(forEntityName:
"Paintings", into: context)
    // use new entity for CoreData defined in Artbook.xcdatamodeld
    // need to import CoreData

    newArt.setValue(nameText.text, forKey: "name") // use attributes of above
entity, as just now defined in Artbook.xcdatamodeld, coming from control in
ViewController
    // NB for all TextField attributes 'Correction' and 'Check Spelling', set to
'No' https://stackoverflow.com/a/42752380
    newArt.setValue(artistText.text, forKey: "artist")
    //newArt.setValue(artistText.text, forKey: "style") // not using 'style' field/
attribute at mo

    if let year = Int(yearText.text) // checking of text cast-able as integer
    {
        newArt.setValue(year, forKey: "year")
    }

    //let data = UIImageJPEGRepresentation(<#T##image:
UIImage##UIImage#, <#T##compressionQuality: CGFloat##CGFloat#>)
    let data = UIImageJPEGRepresentation(imageView.image!, 0.5)
    newArt.setValue(data, forKey: "image") // set binary data for 'image'
attribute/field

    // now save values
    do
    {
        try context.save()
        print("Saved successfully")
    }
    catch
    {

```

```

        print("Error trying to save")
    }

}

```

// **NB** for all TextField attributes 'Correction' and 'Check Spelling', set to '**No**'
<https://stackoverflow.com/a/42752380>
from 'Default' using Attribute Inspector else error: *Unable to copy asset information from https://mesu.apple.com/assets/ for asset type*

11-85 CoreData retrieved

```

// set up arrays in class:
var nameArray = [String]()
var yearArray = [Int]()
var artistArray = [String]()
var imageArray = [UIImage]()

// call retrieveInfo in viewDidLoad:
@IBOutlet weak var tableView: UITableView!
override func viewDidLoad() {
    super.viewDidLoad()
    // Do any additional setup after loading the view, typically from a nib.

    tableView.dataSource = self
    tableView.delegate = self

// function for retrieving CoreData
    retrieveInfo()    // populate table with persisted CoreData results
}

/// function for retrieving Core Data records
func retrieveInfo()
{
    self.nameArray.removeAll()           // clear out previous before fetching
new
    self.yearArray.removeAll()
    self.artistArray.removeAll()
    self.imageArray.removeAll()

let appDelegate = UIApplication.shared.delegate as! AppDelegate

```

```

let context = appDelegate.persistentContainer.viewContext

let fetchRequest =
NSFetchRequest<NSFetchRequestResult>(entityName: "Paintings") // need to import CoreData
    // NB template of NSFetchedResultsController, to pass to context to receive CoreData results

fetchRequest.returnsObjectsAsFaults = false // have to write this

do
{
    let results = try context.fetch(fetchRequest)
    if results.count > 0 // NB results in an array
    {
        for result in results as! [NSManagedObject] // cast as NSManagedObject, to use .value(forKey...
        {
            if let name = result.value(forKey: "name") as? String
            {
                self.nameArray.append(name)
            }
            if let year = result.value(forKey: "year") as? Int
            {
                self.yearArray.append(year)
            }
            if let artist = result.value(forKey: "artist") as? String
            {
                self.artistArray.append(artist)
            }
            if let imageData = result.value(forKey: "image") as? Data
            {
                let image = UIImage(data: imageData)
                self.imageArray.append(image!) // '!' else error/warning - image should exist cos of if check above
            }
            self.tableView.reloadData() // refresh the table view - delegate cellForRowAt should display new data
        }
    }
}
catch
{

```

```

        print("There was an error while retrieving")
    }
}

```

11-86 prepare for segue to pass chosen variable back to first ViewController
add delegates &c and set up
selected/chosen variable correspondence as in 10-72, 10-76, 10-77

11-87 predicates for filters in CoreData model queries

in createVC

```

override func viewDidLoad() {
    super.viewDidLoad()

    if chosenPainting != ""
    {
        print(chosenPainting) // diagnostic
        // prep context for another CoreData query
        let appDelegate = UIApplication.shared.delegate as! AppDelegate
        let context = appDelegate.persistentContainer.viewContext

        let fetchRequest =
        NSFetchRequest<NSFetchRequestResult>(entityName: "Paintings")
            // fetchRequest.predicate = NSPredicate(format: <#T##String#>,
            <#T##args: CVarArg...##CVarArg#>)
            fetchRequest.predicate = NSPredicate(format: "name = %@", self.chosenPainting) // NB predicate for CoreData query
        fetchRequest.returnsObjectsAsFaults = false

        do
        {
            // could refactor this into a function
            let results = try context.fetch(fetchRequest)
            if results.count > 0
            {
                for result in results as! [NSManagedObject]
                {
                    if let name = result.value(forKey: "name") as? String
                    {
                        nameText.text = name // populate textField with retrieved value for name (according to key/value query from other ViewController)
                    }
                }
            }
        }
    }
}

```

```

        if let year = result.value(forKey: "year") as? Int
        {
            yearText.text = String(year) // populate textField with (cast)
retrieved value for name (according to key/value query from other
ViewController)
        }
        if let artist = result.value(forKey: "artist") as? String
        {
            artistText.text = artist // populate textField with retrieved
value for name (according to key/value query from other ViewController)
        }
        if let imageData = result.value(forKey: "image") as? Data
        {
            let image = UIImage(data: imageData)
            imageView.image = image // populate textField with
retrieved value for name (according to key/value query from other
ViewController)
            // do we need self.imageView.image as in video here??
        }

    }
}
catch
{
    print("Error trying to fetch using predicate")
}
}

imageView.isUserInteractionEnabled = true

// as per 6-49 setup & add gesture recogniser for clicking this image

let gestureRecognizer = UITapGestureRecognizer(target: self, action:
#selector(createVC.selectImage))

imageView.addGestureRecognizer(gestureRecognizer)

// Do any additional setup after loading the view.
}

```

11-88 **Notification Center** used to coordinate view actions (using **viewWillLoad**) according to user actions

```

override func viewWillAppear(_ animated: Bool) {
    // look out for "paintingRecorded" from saveButtonClicked in other
    ViewController
    // NotificationCenter.default.addObserver(<#T##observer:
    Any##Any#>, selector: <#T##Selector#>, name: <#T##NSNotification.Name?
    #>, object: <#T##Any?#>)
    // use polymorph with selector, so as to connect this code to another
    (handler-seque) function: the bespoke retrieveInfo function
    NotificationCenter.default.addObserver(self, selector:
    #selector(ViewController.retrieveInfo), name:
    NSNotification.Name(rawValue:"paintingRecorded"), object: nil)

    // NB viewWillAppear called *every* time view shown; viewDidLoad only
    once, on first showing of view
}

```

NB // NB viewWillAppear called *every* time view shown; viewDidLoad only once, on first showing of view

constraints - compression resistance priority (1000 for required). <https://stackoverflow.com/a/42324608>
NB dashed line in 'add constraint' dialogue click line to add constraint, viz separation

NB **git log** to see commits & tags

12-91 Map Kit View into view & code

mapView.delegate = self, & subclass as MKMapViewDelegate

12-92 subclass CLLocationManagerDelegate &

```

override func viewDidLoad() {
    super.viewDidLoad()
    // Do any additional setup after loading the view, typically from a nib.

    mapView.delegate = self
    locationManager.delegate = self
    locationManager.desiredAccuracy = kCLLocationAccuracyBest // NB
    battery drain with accuracy
    locationManager.requestWhenInUseAuthorization() // how often to ask for
    permission (cf adding Privacy-Location when in use' in Info.plist - )

```

```

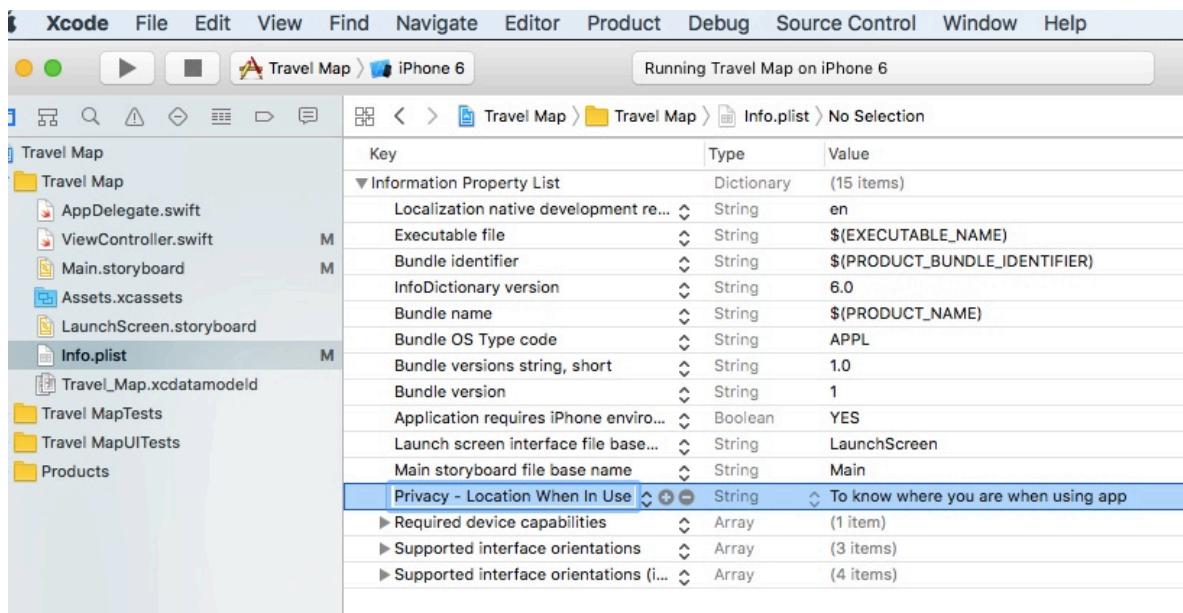
        locationManager.startUpdatingLocation() // NB function
didUpdateLocation function
    }

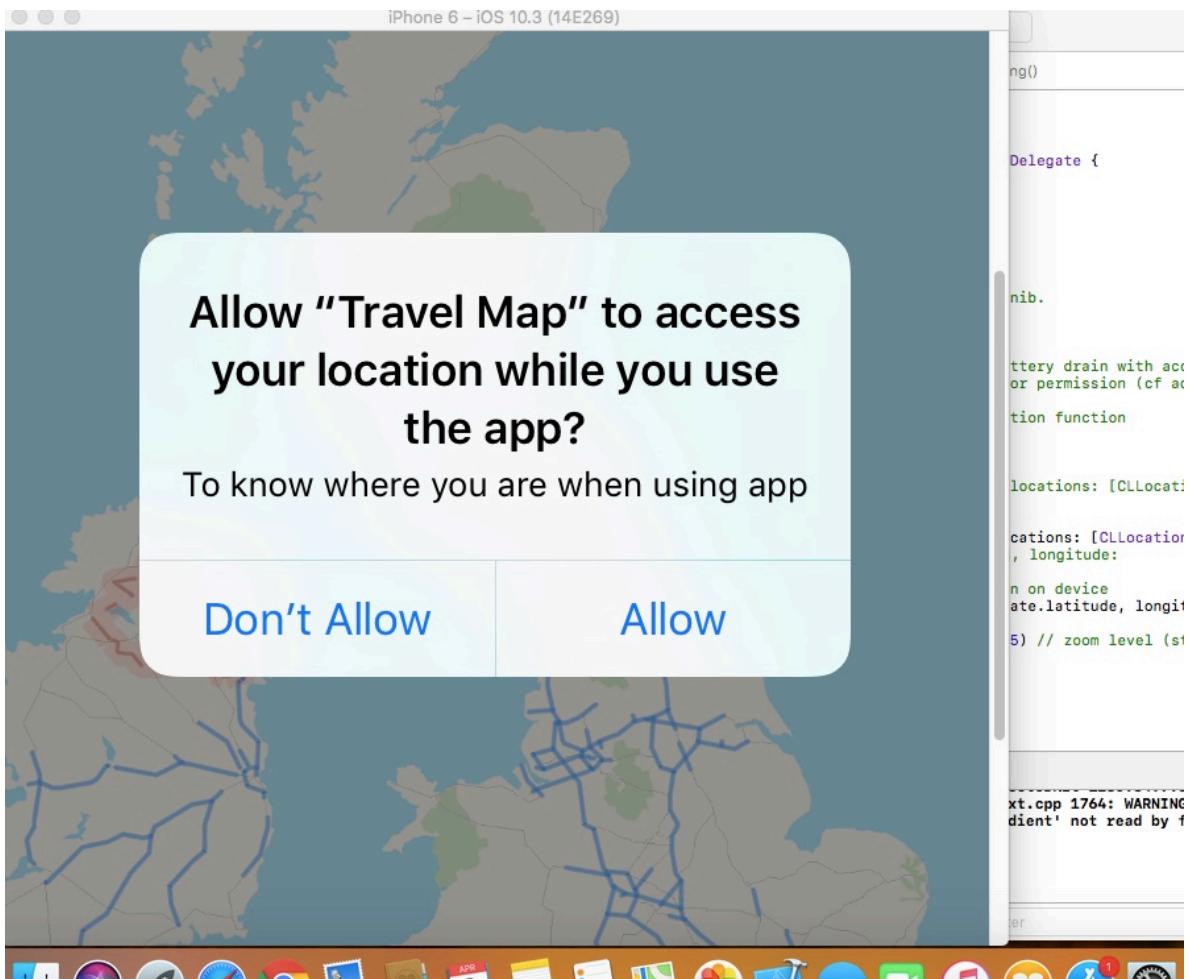
// function to handle location update NB parameters
//func locationManager(_ manager: CLLocationManager,
didUpdateLocations locations: [CLLocation]) {
    // code
//}

func locationManager(_ manager: CLLocationManager, didUpdateLocations
locations: [CLLocation]) {
    //let location = CLLocationCoordinate2D(latitude:
<#T##CLLocationDegrees#>, longitude: <#T##CLLocationDegrees#>
    // NB locations is a parameter in this 'handler' for updated location on
device
    let location = CLLocationCoordinate2D(latitude:
locations[0].coordinate.latitude, longitude: locations[0].coordinate.longitude)
    let span = MKCoordinateSpan(latitudeDelta: 0.05, longitudeDelta: 0.05) //
zoom level (standard 0.05)
    let region = MKCoordinateRegion(center: location, span: span)
    self.mapView.setRegion(region, animated: true)
}

```

'Privacy-Location when in use' in Info.plist





Simulator - Debug/location/ Custom location - by default, Apple office at Cupertino (old office?)

12-93 Import CoreLocation library

Also need to select project (eg 'Travel Map'), and from tabs '**Build Phases**' - '**Link Binary with Libraries**' - '+' button, in list find 'CoreLocation.framework'. 'Add' this (framework/library).

(these not added by default to save space) - also import CoreLocation

12-94 Annotations

Annotations are tags within maps; a different gesture recogniser used (long press)

in viewDidLoad

```
locationManager.startUpdatingLocation() // NB function  
didUpdateLocation function
```

```
// declare gesture recognition  
let recognizer = UILongPressGestureRecognizer(target: self, action:  
#selector(ViewController.chooseLocation(gestureRecognizer:)))
```

```
recognizer.minimumPressDuration = 3 // 3 seconds press  
mapView.addGestureRecognizer(recognizer)
```

and

```
func chooseLocation(gestureRecognizer:  
UILongPressGestureRecognizer) // bespoke handler function  
{  
    if gestureRecognizer.state == UIGestureRecognizerState.began  
    {  
        let touchedPoint = gestureRecognizer.location(in: self.mapView)  
        // let chosenCoordinates = self.mapView.convert(<#T##point:  
CGPoint##CGPoint#>, toCoordinateFrom: <#T##UIView?#>)  
        let chosenCoordinates = self.mapView.convert(touchedPoint,  
toCoordinateFrom: self.mapView)  
        let annotation = MKPointAnnotation()  
        annotation.coordinate = chosenCoordinates  
        annotation.title = "A new annotation"  
        annotation.subtitle = "This is the point chosen"  
        self.mapView.addAnnotation(annotation)  
    }  
}
```

12-95 title and subtitle of annotation from textfields on ViewController

12-96 CoreData to save the location

12-96 Saving location information to Coredata context

new Entity in TravelMap.xcdatamodelid

as per 11-84

12-97

new ViewController, as firstViewController

(Cocoa Touch .swift & in Main.storyboard)

-> arrow is entry point into application (draggable)

drag firstViewController to left of other VC

segue with identifier "toMapVC"

drag TableView to firstVC and BarButton (System Item 'Add') to firstVC nav pane - drag into code

then (as ever) subclass and

```
tableView.delegate = self  
tableView.dataSource = self  
}
```

```

// delegates added for tableView
func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {
    return 5      // 5 rows, say
}

func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {
    let cell = UITableViewCell()
    cell.textLabel?.text = "checking text"
    return cell
}

@IBAction func addBtnClicked(_ sender: Any) {
    performSegue(withIdentifier: "toMapVC", sender: nil)
}

```

12-98 Retrieve Coredata of location info (as per 11-85)

```

func fetchData()
{
    let appDelegate = UIApplication.shared.delegate as! AppDelegate
    let context = appDelegate.persistentContainer.viewContext
    // NB autocomplete trying to write AppDelegate not appDelegate

    let request = NSFetchedResultsController<NSFetchRequestResult>(entityName: "Locations")
    // cf Entity entry in Travel_Map.xcdatamodeld
    // NB add <NSFetchRequestResult> template specifier manually - ensure
    // autocomplete doesn't pick wrong one!
    request.fetchBatchSize = 20
    request.returnsObjectsAsFaults = false
    do
    {
        let results = try context.fetch(request)
        if results.count > 0
        {
            // clear all arrays
            self.titleArray.removeAll(keepingCapacity: false)
            self.subtitleArray.removeAll(keepingCapacity: false)
            self.latitudeArray.removeAll(keepingCapacity: false)
            self.longitudeArray.removeAll(keepingCapacity: false)

            for result in results as! [NSManagedObject]

```

```
        if let title = result.value(forKey: "title") as? String
        {
            self.titleArray.append(title)
        }
        if let subtitle = result.value(forKey: "subtitle") as? String
        {
            self.subtitleArray.append(subtitle)
        }
        if let latitude = result.value(forKey: "latitude") as? Double
        {
            self.latitudeArray.append(latitude)
        }
        if let longitude = result.value(forKey: "longitude") as? Double
        {
            self.longitudeArray.append(longitude)
        }
        print("got here")
        self.tableView.reloadData() // refresh table view with retrieved
results
    }
}
catch
{
}

// delegates added for tableView
func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {
    // return 5      // 5 rows, say
    return titleArray.count // however many as are in the titleArray
}

func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {
    let cell = UITableViewCell()
    //cell.textLabel?.text = "checking text"
    cell.textLabel?.text = titleArray[indexPath.row]
    return cell
}
```

```
// and call fetchData from viewDidLoad
```

12-99 pass data between ViewControllers

corresponding sets of array variables in ViewControllers

in firstViewCOnroller:

```
// additional delegate for when a tableView row (ie location title) selected
func tableView(_ tableView: UITableView, didSelectRowAt indexPath: IndexPath) {
```

```
    chosenTitle = titleArray[indexPath.row] // pass over data to other
    ViewController (ie the map view)
```

```
    chosenSubtitle = subtitleArray[indexPath.row]
```

```
    chosenLatitude = Double(latitudeArray[indexPath.row])
```

```
    chosenLongitude = Double(longitudeArray[indexPath.row])
```

```
}
```

```
override func prepare(for segue: UIStoryboardSegue, sender: Any?) {
```

```
    if segue.identifier == "toMapVC"
```

```
{
```

```
    let destinationVC = segue.destination as! ViewController
```

```
    // pass over/transmit data to the other ViewController
```

```
    destinationVC.transmittedTitle = self.chosenTitle
```

```
    destinationVC.transmittedSubtitle = self.chosenSubtitle
```

```
    destinationVC.transmittedLatitude = self.chosenLatitude
```

```
    destinationVC.transmittedLongitude = self.chosenLongitude
```

```
}
```

```
}
```

12-101

```
// customising annotation, to show new kind of non-standard button
bespoke for this app
```

```
func mapView(_ mapView: MKMapView, viewFor annotation: MKAnnotation)
-> MKAnnotationView? {
```

```
if annotation is MKUserLocation {
```

```
    return nil
}
```

```
let reuseId = "myAnnotation"
```

```

var pinView = mapView.dequeueReusableCell(withIdentifier:
reuseld) as? MKPinAnnotationView
// reuse just the 1 annotation view (good for memory resources)

if pinView == nil
{
    pinView = MKPinAnnotationView(annotation: annotation, reuseIdentifier:
reuseld)
    pinView?.canShowCallout = true
    pinView?.pinTintColor = UIColor.red


    let button = UIButton(type: .detailDisclosure) // .detailDisclosure ==
UIButton.detailDisclosure
    pinView?.rightCalloutAccessoryView = button
}
else // if pinView already existent (already defined, and reused)
{
    pinView?.annotation = annotation
}
return pinView
}

```

12-102 handler function for mapView Map View so that, if the map is not being used to display current location (ie is used in browsing other locations instead) then alter to show a bespoke annotation which, when clicked, will show driving directions

```

// 1 of 2 - customising annotation, to show new kind of non-standard button
bespoke for this app
func mapView(_ mapView: MKMapView, viewFor annotation: MKAnnotation)
-> MKAnnotationView? {
    //

    if annotation is MKUserLocation {
        return nil
    } // don't do anything if the map is opening for the purpose only of
showing the current user location (because directions to current location would
be pointless!)

    let reuseld = "myAnnotation"

    var pinView = mapView.dequeueReusableCell(withIdentifier:
reuseld) as? MKPinAnnotationView
// reuse just the 1 annotation view (good for memory resources)

```

```

if pinView == nil
{
    pinView = MKPinAnnotationView(annotation: annotation, reuseIdentifier:
reuseId)
    pinView?.canShowCallout = true
    pinView?.pinTintColor = UIColor.cyan


        let button = UIButton(type: .detailDisclosure) // .detailDisclosure ==
UIButton.detailDisclosure - detailDisclosure is 'i' in circle
        pinView?.rightCalloutAccessoryView = button // on right hand side
(could've been on left)
    }
else // if pinView already existent (already defined, and reused)
{
    pinView?.annotation = annotation
}
return pinView
}

// 2 of 2 - for custom annotation, to pull up driving directions from current
location to that location
//func mapView(_ mapView: MKMapView, annotationView view:
MKAnnotationView, calloutAccessoryControlTapped control: UIControl) {
//<#code#>
//} // had to auto-complete by calloutAccess.....
func mapView(_ mapView: MKMapView, annotationView view:
MKAnnotationView, calloutAccessoryControlTapped control: UIControl) {
    if transmittedLatitude != 0 && transmittedLongitude != 0
    {
        self.requestCLLocation = CLLocation(latitude: transmittedLatitude,
longitude: transmittedLongitude) // to use in reverse geolocation below
    }

    //CLGeocoder().reverseGeocodeLocation(<#T##location:
CLLocation##CLLocation#, completionHandler:
<#T##CLGeocodeCompletionHandler##CLGeocodeCompletionHandler##([CL
Placemark]?, Error?) -> Void#>)
    CLGeocoder().reverseGeocodeLocation(requestCLLocation)
    {
        (placemarks, error) in
        if let placemark = placemarks
        {
            if placemark.count > 0 // if successfully reverse geo-coded
            {

```

```

        let newPlaceMark = MKPlacemark(placemark: placemark[0])
        let item = MKMapItem(placemark: newPlaceMark)
        item.name = self.transmittedTitle

        // launch a screen showing travel directions to/from
        let launchOptions = [MKLaunchOptionsDirectionsModeKey :
            MKLaunchOptionsDirectionsModeDriving]
        item.openInMaps(launchOptions: launchOptions)
    }
}
}
}

```

13-105 Touch ID

```

import LocalAuthentication
authContext = LAContext()

```

var error: NSError?

```

if authContext.canEvaluatePolicy(.deviceOwnerAuthenticationWithBiometrics,
error: &error)

```

```

authContext.evaluatePolicy(.deviceOwnerAuthenticationWithBiometrics,localizedReason: "To check if tis you!", reply: (Bool, Error) -> Void)

```

TODO: & in Swift

<https://stackoverflow.com/a/24084786>

also <https://docs.swift.org/swift-book/LanguageGuide/AdvancedOperators.html>

can then use suitable device to check fingerprint

13-107 iMessage stickers

Sticker application in XCode

download some icons into project - as sticker pack

extension created, like app, so that icons can be selected and used in chat apps and so on

13-109 iMessage app

choose iMessage application from XCode

.swift file also having willBecome Active and didResignActive, didReceive,

didStartSending, didCancelSending, willTransition, didTransition

user can choose another application from within iMessage application

13-110 customised keyboard
usual XCode porjewctFile/New... Target
from list, Custom Keyboard extension

(cf android clamshell keyboard?)

KeyboardViewController.swift

functions eg nextKeyboardButton (go to next keyboard type)

populated viewDidLoad. setting up keyboard

didReceiveMemoryWarning, ...
(keyboard starting empty)

choose an app to run

Settings/General/Keyboards/Add new keyboard
1-111 png exportable into keyboard (add to target keyboard as well as project) as button
let myHeart = UIButton(type: UIButtonType.system)
myHeart.frame = CGRectMake(x: 50,y: 50, width: 50, height:50)

```
myHeart.setBackgroundImage(UIImage(named:"heart.png"), for:  
UIControlState.normal)  
myHeart.addTarget(self, action:  
#selector(keyboardViewController.myHeartTapped), for:  
UIControlEvents.touchUpInside)  
self.view.addSubview(myHeart)
```

```
func myHeartTapped()  
{  
    let textProxy = textDocumentProxy as UITextDocumentProxy  
    textProxy.insertText("some text as if typed")  
}
```

14-114

[fixer.io](#) API link for currency type
<http://api.fixer.io/latest/base=USD>

SeachBar dragged in 14-116

The screenshot shows the Xcode interface. The top menu bar includes Apple, Xcode, File, Edit, View, Find, Navigate, Editor, Product, Debug, Source Control, Window, and Help. Below the menu is a toolbar with standard icons. The main area displays the 'Info.plist — Edited' file for a project named 'Currency Converter'. The plist table shows various keys like 'Localization native development region', 'Executable file', and 'Bundle identifier'. To the right of the plist, a portion of a Swift file is visible, showing code related to a search bar and currency conversion.

```

    <Swift>
    import UIKit
    import Alamofire
    import SwiftyJSON

    class ViewController: UIViewController, UISearchBarDelegate {
        @IBOutlet weak var searchBar: UISearchBar!
        @IBOutlet weak var tableView: UITableView!

        override func viewDidLoad() {
            super.viewDidLoad()
            // Additional setup after loading the view.
            searchBar.delegate = self
        }

        func searchBarSearchButtonClicked(_ searchBar: UISearchBar) {
            print(searchBar.text!) // NB '!' unwrapping
            // clicked - pressing enter &c
            getCurrency(currency: searchBar.text!)
        }

        func getCurrency(currency: String) {
            let fixerAccessKey = "d1b90095d0b20832def727c348206e98"
            let url = URL(string: "http://api.fixer.io/latest?base=\(currency)?access_key=\(fixerAccessKey)") // deprecated
            let url = URL(string: "http://data.fixer.io/api/latest?base=\(currency)&access_key=\(fixerAccessKey)")
        }
    }
}

```

```

func searchBarSearchButtonClicked(_ searchBar: UISearchBar) {
    print(searchBar.text!) // NB '!' unwrapping
    // clicked - pressing enter &c
    getCurrency(currency: searchBar.text!)
}

// bespoke function to get the kind of currency used by user
func getCurrency(currency:String)
{
    // e.g. http://api.fixer.io/latest/base=USD

    // key d1b90095d0b20832def727c348206e98 - https://github.com/fixerAPI/fixer#readme
    // Request specific exchange rates by setting the symbols parameter.
    // GET /latest?symbols=USD,GBP

    let fixerAccessKey = "d1b90095d0b20832def727c348206e98"
    //let url = URL(string: "http://api.fixer.io/latest/base=\(currency)?access_key=\(fixerAccessKey)") // deprecated

    let url = URL(string: "http://data.fixer.io/api/latest?base=\(currency)&access_key=\(fixerAccessKey)")
}

```

```

print(url!)

// NB http not liked, https only allowed by default
// Add Row -
// 'App Transport security settings' - select and 'Add Row again' Allow
Arbitrary Loads to 'Yes' in Info.plist

let session = URLSession.shared

let task = session.dataTask(with: url!)
{(data, response, error) in

    if error != nil
    {
        let alert = UIAlertController(title:"Error", message:
error?.localizedDescription, preferredStyle: UIAlertControllerStyle.alert)
        let okButton = UIAlertAction(title: "OK", style:
UIAlertActionStyle.cancel, handler:nil)
        alert.addAction(okButton)
        self.present(alert, animated: true, completion: nil)
    }
    else // if all's ok, no error
    {

        do
        {
            let jSONResult = try JSONSerialization.jsonObject(with: data!, options: JSONSerialization.ReadingOptions.mutableContainers) as!
Dictionary<String, AnyObject>
                // from API docs, some fields arrays not strings, so use AnyObject -
into Dictionary key/value pairs
            if data != nil
            {
                DispatchQueue.main.async { // NB asynchronously
DispatchQueue.main.async
                    print(jSONResult)

                    print(jSONResult["rates"])

                let rates = jSONResult["rates"] as! [String: AnyObject]

                let usd = String(describing: rates["USD"]!) // from AnyObject
type - NB (i) describing (ii) ! to avoid 'Optional' printing out around string
                    self.usdLabel.text = "USD: \(usd)"
                    // NB also ?? coalescing/unpacking operator

```

```

        let gbp = String(describing: rates["GBP"]!) // from AnyObject
type
        self.ukLabel.text = "GBP: \(gbp)"

        let cad = String(describing: rates["CAD"]!) // from AnyObject
type
        self.canadaLabel.text = "CAD: \(cad)"

    }

}

} catch
{

}

}

}

task.resume() // start task actually
}

}

```

15-120 re Firebase <https://firebase.google.com/>
 15-121 re CocoaPods <https://cocoapods.org/>
 dependency/library manager cf npm

sudo gem install cocoapods

15-122 tabbed application

add another ViewController to left, drag arrow to make entry point, Ctrl-drag segue from 1st ViewController to tab bar controller

Add textfields and buttons and a label, and Ctrl+drag into (what will be signInVC's) code as outlets and actions.

File/New.. Cocoa Touch class of UIViewController (class 'signInVC'). In Identity Inspector for 1st View Controller, set Class to 'signInVC'.

Change filename and class name (code) of FirstViewController 1st tabbed VC to 'feedVC', and, using Identity Inspector, set its class to 'feedVC'.

(NB Outlets above class methods, actions below class methods, for neatness.)

Change filename and class name (code) of SecondViewController 1st tabbed VC to 'uploadVC', and, using Identity Inspector, set its class to 'uploadVC'.

Change tab label at bottom of (now) feedVC from 'First' to 'Feed' and change tab label at bottom of (now) uploadVC from 'Second' to 'Upload'.

Also change label in ViewController from FirstView to 'Feed' and change label in ViewController from FirstView to 'Upload'.

15-123 at Firebase <https://firebase.google.com/> go to console and '+ Add Project'

Add an app to get started: choose from iOS, Android or </> web code (cf Angular course notes/repo)

select iOS to start

Register app: Bundle identifier from Project view e.g. com.peternjenkin.Instagram-Firebase

after Register App, Download Config File is a **Google-Service-Info.plist** file
show in Finder and then drag/add into project

Add Firebase SDK

Instructions for CocoaPods

|

[Download ZIPC++](#)

Google services use [CocoaPods](#) to install and manage dependencies. Open a terminal window and navigate to the location of the Xcode project for your app (the one containing project folders).

Create a Podfile if you don't have one:

(i) pod init

content_copy

Copied

(ii) Open your Podfile (**using a text editor eg atom**) and add:
pod 'Firebase/Core'

content_copy

Copied

Includes Analytics by default

(iii) Save the file and run:
pod install

content_copy

Copied

This creates an .xcworkspace file for your app. Use this file for all future development on your application.

console for (i-iii):

```
PNJ-dev-Mac-09:Insta Clone Firebase peterjenkin$ ls
Insta Clone Firebase      Insta Clone FirebaseUITests
Insta Clone Firebase.xcodeproj Podfile
Insta Clone FirebaseTests

PNJ-dev-Mac-09:Insta Clone Firebase peterjenkin$ atom Podfile
PNJ-dev-Mac-09:Insta Clone Firebase peterjenkin$ pod install
Analyzing dependencies
Setting up CocoaPods master repo
$ /usr/bin/git clone https://github.com/CocoaPods/Specs.git master --
progress
Cloning into 'master'...
remote: Enumerating objects: 336, done.
remote: Counting objects: 100% (336/336), done.
remote: Compressing objects: 100% (248/248), done.
remote: Total 2965964 (delta 114), reused 188 (delta 66), pack-reused
2965628
Receiving objects: 100% (2965964/2965964), 630.59 MiB | 1.71 MiB/s, done.
Resolving deltas: 100% (1781448/1781448), done.
Checking out files: 100% (319288/319288), done.
```

CocoaPods 1.7.0.beta.3 is available.

To update use: `sudo gem install cocoapods --pre`
[!] This is a test version we'd love you to try.

For more information, see <https://blog.cocoapods.org> and the CHANGELOG for this version at <https://github.com/CocoaPods/CocoaPods/releases/tag/1.7.0.beta.3>

Setup completed

```
Downloading dependencies
Installing Firebase (5.20.1)
Installing FirebaseAnalytics (5.8.0)
Installing FirebaseCore (5.4.1)
Installing FirebaseInstanceID (3.8.1)
Installing GoogleAppMeasurement (5.8.0)
Installing GoogleUtilities (5.5.0)
Installing nanopb (0.3.901)
Generating Pods project
Integrating client project
```

[!] Please close any current Xcode sessions and use `Insta Clone Firebase.xcworkspace` for this project from now on.
Sending stats
Pod installation complete! There is 1 dependency from the Podfile and 7 total pods installed.

[!] Automatically assigning platform `ios` with version `10.3` on target `Insta Clone Firebase` because no platform was specified. Please specify a platform for this target in your Podfile. See `https://guides.cocoapods.org/syntax/podfile.html#platform`.

```
PNJ-dev-Mac-09:Insta Clone Firebase peterjenkin$ ls
Insta Clone Firebase           Insta Clone FirebaseUITests
Insta Clone Firebase.xcodeproj Podfile
Insta Clone Firebase.xcworkspace Podfile.lock
Insta Clone FirebaseTests      Pods
PNJ-dev-Mac-09:Insta Clone Firebase peterjenkin$
```

(4) Add initialisation code

To connect Firebase when your app starts up, add the initialisation code below (**2 lines in Swift**) to your main **AppDelegate** class (**AppDelegate.swift**).

```
import UIKit
import Firebase

@UIApplicationMain
class AppDelegate: UIResponder, UIApplicationDelegate {

    var window: UIWindow?

    func application(_ application: UIApplication,
                    didFinishLaunchingWithOptions launchOptions:
                    [UIApplicationLaunchOptionsKey: Any]?) -
        -> Bool {
        FirebaseApp.configure()
        return true
    }
}
```

NB <https://guides.cocoapods.org/using/using-cocoapods.html>
Should use xyzApp.xcworkspace to open XCode to work on app thereafter

NB error in AppDelegate.swift "**swift compiler error" could not build objective-c module 'firebase'**

<https://github.com/invertase/react-native-firebase/issues/1764#issuecomment-452178852>

```
add to  
post_install do |installer|  
  system("mkdir -p Pods/Headers/Public/FirebaseCore && cp Pods/  
FirebaseCore/Firebase/Core/Public/* Pods/Headers/Public/FirebaseCore/")  
end
```

ie copying everything in Pods/FirebaseCore/Firebase/Core/Public/ to a slightly different path Headers/Public/FirebaseCore/"

got rid of errors for the mo

may be a workaround but error reported on running on device, I get the Missing Module Error:
in native.js file inside initialiseNativeModule function

oh dear

<https://firebase.google.com/docs/ios/setup>

Add Firebase to your iOS Project

Prerequisites

Before you begin, you need a few things set up in your environment:

- Xcode 10.1 or later

Moving to Xcode 10: In September 2018, Apple announced that [starting March 2019, all iOS apps submitted to the App Store must be built with the iOS 12.1 SDK](#), which is included in Xcode 10.1. In March 2019, updates to the Firebase iOS SDKs will no longer support Xcode 9.

using 8.3 :(

cannot now continue building project

could go back one day and try this <https://stackoverflow.com/a/51289235>.
<https://github.com/firebase/firebase-ios-sdk/pull/1519> perhaps but outside chance

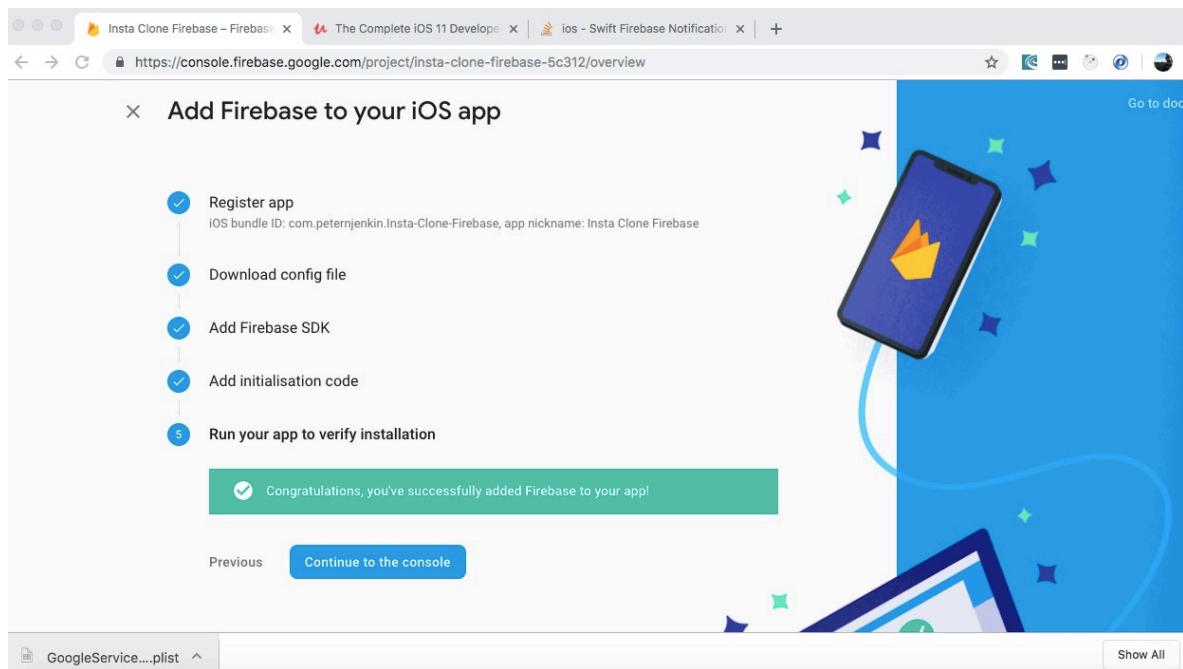
tried pulling from master of sdk repo (as per) but couldn't get pulling from master repo to work.

Downloaded/installed XCode 9.4.1 (the latest possible with 10.13.3 High Sierra)
- problem no longer there, basic program compiled & ran on simulator.

(next day) Firebase SDK registration final step 'Run your app to register with our servers': web page w
NB ensure GoogleService-info.plist in project, else Firebase.configure() will throw error

in XCode debug console while running

**2019-04-10 09:18:25.059840+0100 Insta Clone Firebase[1599:31743]
TIC Read Status [3:0x0]: 1:57
2019-04-10 09:18:25.629529+0100 Insta Clone Firebase[1599:31743]
TIC Read Status [4:0x0]: 1:57**



15-124. Firebase authentication:

into view import Firebase

Firebase-Authentication for users & logging-in various ways

Firebase-Database for records (flat, real-time)

Firebase-Storage for media files &c

NB In Authentication, set up sign-in method... **Email and password ENABLE**

separate pods for Firebase modules: Authentication, Database and Search

```
// import FirebaseAuth - has its own pod // - not done
// would need to add to Podfile:
/*
pod 'Firebase/Core'
pod 'Firebase/Auth'
pod 'Firebase/Database'
pod 'Firebase/Storage'
*/
// and 'pod install' again
```

```
PNJ-dev-Mac-09:Insta Clone Firebase peterjenkin$ pod install
Analyzing dependencies
Downloading dependencies
Installing Firebase 5.20.1
Using FirebaseAnalytics (5.8.0)
Installing FirebaseAuth (5.4.2)
Installing FirebaseAuthInterop (1.0.0)
Using FirebaseCore (5.4.1)
Installing FirebaseDatabase (5.1.1)
Using FirebaseInstanceId (3.8.1)
Installing FirebaseStorage (3.1.1)
Installing GTMSessionFetcher (1.2.1)
Using GoogleAppMeasurement (5.8.0)
Using GoogleUtilities (5.5.0)
Installing leveldb-library (1.20)
Using nanopb (0.3.901)
Generating Pods project
Integrating client project
Sending stats
Pod installation complete! There are 4 dependencies from the Podfile and 13
total pods installed.
```

Clean and build required, else error 'No such module FirebaseAuth' shown.

```
Auth.auth.signIn(withEmail: email.text! password.....)
```

```
@IBAction func signUpBtnClicked(_ sender: Any) {
    if emailText.text != "" && passwordText.text != ""
    {
```

```

Auth.auth().createUser(withEmail: emailText.text!, password:
passwordText.text!) { (user, error) in
    if error != nil
    { // if there's an error, tell user
        // copy/paste of below code, & alter to allow specific error
message!
        let alert = UIAlertController(title: "Error",
message:error?.localizedDescription, preferredStyle:
UIAlertControllerStyle.alert)

        let okButton = UIAlertAction(title: "OK", style:
UIAlertActionStyle.cancel, handler: nil)
        // NB alert is Controller, ok is Action

        alert.addAction(okButton)
        self.present(alert, animated: true, completion: nil)
        // set up a warning to user in case of error
    }
    else
    {
        print(user?.user.email) // NB extra .user https://stackoverflow.com/
a/50419010
        // print(user?.email)
        print("successful sign-up")
    }
} // completion:<#T##AuthDataResultCallback?
##AuthDataResultCallback?##(AuthDataResult?, Error?) -> Void#> NB
pressing enter can expand into block if appropriate, as above with (user, error)
in ....

} else
{
    let alert = UIAlertController(title: "Error", message: "Problem signing
up", preferredStyle: UIAlertControllerStyle.alert)

    let okButton = UIAlertAction(title: "OK", style: UIAlertActionStyle.cancel,
handler: nil)

    alert.addAction(okButton)
    self.present(alert, animated: true, completion: nil)
    // set up a warning to user in case of error
}
}

```

check Authentication in Firebase - a user should be found, on successful sign-up

```

15-125 Sign-In (via Firebase Auth)

@IBAction func signInBtnClicked(_ sender: Any) {
    //performSegue(withIdentifier: "toTabBar", sender: nil)

    if emailText.text != "" && passwordText.text != ""
    {
        Auth.auth().signIn(withEmail: emailText.text!, password:
passwordText.text!, completion: { (user, error) in
            if error != nil
            {
                // if there's an error, tell user
                // copy/paste of below code, & alter to allow sepcific error
message!
                let alert = UIAlertController(title: "Error",
message:error?.localizedDescription, preferredStyle:
UIAlertControllerStyle.alert)

                let okButton = UIAlertAction(title: "OK", style:
UIAlertActionStyle.cancel, handler: nil)
                // NB alert is Controller, ok is Action

                alert.addAction(okButton)
                self.present(alert, animated: true, completion: nil)
                // set up a warning to user in case of eror

            }
            else
            {
                // if no error, signed-in ok
                // go on to app
                print("successful sign in")
                self.performSegue(withIdentifier: "toTabBar", sender: nil)
                // NB *self*.performSegue needed here because in a closure code-
block
            }
        }
    }
    else
    {
        let alert = UIAlertController(title: "Error", message: "Problem signing
up", preferredStyle: UIAlertControllerStyle.alert)

        let okButton = UIAlertAction(title: "OK", style: UIAlertActionStyle.cancel,
handler: nil)
    }
}

```

```

        alert.addAction(okButton)
        self.present(alert, animated: true, completion: nil)
        // set up a warning to user in case of error
    }
}

```

15-126 'Remembering' users if logged-in already

User Defaults used to save email address (login) on phone for this app

func RememberLogin within AppDelegate.swift - to see if User Defaults has a recorded string for user, and, if so, skip signInVC and (UIStoryboard) go direct to feedVC.

```
let board : UIStoryboard(name:"Main", bundle:nil)...
```

give tabBar an ID "tabbar"

in AppDelegate.swift,

```

/// bespoke function to remember user logged in to this app
func rememberLogin()
{
    let user: String? = UserDefaults.standard.string(forKey: "user") // record
with key user

    // skip login to main app controller using UIStoryboard, if already looged-
in

    if user != nil
    {
        let board : UIStoryboard = UIStoryboard(name: "Main", bundle: nil) // /
NB type before assign/instantiate; also case sensitive "Main"
        let tabBar = board.instantiateViewController(withIdentifier: "tabBar")
as! UITabBarController // for tab bar Controller in Main.Storyboard, make up/
add an ID string in 'Identity' eg "tabBar" - use this here
        // beware auto-complete of .instantiate*Initial*ViewController :-/
        window?.rootViewController = tabBar // like dragging storyboard
entry-point arrow to TabBarController (in code)

    }
}

```

and call rememberLogin() in

```

func application(_ application: UIApplication,
didFinishLaunchingWithOptions launchOptions:
[UIApplicationLaunchOptionsKey: Any]?) -> Bool {
    // Override point for customization after application launch.

    rememberLogin()      // if user logged in already, skip signIn
ViewController screen

    FirebaseApp.configure() // this line d'require GoogleService-Info.plist in
project - working with Firebase registration
    return true
}

```

and in signInVC.signInBtnClicked and signInVC.signUpBtnClicked

```

{
    // if already logged-in, remember details
    // remembering-logged-in-type function in AppDelegate.swift
(qv)

    UserDefaults.standard.set(user!.user.email, forKey: "user") // NB
extra .user https://stackoverflow.com/a/50419010
    UserDefaults.standard.synchronize() // must synchronise

    let delegate : AppDelegate = UIApplication.shared.delegate as!
AppDelegate

    delegate.rememberLogin()

    // if no error, signed-in ok
    // go on to app
    print("successful sign in")
    // self.performSegue(withIdentifier: "toTabBar", sender: nil)
    // NB *self*.performSegue needed here because in a closure
code-block
}

```

15-127 Logging Out

drag Navigation bar to both VCs (feed/other)
custom bar button logout to FeedVC

```

@IBAction func logoutClicked(_ sender: Any) {
    UserDefaults.standard.removeObject(forKey: "user")
}

```

```

        UserDefaults.standard.synchronize()

        let signIn = self.storyboard?.instantiateViewController(withIdentifier:
    "signInVC") as! signInVC
        // need to have, using Identity Inspector, StoryboardID assigned to
    signInVC ("signInVC")
        let delegate : AppDelegate = UIApplication.shared.delegate as!
    AppDelegate
        delegate.window?.rootViewController = signIn // set start back to signIn
    having logged out

        delegate.rememberLogin() // update logged-in status (ie
not)
    }

```

15-128
new 'select picture' in Gimp of width x height that of ImageView
edit Info.plist to Add Row with 'Privacy - Photo Library Usage Description' 'To
choose a photo to upload'
in uploadVC

```

func choosePhoto()
{
    let picker = UIImagePickerController()
    picker.delegate = self // need to subclass ViewController (self) as
delegate for UIImagePickerController and UINavigationController, therefore
    picker.sourceType = .photoLibrary // could be camera as source
    present(picker, animated: true, completion: nil)
}

func imagePickerController(_ picker: UIImagePickerController,
didFinishPickingMediaWithInfo info: [String : Any]) {

    var selectedImageFromPicker: UIImage? // workaround to get original
image https://stackoverflow.com/a/53219069

    // postImage.image = info[UIImagePickerControllerEditedImage] as?
    UIImage // didn't work as original image slipped through unused - no edited
image to use

    if let editedImage = info[UIImagePickerControllerEditedImage] as?
    UIImage
    { // NB Any type from dictionary - try to cast to UIImage
        selectedImageFromPicker = editedImage
    }
}

```

```

    }
    else if let originalImage = info[UIImagePickerControllerOriginalImage] as?
        UIImage
    {
        selectedImageFromPicker = originalImage
    }

    // cautious approach here!
    if let selectedImage = selectedImageFromPicker
    {
        postImage.image = selectedImage
    }

    self.dismiss(animated: true, completion: nil)
}

```

and in upLoadVC.viewDidLoad()

```

.....
postImage.isUserInteractionEnabled = true
let recognizer = UITapGestureRecognizer(target: self, action:
#selector(uploadVC.choosePhoto))
postImage.addGestureRecognizer(recognizer)
}

```

15-129 Uploading & storing (in Firebase) images & text
in Firebase-Storage, **add a folder** called 'media'

ought to write rules to protect storage:

Security rules

By default, your rules allow all reads and writes. Once you have defined your data structure, you will have to write rules to secure the data that is specific to your app. [Learn more](#)

```

service firebase.storage {
  match /b/{bucket}/o {
    match /{allPaths=**} {
      allow read, write: if request.auth != null;
    }
  }
}

```

```
import Firebase
import FirebaseAuth
import FirebaseStorage
import FirebaseDatabase
```

parent-child hierarchy for Storage folders-files(-users)

Error in assigning downloadUrl
Optional("Object media does not exist.")
ah! in Storage/Rules,

allow read, write: if request.auth != null;
<https://firebase.google.com/docs/storage/security/start?authuser=0>

```
@IBAction func postBtnClicked(_ sender: Any) {
    //let mediaFolder = Storage.storage().reference().child("media")
    let mediaFolder =
Storage.storage().reference().child("media").child("\(UUID().uuidString).jpg")
    // folder 'media' previously formed in Storage in Firebase console
    // 'Storage' as base, 'media' as child
    // save as randomly generated name (UUID string) each time
    if let data = UIImageJPEGRepresentation(postImage.image!, 0.5)
    {
        guard let userId = Auth.auth().currentUser?.uid else
        {
            print ("User not authenticated - no can upload")
            return
        }
        // https://stackoverflow.com/a/53929727
        let StorageRef = mediaFolder.child(userId)
        //mediaFolder.child("\(UUID().uuidString).jpg").putData(data, metadata: nil)
        { (metadata, error) in // - deprecated
            StorageRef.putData(data, metadata: nil) { (metadata, error) in
                if error != nil
                {
                    let alert = UIAlertController(title: "Error", message:
                        error?.localizedDescription, preferredStyle: UIAlertControllerStyle.alert)

                    let okButton = UIAlertAction(title: "OK", style:
                        UIAlertActionStyle.cancel, handler: nil)

                    alert.addAction(okButton)

                    self.present(alert, animated: true, completion: nil)
                }
            }
        }
    }
}
```

```

        print("upload path: \$(String(describing:metadata?.path))")

    }

else // if there's no problem with the file storage, go ahead
{
    // let imageUrl = metadata?.downloadURL()?.absoluteString // deprecated

        // downloadUrl must be obtained with async completion, closure, and authentication must be used if the Firebase storage rules are set to same e.g. allow read, write: if request.auth != null; https://stackoverflow.com/a/50464820

    StorageRef.downloadURL(completion: { (url, error) in
        if error != nil
        {
            print("Error getting download url: \$(String(describing: error?.localizedDescription))"
        }
        else
        {
            print("url: \$(String(describing: url!.absoluteURL))"
        }
    })
}

// putFile(from: // , metadata: <#T##StorageMetadata?#>, completion: <#T##((StorageMetadata?, Error?) -> Void)?##((StorageMetadata?, Error?) -> Void)?##(StorageMetadata?, Error?) -> Void#>)

}
}
}

```

15-130 Database
parent/child hierarchy of database records as per storage

NB instead of `print("thing: |(myObj.myInterpolatedVariable)")`
NB `print("thing: |String(describing:myObj!myInterpolatedVariable))")` to avoid **Optional** in result. thing:**Optional(123)**

<https://stackoverflow.com/questions/47037713/unrecognized-platform-name-ios-in-lotti>

<https://idmsa.apple.com/IDMSWebAuth/signin?appldKey=891bd3417a7776362562d2197f89480a8547b108fd934911bcbea0110d07f757&path=%2Fdownload%2Fmore%2F&rv=1>
developer downloads
<https://developer.apple.com/download/more/>

15-130

Realtime Database - start in **test mode** - inclusive **rules** (not secure)
Your security rules are defined as public, so anyone can steal, modify or delete data in your database

```
{  
  "rules": {  
    ".read": true,  
    ".write": true  
  }  
}
```

- [insta-clone-firebase-5c312 users 4n4BBtwt2YX0eYo5U3RXzyMhom73 post](#)
- **post**
 - **-Lc6uBgtxoHw_UmaK29c**
 - ◆ **image:**
 - ◆
 - ◆ **postText:**
 - ◆
 - ◆ **postedBy:**
 - ◆
 - ◆ **uuid:**

```
else // if there's no problem with the file storage, go ahead  
{  
  // let imageUrl = metadata?.downloadURL()?.absoluteString //  
  deprecated  
  
  // downloadUrl must be obtained with async completion/closure,  
  and authentication must be used if the Firebase storage rules are set to same  
  e.g. allow read, write: if request.auth != null; https://stackoverflow.com/a/50464820
```

```

    StorageRef.downloadURL(completion: { (url, error) in
        if error != nil
        {
            print("Error getting download url: \(String(describing:
error?.localizedDescription))")
        }
        else
        {
            // TODO: would be nice to do this with a callback for
            completion/completion handler e.g. https://medium.com/@abhimuralidharan/methods-with-completion-callback-in-ios-swift-fd889fca86dc

            let imageUrl = url!.absoluteString // NSURL type
            inadmissible in Firebase db (not easily, anyway) so just get string of URL

            print("url: \(String(describing: imageUrl))")

            let post = ["image" : imageUrl, "postedBy" :
                Auth.auth().currentUser!.email!, "uuid" : self.uuid,
                "postText" : self.postComment.text]
            as [String: Any]
            // as dictionary ie key/value; unique post per post (as per
filename)
            // set of 'fields' to save to Firebase database

Database.database().reference().child("users").child((Auth.auth().currentUser
?.uid)!).child("post").childByAutoid().setValue(post)

            // NB syntax Auth.auth(), Storage.storage(),
Database.database()
            // form a 'table' called 'users', each user to have table of 'post'
entries, each post (with automatic ID primary key) to have field values of url,
postedBy, postText &c from app as above

            // assuming successful upload, reset controls of upload view
            self.postImage.image = UIImage(named: "select-picture.png")
            self.postComment.text = ""
            self.tabBarController?.selectedIndex = 0 // 0: viz the initial
tab view controller, to the first ViewController ie feedVC - redirect thither

        }
    })
}

```

```

https://insta-clone-firebase-5c312.firebaseio.com/users/4n4BBtwl2YX0eYo5U3RXzyMhom73/post

post
  -Lc6uBgtxoHw_UmaK29c
    image: "https://firebasestorage.googleapis.com/v0/b/ins...)"
    postText: "An Tas a nev y'M gelwir"
    postedBy: "peternjenkin@gmail.com"
    uuid: "DBE9C1CD-C4FB-4B76-9216-6DE022443DE9"
  -Lc6wAFGthHXQQIZ0mN1
    image: "https://firebasestorage.googleapis.com/v0/b/ins...)"
    postText: "An Tas a nev y'M gelwir"
    postedBy: "peternjenkin@gmail.com"
    uuid: "CAC4FA96-2F10-497C-879A-494C9F7BAC09"

```

15-131 showing rows of successive posts on Feed view
table view to fill Feed view controller

in viewDidLoad

```

tableView.delegate = self
tableView.dataSource = self
and subclass x2 on ViewController UITableViewDelegate,
UITableViewDataSource
and then as demanded , 2 delegate functions required:
numberOfRows...
cellForRowat...

```

Select TableView, in Main.storyboard, in Attributes Inspector, set '**Prototype Cells**' to '1'

Zoom in, stretch and select cell so that 'Table View Cell' in Utilities/Inspector view - set *Identity* to 'feedCell' - **NB must be 'Table View Cell' (*not* 'View' or 'Custom Class')** in top left of Utility inspector panel, and must be '*Identifier*' in Attributes Inspector, *not* '*Restoration Identity*' in Identity Inspector (confusing) - this is used in code

`tableView.dequeueReusableCell(withIdentifier: "feedCell", for: indexPath)`
`as! feedCell`

NB Ctrl+drag-outlet from prototype cell, but use 'Structure' hierarchy to

left of Main.storyboard graphic, and ctrl+drag from elements there, rather than from graphic, to be able to link to/use the class linked to the prototype cell.

<https://stackoverflow.com/q/29282447>

Drag ImageView *into prototype cell*, to try out

Fettle in position within prototype cell (constraints will work within cell)

imageview + textview (for comments) + label (for user name)

- Could define *protoype cell* under existing classes, but add new File/Cocoa Touch class - **feedCell** - of class *UITableViewCell*, - in Attribute Inspector, select cell and set its Class to 'feedCell'

delegates & subclassing in feedVC

feedCellOld/feedCell

TODO: proper notes here including code, and for next sections too
import UIKit

```
class postCell: UITableViewCell {

    // @IBOutlet weak var usernameLabel: UILabel!
    // treid manually not drag/drop

    @IBOutlet weak var postImage: UIImageView!
    @IBOutlet weak var usernameLabel: UILabel!

    @IBOutlet weak var postComment: UITextView!

    override func awakeFromNib() {
        super.awakeFromNib()
        // Initialization code
    }

    override func setSelected(_ selected: Bool, animated: Bool) {
        super.setSelected(selected, animated: animated)

        // Configure the view for the selected state
    }
}
```

```

func getDataFromServer()
{

Database.database().reference().child("users").observe(DataEventType.childAdded) { (snapshot) in
    print("snapshot.value: \(snapshot.value)")
    print("snapshot.key: \(snapshot.key)")
    // snapshot key is the user's unique id / uuid
    // snapshot value is the actual post (dictionary of field key/value pairs)
    // how to traverse structure

    let values = snapshot.value! as! NSDictionary

    let post = values["post"] as! NSDictionary
    // actually a dictionary of dictionaries (individual posts are dictionaries
    // of about 4 fields|key/value pairs)

    print(post)

    let postIds = post.allKeys // use NSDictionary method to search keys
    for posts sought

        for id in postIds
        {
            let singlePost = post[id] as! NSDictionary
            // search within fields eg 'postedBy' 'field' in post
            print("username: \(singlePost["postedBy"])")
            print("postText: \(singlePost["postText"])")
            print("image: \(singlePost["image"])")

            self.userEmailArray.append(singlePost["postedBy"] as! String)
            self.postCommentArray.append(singlePost["postText"] as! String)
            self.postImageUrlArray.append(singlePost["image"] as! String)
        }

        // refresh view after fetching data from server
        self.tableView.reloadData()
    }

    // could be childAdded, Removed, Changed, Moved
}

@IBAction func logoutClicked(_ sender: Any) {
    UserDefaults.standard.removeObject(forKey: "user")
    UserDefaults.standard.synchronize()
}

```

```

        let signIn = self.storyboard?.instantiateViewController(withIdentifier:
    "signInVC") as! signInVC
        // need to have, using Identity Inspector, StoryboardID assigned to
    signInVC ("signInVC")
        let delegate : AppDelegate = UIApplication.shared.delegate as!
    AppDelegate
        delegate.window?.rootViewController = signIn // set start back to signIn
    having logged out

        delegate.rememberLogin() // update logged-in status (ie
not)
    }
}

```

NB // freaky - for error "could not find instance of class 'feedCellOld' in language Swift "- write var first then drag & drop, direct into pre-written declaration if possible - & fill in details (if needed) - then delete spare declaration (if needed) <https://stackoverflow.com/a/34733798>

NB Ensure table row height and prototype cell row height are the same value eg '300' <https://stackoverflow.com/a/34543713>

15-132 getDataFromServer

15-133

15-134

Fast image - caching

<https://github.com/SDWebImage/SDWebImage>

need to add to Podfile:

from terminal in project directory:

pod install

(again)

close XCode - reopen .xcwkspace, check in 'pods' folder that SDWeb under Project click on top-left name of project 'Insta Clone Firebase' list, in menu/list under 'Targets' select project name - Project settings d'open....
... from project settings, select Build Phases, got to **Link Binary with**

Libraries click '+' Add, from Workspace, e.g. SDWebImage.Framework - **but this may not be necessary if cocoa install worked ok**

15-135
peternjenkin@gmail.com
passwords 123456

NB ⌘ == alt

NB Cmd+up-arrow to go up a folder in dialogue

Firebase good at synchronising across all users in realtime, good for background jobs with Google Cloud functions

Firebase bad at querying

Instagram clone using Parse and AWS
16-138 **Parse** - back end server - open source
parseplatform.github.io

Very good..

16-139 install Parse:
from the Parse web page, view on github the iOS library/SDK
pod init
in that Podfile, add the line
pod Parse
from project root (including project folders)

pod will make a **xyzApp.xcworkspace** - use .xcworkspace file from hereon as the project file with XCode **instead of** App.xcproj file.

16-140 UI laid out (extra view, rename classes & segue) as per Firebase Insta Clone

16-141 Begin configuring **Parse**

In *AppDelegate.swift* are functions to do with running application
func application(... for configuring - cf Firebase.configure() 15-126

import Parse

```

class AppDelegate: UIResponder, UIApplicationDelegate {

    var window: UIWindow?


        func application(_ application: UIApplication,
didFinishLaunchingWithOptions launchOptions:
[UIApplicationLaunchOptionsKey: Any]?) -> Bool {
            // Override point for customization after application launch.

        let config = ParseClientConfiguration {
            (ParseMutableClientConfiguration) in
            ParseMutableClientConfiguration.applicationId = ""
            ParseMutableClientConfiguration.clientKey = ""
            ParseMutableClientConfiguration.server = ""
            // NB 3 vital values for Parse server connection (applicationId,
clientKeyserver)
            // used from AWS or other host/server
            // all using ParseMutableClientConfiguration
        }
        Parse.initialize(with: config)
        let defaultACL = PFACL() // Parse File? Access Control List ??
        defaultACL.hasPublicReadAccess = true // getPublicReadAccess =
true??
        defaultACL.hasPublicWriteAccess = true // getPublicWriteAccess =
true??
        PFACL.setDefault(defaultACL, withAccessForCurrentUser: true)
        return true
    }
}

```

16-142 - AWS, keys &c

The screenshot shows the AWS homepage. At the top, there's a navigation bar with links for Contact Sales, Support, English, My Account, and a prominent orange "Create an AWS Account" button. Below the navigation is a large banner with the heading "Start Building on AWS Today". The banner text says: "Whether you're looking for compute power, database storage, content delivery, or other functionality, AWS has the services to help you build sophisticated applications with increased flexibility, scalability and reliability". Below this text are two buttons: "Create a Free Account" and "View AWS Free Tier Details". At the bottom left of the banner, there's a "GET STARTED TODAY" section for "Amazon Lightsail". On the right side of the banner, there's a "Sign up for free" button. The background of the banner features a dark blue geometric pattern of cubes.

<https://aws.amazon.com/>

appID, masterKey & serverUrl from server.js

The screenshot shows the "Select a Support Plan" page. The URL in the browser is https://portal.aws.amazon.com/billing/signup?type=resubscribe#/support. The page features the AWS logo at the top. Below it, the heading "Select a Support Plan" is displayed. A subtext states: "AWS offers a selection of support plans to meet your needs. Choose the support plan that best aligns with your AWS usage. [Learn more](#)". There are three support plan options: "Basic Plan" (Free), "Developer Plan" (From \$29/month), and "Business Plan" (From \$100/month). Each plan is represented by an icon: a briefcase for Basic, a laptop for Developer, and a smartphone for Business. Below each icon is a list of benefits. The "Basic Plan" includes: "Included with all accounts", "24/7 self-service access to forums and resources", "Best practice checks to help improve security and performance", and "Access to health status and notifications". The "Developer Plan" includes: "For early adoption, testing and development", "Email access to AWS Support during business hours", "1 primary contact can open an unlimited number of support cases", and "12-hour response time for nonproduction systems". The "Business Plan" includes: "For production workloads & business-critical dependencies", "24/7 chat, phone, and email access to AWS Support", "Unlimited contacts can open an unlimited number of support cases", and "1-hour response time for production systems".

(had to change password by following signing in to console and clicking 'forgot password' lots of captchas)

handed over mobile phone number and debit card number

EC2 - "search for AMI term": "parse" - not in quick start catalogue but in AWS Marketplace - Parse server powered by Bitnami (free tier eligible)

EC2 charges for Micro instances are free for up to 750 hours a month if you qualify for the [AWS Free Tier](#).

The screenshot shows the AWS Marketplace product page for 'Parse Server Certified by Bitnami'. The page includes a search bar, navigation links like 'Categories', 'Delivery Methods', 'Solutions', 'Migration Mapping Assistant', 'Your Saved List', 'Partners', 'Sell in AWS Marketplace', 'Amazon Web Services Home', and 'Help'. The main content area features a product icon, the title 'Parse Server Certified by Bitnami', and tabs for 'Overview', 'Pricing', 'Usage', 'Support', and 'Reviews'. Under 'Usage', there's a table showing software and infrastructure pricing for services hosted in US East (N. Virginia). The table includes columns for EC2 Instance type, Software/hr, EC2/hr, and Total/hr. The table shows the following data:

EC2 Instance type	Software/hr	EC2/hr	Total/hr
t2.nano	\$0	\$0.006	\$0.006
t2.micro	\$0	\$0.012	\$0.012
t2.small	\$0	\$0.023	\$0.023
t2.medium	\$0	\$0.046	\$0.046
t2.large	\$0	\$0.093	\$0.093
t2.xlarge	\$0	\$0.186	\$0.186

The screenshot shows the AWS EC2 instance launch wizard, Step 1: Choose an Amazon Machine Image (AMI). It displays the 'Parse Server Certified by Bitnami' AMI details. The page includes sections for 'Parse Server Certified by Bitnami' description, 'Pricing Details', 'Hourly Fees' table, 'Why use Bitnami Certified Apps?' section, 'Product Details' (By Bitnami, Customer Rating 4.5 stars (2), Latest Version 3.2.3-0 on Ubuntu 16.04, Base Operating System Linux/Unix, Ubuntu 16.04, Delivery Method 64-bit (x86) Amazon Machine Image (AMI), License Agreement End User License Agreement, On Marketplace Since 2/10/16, AWS Services Required Amazon EC2, Amazon EBS), and a 'Continue' button.

Instance Type	Software	EC2	Total
t2.nano	\$0.00	\$0.006	\$0.006/hr
t2.micro	\$0.00	\$0.013	\$0.013/hr
t2.small	\$0.00	\$0.025	\$0.025/hr
t2.medium	\$0.00	\$0.05	\$0.05/hr
t2.large	\$0.00	\$0.101	\$0.101/hr
t2.xlarge	\$0.00	\$0.202	\$0.202/hr
t2.2xlarge	\$0.00	\$0.403	\$0.403/hr
t3.nano	\$0.00	\$0.006	\$0.006/hr
t3.micro	\$0.00	\$0.011	\$0.011/hr
t3.small	\$0.00	\$0.023	\$0.023/hr
t3.medium	\$0.00	\$0.046	\$0.046/hr
t3.large	\$0.00	\$0.091	\$0.091/hr
t3.xlarge	\$0.00	\$0.182	\$0.182/hr
t3.2xlarge	\$0.00	\$0.365	\$0.365/hr
m5d.large	\$0.00	\$0.126	\$0.126/hr
m5d.xlarge	\$0.00	\$0.252	\$0.252/hr
m5d.2xlarge	\$0.00	\$0.504	\$0.504/hr

https://eu-west-1.console.aws.amazon.com/ec2/v2/home?region=eu-west-1#LaunchInstanceWizard:

AWS Services Resource Groups

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 2: Choose an Instance Type
Your computing needs.

Filter by: All instance types Current generation Show/Hide Columns

Currently selected: t3.small (Variable ECUs, 2 vCPUs, 2.5 GHz, Intel Skylake P-8175, 2 GiB memory, EBS only)

Note: The vendor recommends using a t3.small instance (or larger) for the best experience with this product.

Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
General purpose	t2.micro	1	1	EBS only	-	Low to Moderate	Yes
General purpose	t2.small	1	2	EBS only	-	Low to Moderate	Yes
General purpose	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
General purpose	t2.large	2	8	EBS only	-	Low to Moderate	Yes

Cancel Previous Review and Launch Next: Configure Instance Details

https://eu-west-1.console.aws.amazon.com/ec2/v2/home?region=eu-west-1#LaunchInstanceWizard:

AWS Services Resource Groups

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 7: Review Instance Launch

Your instances may be accessible from any IP address. We recommend that you update your security group rules to allow access from known IP addresses only.
You can also open additional ports in your security group to facilitate access to the application or service you're running, e.g., HTTP (80) for web servers. [Edit security groups](#)

AMI Details Edit AMI

Parse Server Certified by Bitnami
 This image may not be the latest version available and might include security vulnerabilities. Please check the latest, up-to-date, available version at <https://bitnami.com/stacks>.
Free tier eligible
Root Device Type: ebs Virtualization type: hvm

Hourly Software Fees: \$0.00 per hour on t2.micro instance. Additional taxes or fees may apply.
Software charges will begin once you launch this AMI and continue until you terminate the instance.

By launching this product, you will be subscribed to this software and agree that your use of this software is subject to the pricing terms and the seller's [End User License Agreement](#)

Instance Type Edit instance type

Cancel Previous Launch

Screenshot of the AWS Launch Instance Wizard Step 7: Review Instance Launch.

The page shows the configuration for launching an instance of Parse Server Certified by Bitnami. The instance type is t2.micro, with 1 vCPU and 1 GiB of memory, using EBS storage. The network performance is set to Low to Moderate. The instance is in the Free tier eligible category.

Step 7: Review Instance Launch

AMI Details

Parse Server Certified by Bitnami

This image may not be the latest version available and might include security vulnerabilities. Please check the latest, up-to-date, available version at <https://bitnami.com/stacks>.

Root Device Type: ebs Virtualization type: hvm

Hourly Software Fees: \$0.00 per hour on t2.micro instance. Additional taxes or fees may apply.
Software charges will begin once you launch this AMI and continue until you terminate the instance.

By launching this product, you will be subscribed to this software and agree that your use of this software is subject to the pricing terms and the seller's End User License Agreement

Instance Type

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	Variable	1	1	EBS only	-	Low to Moderate

Launch

Screenshot of the AWS Launch Instance Wizard Step 7: Review Instance Launch.

The page shows the configuration for launching an instance of Parse Server Certified by Bitnami. The instance type is t2.micro, with 1 vCPU and 1 GiB of memory, using EBS storage. The network performance is set to Low to Moderate. The instance is in the Free tier eligible category.

Step 7: Review Instance Launch

AMI Details

Parse Server Certified by Bitnami

This image may not be the latest version available and might include security vulnerabilities. Please check the latest, up-to-date, available version at <https://bitnami.com/stacks>.

Root Device Type: ebs Virtualization type: hvm

Hourly Software Fees: \$0.00 per hour on t2.micro instance. Additional taxes or fees may apply.
Software charges will begin once you launch this AMI and continue until you terminate the instance.

By launching this product, you will be subscribed to this software and agree that your use of this software is subject to the pricing terms and the seller's End User License Agreement

Instance Type

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	Variable	1	1	EBS only	-	Low to Moderate

Launch

Feedback English (US)

Launch Status

YOUR INSTANCES ARE NOW LAUNCHING

The following instance launches have been initiated: i-063092a3e4a656fe1 [View launch log](#)

Get notified of estimated charges

Create billing alerts to get an email notification when estimated charges on your AWS bill exceed an amount you define (for example, if you exceed the free usage tier).

How to connect to your instances

Your instances are launching, and it may take a few minutes until they are in the **running** state, when they will be ready for you to use. Usage hours on your new instances will start immediately and continue to accrue until you stop or terminate your instances.

Click [View Instances](#) to monitor your instances' status. Once your instances are in the **running** state, you can [connect](#) to them from the Instances screen. [Find out](#) how to connect to your instances.

Getting started with your software

To get started with Parse Server Certified by Bitnami [To manage your software subscription](#)

Feedback English (US)

© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#)

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with navigation links for EC2 Dashboard, Events, Tags, Reports, Limits, Instances (selected), Launch Templates, Spot Requests, Reserved Instances, Dedicated Hosts, Scheduled Instances, Capacity Reservations, Images (AMIs, Bundle Tasks), and Elastic Block Store. The main content area has tabs for Launch Instance, Connect, and Actions. A search bar at the top right says "search : i-063092a3e4a656fe1" and "Add filter". Below it is a table with columns: Name, Instance ID, Instance Type, Availability Zone, Instance State, Status Checks, Alarm Status, and Public DNS (IPv4). One row is selected, showing "i-063092a3e4a656fe1", "t2.micro", "eu-west-1b", "running", "2/2 checks ...", "None", and "ec2-34-247-48-250". At the bottom, there's a detailed view for the selected instance, showing its Public DNS as "ec2-34-247-48-250.eu-west-1.compute.amazonaws.com".

TODO: EC2 AIM (machine) types

(1) tick t2.micro (free tier) 1Gb

could do next: (3) Configure Instance (network, subnet)(4) Storage. (5) Tags (key/value). (6) Security Group (and rules for protocols/ports) (see Everest notes)

'review & launch' - ignore 'improve your instance's security'

Launch...

Select or Create a new key pair 'Create a new key/pair' - give key pair a name, eg '*Parse Insta Clone*' - 'Download Key Pair' -> 1-off .pem file will be downloaded - eg ParseInstaClone.pem - pasted into folder (maybe not push/save to repo though TODO: cf vaultproject, etcd)

'Launch Instance'

(from 'View Usage Instructions' link on AWS page)

3.2.3-0 on Ubuntu 16.04 Usage Instructions for Parse Server Certified by Bitnami

Once the instance is running, enter the public DNS provided by Amazon into your browser. You will then see the Parse Server application. The default server administrator is '**user**'. Please check our documentation at <https://docs.bitnami.com/aws/faq/get-started/find-credentials/> to learn how to get your password. You may change this username and password within the application settings. You can also access your instance via SSH using the username '**bitnami**' and your Amazon *private key*. For additional setup instructions and frequently asked questions please go to <https://docs.bitnami.com/aws/apps/parse/>

AWS EC2 console - Instance/Instance Settings/Get System Log

eg

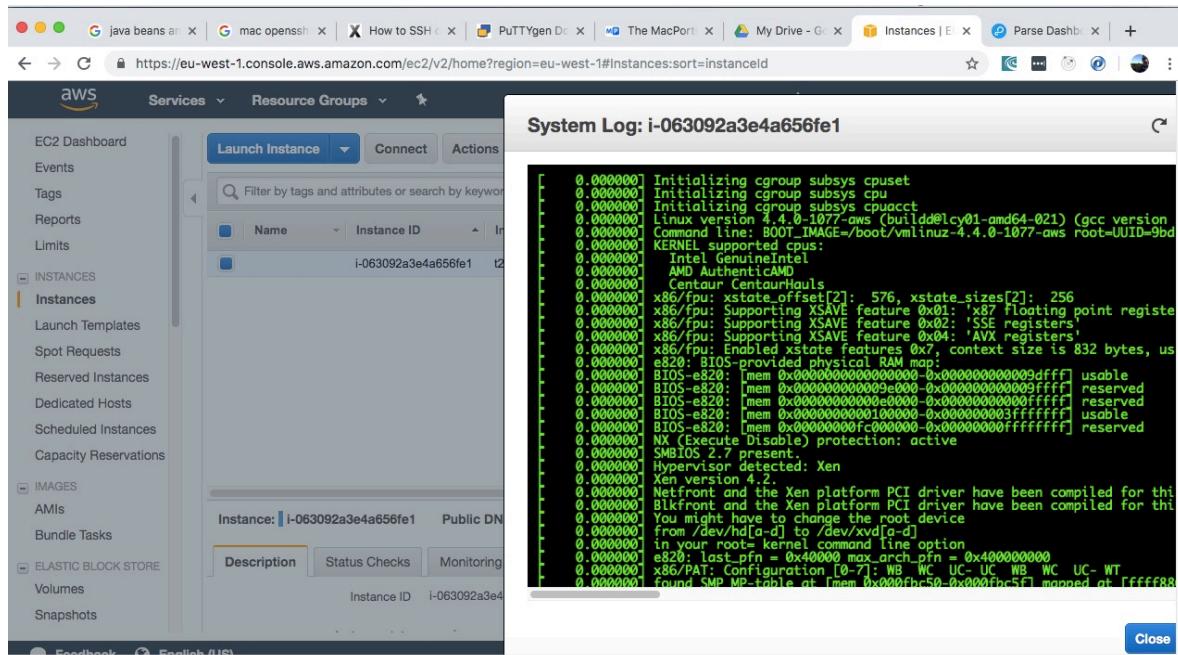
....

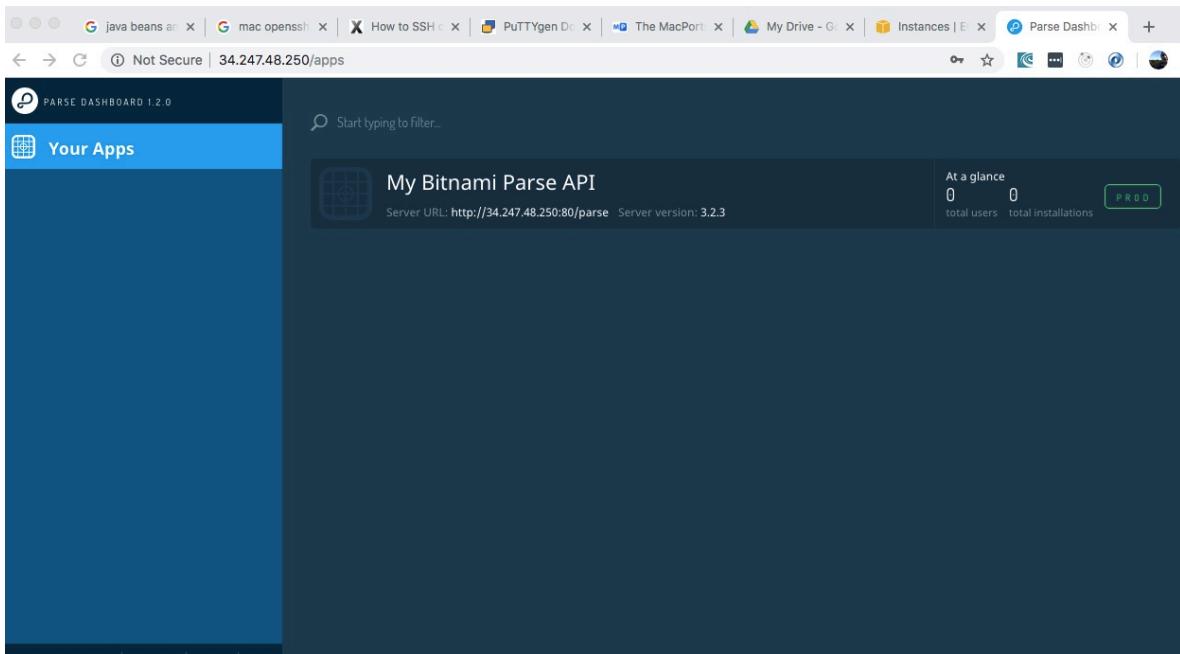
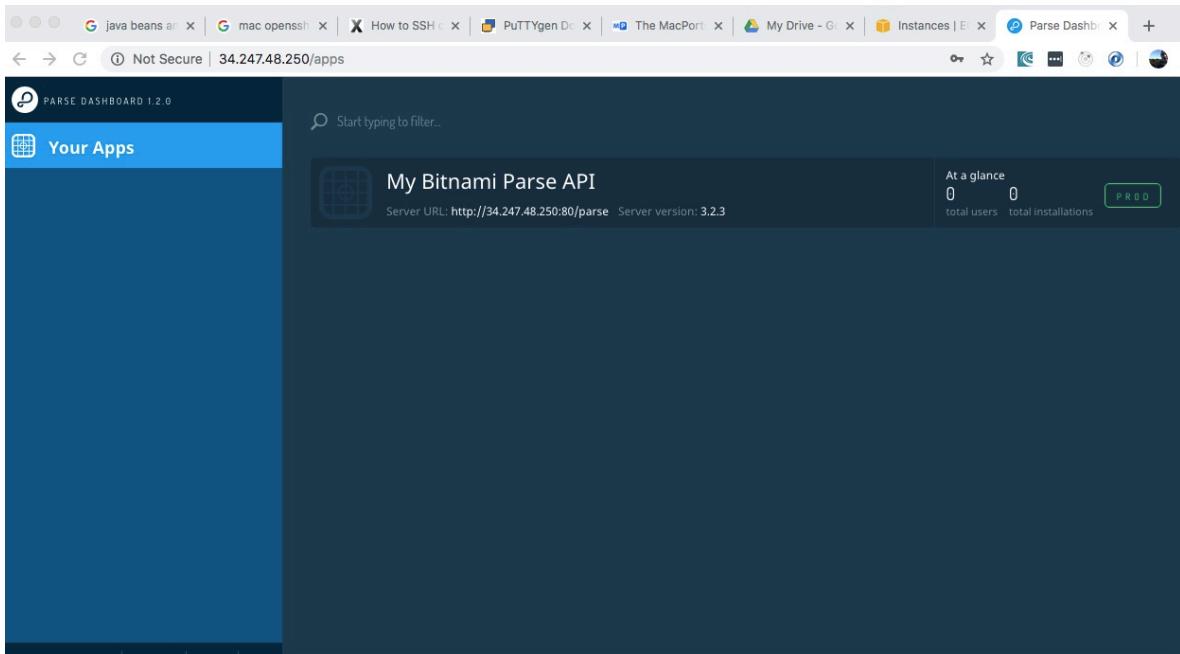
[22.667619] bitnami[1110]: ## 2019-04-11 17:36:02+00:00 ## INFO ## 443

```

has been blocked
[ 22.968709] bitnami[1110]:
#####
##### [ 22.981919] bitnami[1110]: # #
[ 23.004835] bitnami[1110]: #      Setting Bitnami application password to
'ammfOJ3HCNgj'      #
[ 23.025157] bitnami[1110]: #      (the default application username is 'user')
#
[ 23.044059] bitnami[1110]: # #
[ 23.057083] bitnami[1110]:
#####
##### [ 30.834006] bitnami[1110]: 650000+0 records in
[ 30.840981] bitnami[1110]: 650000+0 records out

```

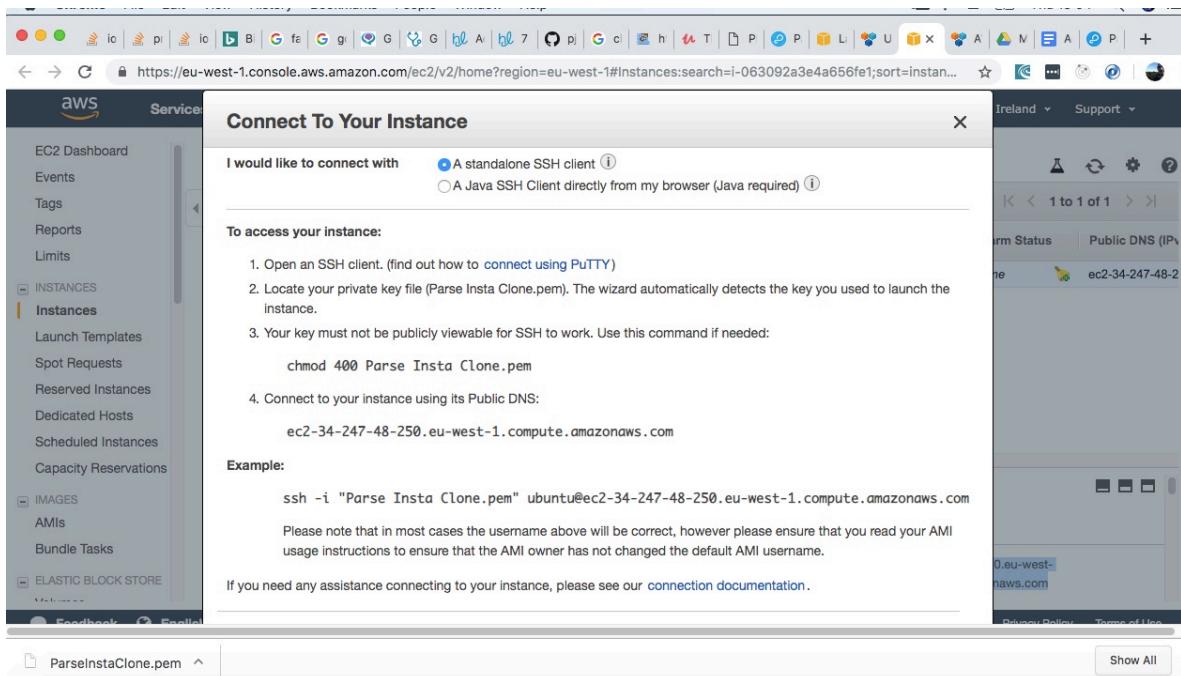




16-143 in list of EC2 instances, at bottom when Ex2 instance selected, at bottom- right is public DNS of instance e.g. ec2-34-247-48-250.eu-west-1.compute.amazonaws.com

append /apps e.g. <http://ec2-34-247-48-250.eu-west-1.compute.amazonaws.com/apps>

in EC2 console, tick instance and click 'Connect' button at top
in this case, although 'standalone SSH client' is an option

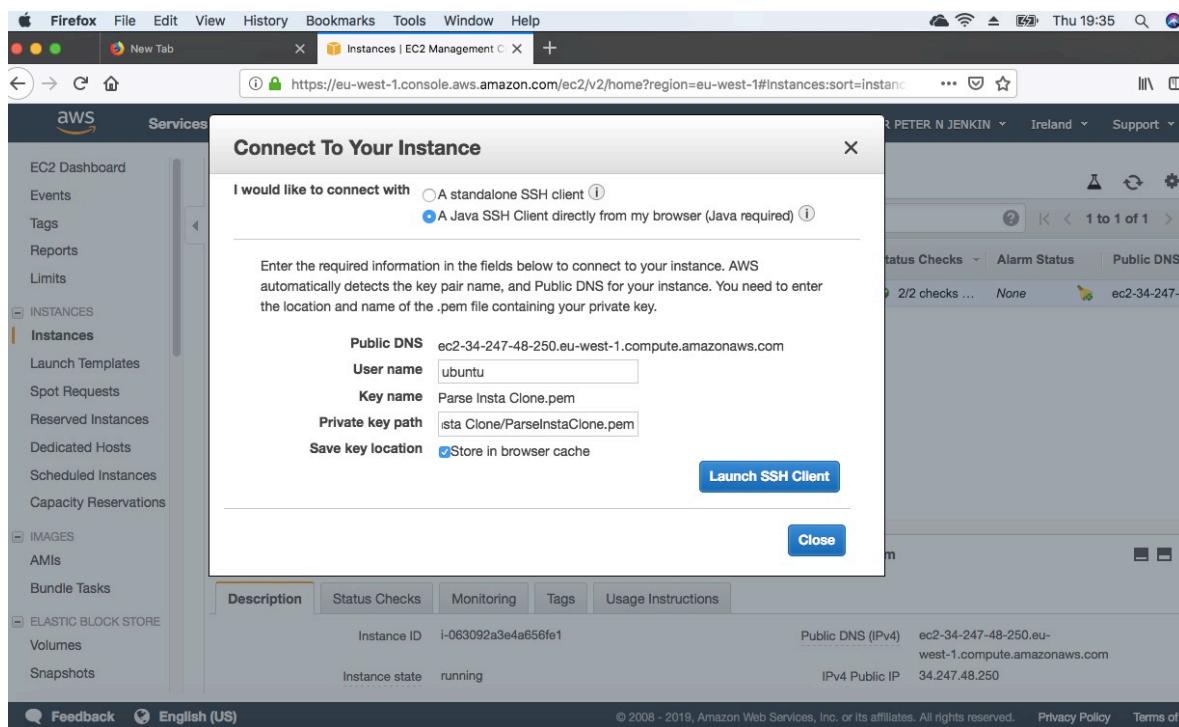
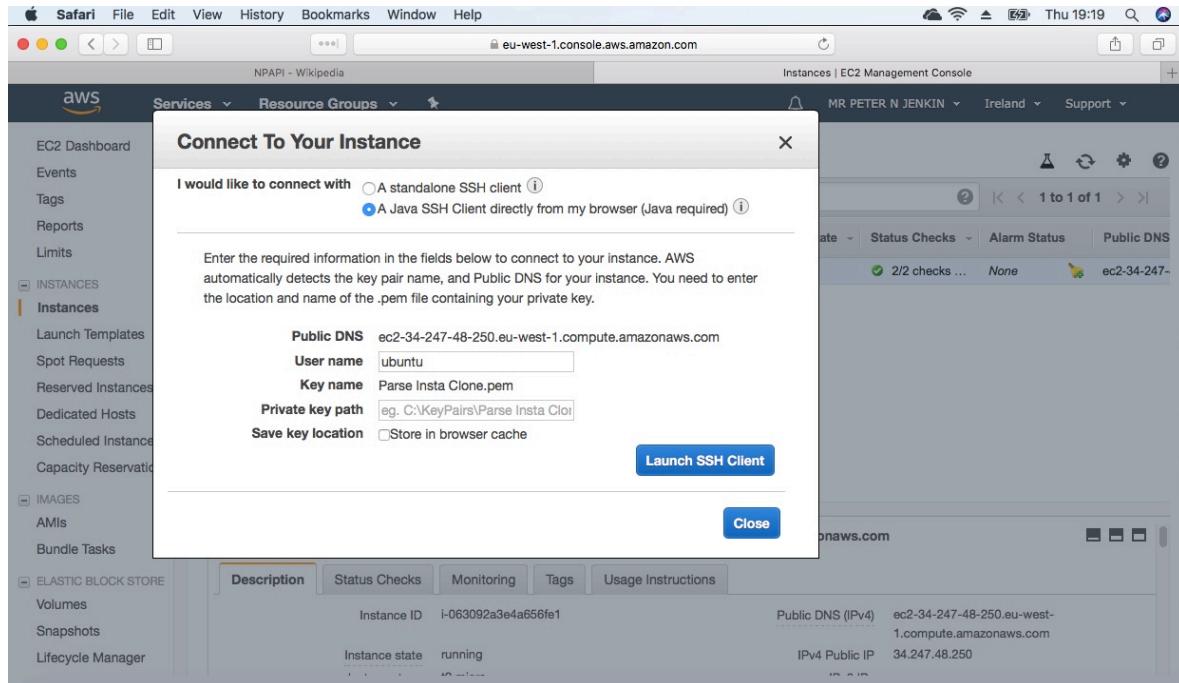


Connect To Your Instance	
	<p>I would like to connect with</p> <p>A standalone SSH client</p> <p>A Java SSH Client directly from my browser (Java required)</p> <p>NPAPI deprecation on Chrome</p> <p>Chrome no longer supports plugins using NPAPI, which will prevent the in-browser Java SSH client from loading. To continue to SSH into your EC2 instances from your browser, use Firefox, Internet Explorer 9 or higher, or Safari. Learn more about NPAPI deprecation on Chrome.</p>

Netscape Plugin Application Programming Interface (**NPAPI**) is a deprecated (see below) application programming interface (API) that allows browser extensions to be developed. It was first developed for Netscape browsers, starting in 1995 with Netscape Navigator 2.0, but was subsequently adopted by other browsers. <https://en.wikipedia.org/wiki/NPAPI>

In NPAPI architecture, a plugin declares **content types** (e.g. "audio/mp3") that it can handle. When the browser encounters a content type it cannot handle natively, it loads the appropriate plugin, sets aside space within the browser context for the plugin to render and then streams data to it. The plugin is responsible for rendering the data. The plugin runs in-place within the page, as opposed to older browsers that had to launch an external application to handle

unknown content types.



using Safari instead for Java SSH client via browser
using Firefox instead for Java SSH client via browser

NB to enable Safari Java (for SSH client &c), Preferences/Websites, tick 'Java' in 'plugins' bottom left, then tick, e.g. AWS website in list on right

from right-click on .pem file, 'Get Info' , highlight 'Where' and Cmd+C copy all of the path - pste this into 'Private Key Path' text input box in SSH browser dialogue, and append '/' and file name e.g. '/ParseInstaClone.pem'. to make

e.g. '/Users/peterjenkin/Projects/Mac/completeios10course/16-17ParsePushInstaClone/Parse Insta Clone/Parse Insta Clone/ParseInstaClone.pem'

Tick box 'Store in browser cache' to keep the key available for use in browser (via its cache) in future.

Keep other dialogue inputs the same as presented.

or use SSH client eg Cyberduck, Putty

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/putty.html?icmpid=docs_ec2_console

To access your instance:

1. Open an SSH client. (find out how to [connect using PuTTY](#))
2. Locate your private key file (Parse Insta Clone.pem). The wizard automatically detects the key you used to launch the instance.
3. Your key must not be publicly viewable for SSH to work. Use this command if needed:`chmod 400 Parse Insta Clone.pem`
4. Connect to your instance using its Public DNS:`ec2-34-247-48-250.eu-west-1.compute.amazonaws.com`

Example:

```
ssh -i "Parse Insta Clone.pem" ubuntu@ec2-34-247-48-250.eu-west-1.compute.amazonaws.com
```

Please note that in most cases the username above will be correct, however please ensure that you read your AMI usage instructions to ensure that the AMI owner has not changed the default AMI username.

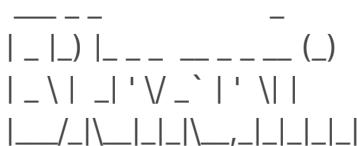
If you need any assistance connecting to your instance, please see our [connection documentation](#).

```
PNJ-dev-Mac-09: ssh -i "ParseInstaClone.pem"
ubuntu@ec2-34-247-48-250.eu-west-1.compute.amazonaws.com
@@@@@@@WARNING: UNPROTECTED PRIVATE KEY FILE! @
Permissions 0644 for 'ParseInstaClone.pem' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
Load key "ParseInstaClone.pem": bad permissions
ubuntu@ec2-34-247-48-250.eu-west-1.compute.amazonaws.com: Permission denied (publickey).
```

```

PNJ-dev-Mac-09:Parse Insta Clone peterjenkin$ chmod 400
ParseInstaClone.pem
PNJ-dev-Mac-09:Parse Insta Clone peterjenkin$ ssh -i
"ParseInstaClone.pem" ubuntu@ec2-34-247-48-250.eu-
west-1.compute.amazonaws.com
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.4.0-1077-aws x86_64)
*** System restart required ***

```



```

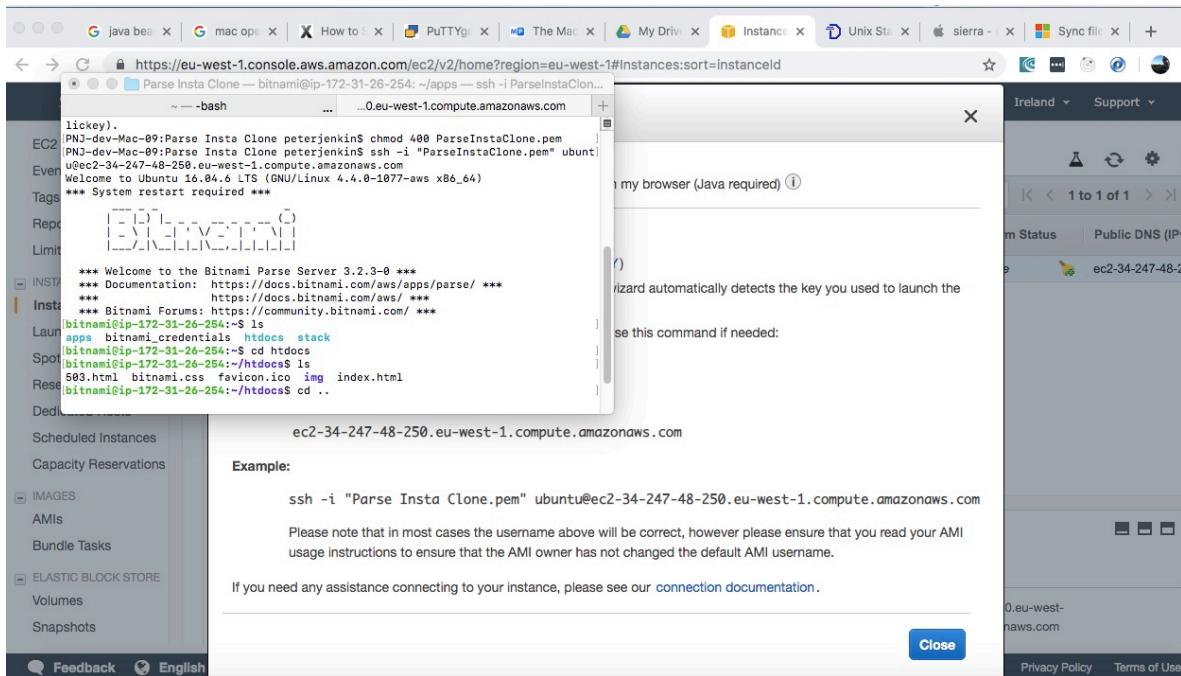
*** Welcome to the Bitnami Parse Server 3.2.3-0 ***
*** Documentation: https://docs.bitnami.com/aws/apps/parse/ ***
*** https://docs.bitnami.com/aws/ ***
*** Bitnami Forums: https://community.bitnami.com/ ***
bitnami@ip-172-31-26-254:~$
```

400 == owner: read-only, group & world: nothing
<https://www.decmember.com/unix/ref/chmod.html>

- 0 = no permissions whatsoever; this person cannot read, write, or execute the file
- 1 = execute only
- 2 = write only
- 3 = write and execute (1+2)
- 4 = read only
- 5 = read and execute (4+1)
- 6 = read and write (4+2)
- 7 = read and write and execute (4+2+1)

Chmod commands on file apple.txt (use wildcards to include more files)

Command	Purpose
chmod 700 apple.txt	Only you can read, write to, or execute apple.txt
chmod 777 apple.txt	Everybody can read, write to, or execute apple.txt
chmod 744 apple.txt	Only you can read, write to, or execute apple.txt Everybody can read apple.txt;
chmod 444 apple.txt	You can only read apple.txt, as everyone else.



cd & ls into **apps/parse/htdocs/** and edit (vi?) or cat **server.js** (so, **apps/parse/htdocs/server.js**), and look through this to find values of keys, e.g.

```
appId: "4db3417ac4d79c34640760d3c13c3728cd26604a",
masterKey: "571c1e762705f9975f4aaf4f91cb373d6e23d9f",
fileKey: "a1fce6651b8726a4ef991570632615baf128d9f3",
production: true,
serverURL: "http://34.247.48.250:80/parse"
```

```
let config = ParseClientConfiguration {
  (ParseMutableClientConfiguration) in
  ParseMutableClientConfiguration.applicationId =
  "4db3417ac4d79c34640760d3c13c3728cd26604a"
  ParseMutableClientConfiguration.clientKey =
  "571c1e762705f9975f4aaf4f91cb373d6e23d9f"
  ParseMutableClientConfiguration.server = "" //
```

//in server.js - applicationId: appId ; clientKey: masterKey; server:
serverURL

**// NB serverUrl will change if free-tier EC2 instance stopped & re-started
- need to SSH in again and find server.js and/or check AWS EC2 console
re Public DNS - errors regarding connection failure may be seen**

```
PNJ-dev-Mac-09:Parse Insta Clone peterjenkin$ chmod 400
ParseInstaClone.pem
PNJ-dev-Mac-09:Parse Insta Clone peterjenkin$ ssh -i "ParseInstaClone.pem"
ubuntu@ec2-34-247-48-250.eu-west-1.compute.amazonaws.com
```

Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.4.0-1077-aws x86_64)

*** System restart required ***



*** Welcome to the Bitnami Parse Server 3.2.3-0 ***

*** Documentation: <https://docs.bitnami.com/aws/apps/parse/> ***

*** <https://docs.bitnami.com/aws/> ***

*** Bitnami Forums: <https://community.bitnami.com/> ***

bitnami@ip-172-31-26-254:~\$ ls

apps bitnami_credentials htdocs stack

bitnami@ip-172-31-26-254:~\$ cd htdocs

bitnami@ip-172-31-26-254:~/htdocs\$ ls

503.html bitnami.css favicon.ico **img index.html**

bitnami@ip-172-31-26-254:~/htdocs\$ cd ..

bitnami@ip-172-31-26-254:~\$ cd apps

bitnami@ip-172-31-26-254:~/apps\$ ls

bitnami parse rockmongo

bitnami@ip-172-31-26-254:~/apps\$ ls

bitnami parse rockmongo

bitnami@ip-172-31-26-254:~/apps\$ cd parse

bitnami@ip-172-31-26-254:~/apps/parse\$ ls

bnconfig conf htdocs licenses scripts updateip

bitnami@ip-172-31-26-254:~/apps/parse\$ cd htdocs

bitnami@ip-172-31-26-254:~/apps/parse/htdocs\$ ls

logs node_modules package.json package-lock.json server.js

bitnami@ip-172-31-26-254:~/apps/parse/htdocs\$ cat server.js

var express = require('express');

var ParseServer = require('parse-server').ParseServer;

var app = express();

// Specify the connection string for your mongodb database

// and the location to your Parse cloud code

var api = new ParseServer({

databaseURI: "mongodb://root:ammfOJ3HCNgj@127.0.0.1:27017/

bitnami_parse",

cloud: "./node_modules/parse-server/lib/cloud-code/Parse.Cloud.js",

appId: "4db3417ac4d79c34640760d3c13c3728cd26604a",

masterKey: "571c1e762705f9975f4aafd4f91cb373d6e23d9f",

fileKey: "a1fce6651b8726a4ef991570632615baf128d9f3",

serverURL: "http://34.247.48.250:80/parse"

});

// Serve the Parse API on the /parse URL prefix

app.use('/parse', api);

```

var port = 1337;
app.listen(port, function() {
  console.log('parse-server running on port ' + port);
});

//Parse Dashboard
var ParseDashboard = require('parse-dashboard');
var dashboard = new ParseDashboard({
  apps: [
    {
      appName: "My Bitnami Parse API",
      appId: "4db3417ac4d79c34640760d3c13c3728cd26604a",
      masterKey: "571c1e762705f9975f4aaf4f91cb373d6e23d9f",
      fileKey: "a1fce6651b8726a4ef991570632615baf128d9f3",
      production: true,
      serverURL: "http://34.247.48.250:80/parse"
    }
  ],
  users: [
    {
      user: "user",
      pass:
      "$2a$10$fW6IQJ2rZL5dTMFwq277Ou0En2YjMfpwSxB6BupSHYtt7V./Yf0iO"
    }
  ], useEncryptedPasswords: true
});

var allowInsecureHTTP = true;

// Serve the Parse Dashboard on the /parsedashboard URL prefix
app.use('/', dashboard);

var portdash = 4040;
app.listen(portdash, function() {
  console.log('parse-dashboard running on port ' + portdash);
});

```

NB in case of SSH error:

```

PNJ-dev-Mac-09:Parse Insta Clone peterjenkin$ ssh -i "ParseInstaClone.pem"
ubuntu@ec2-34-247-48-250.eu-west-1.compute.amazonaws.com
/Users/peterjenkin/.ssh/config: line 3: Bad configuration option:
usekeychain

```

/Users/peterjenkin/.ssh/config: terminating, 1 bad configuration options

, in hidden directory/file **~/.ssh/config** comment out line with "UseKeyChain yes"
e.g.

```
Host *
  AddKeysToAgent yes
# UseKeyChain yes
 IdentityFile ~/.ssh/id_rsa
```

usekeychain was added by Apple in macOS 10.12.2. https://developer.apple.com/library/archive/technotes/tn2449_index.html to enable using the Mac's keychain

(possibly after installing OpenSSH?)

put these values into variables in the AppDelegate.swift file
e.g.

```
let config = ParseClientConfiguration {
    (ParseMutableClientConfiguration) in
    ParseMutableClientConfiguration.applicationId =
"4db3417ac4d79c34640760d3c13c3728cd26604a"
    ParseMutableClientConfiguration.clientKey =
"571c1e762705f9975f4aaf4f91cb373d6e23d9f"
    ParseMutableClientConfiguration.server = "http://34.247.48.250:80/
parse"

    // Parse server values obtained from apps/parse/htdocs/server.js (via
    SSH log-in to AWS EC2 instance)
    // applicationId == appId; clientKey == masterKey; server == serverURL
```

To debug **Parse's** errors, check the **log files** of **Parse** at /opt/bitnami/apps/parse/htdocs/logs/.

<https://docs.bitnami.com/aws/apps/parse/troubleshooting/debug-errors/>
<https://community.bitnami.com/t/how-to-enable-parse-server-logs/43763>

Windows login to Linux EC2 on AWS

Locate **.pem** file from EC2 instance formation

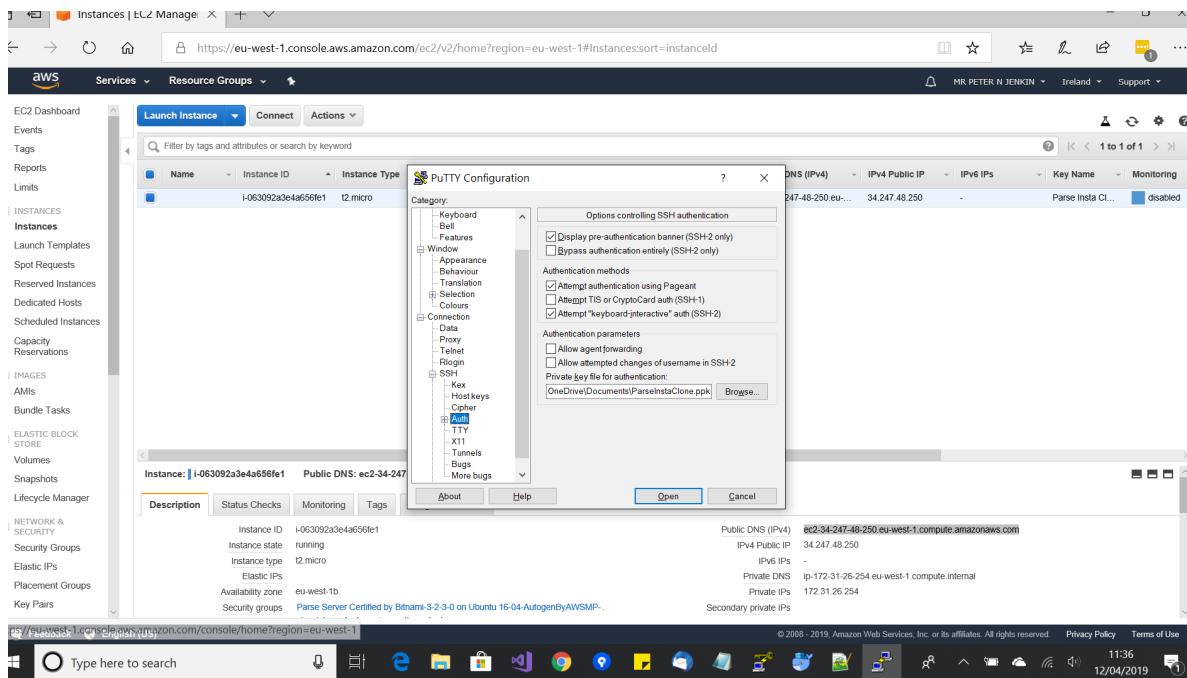
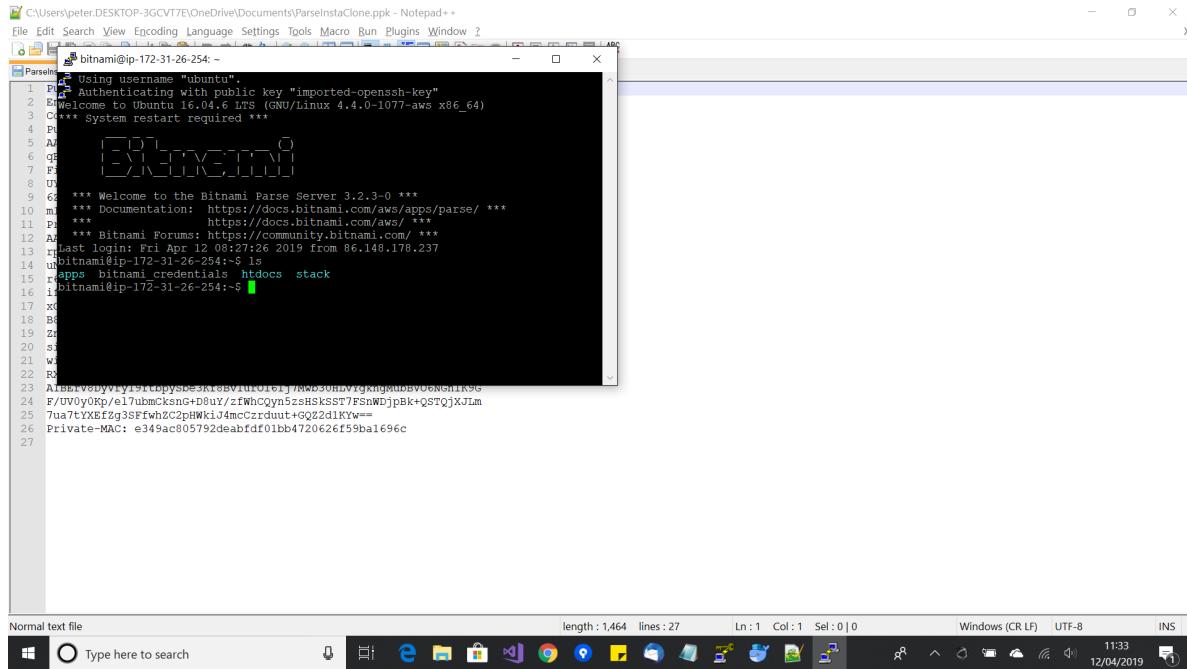
NB **PuttyGen** - Load - All Files ** - select .pem file eg 'ParseInstaClone.pem' - Save Private Key (this is to be used for) - save with same name eg 'ParseInstaClone.ppk'

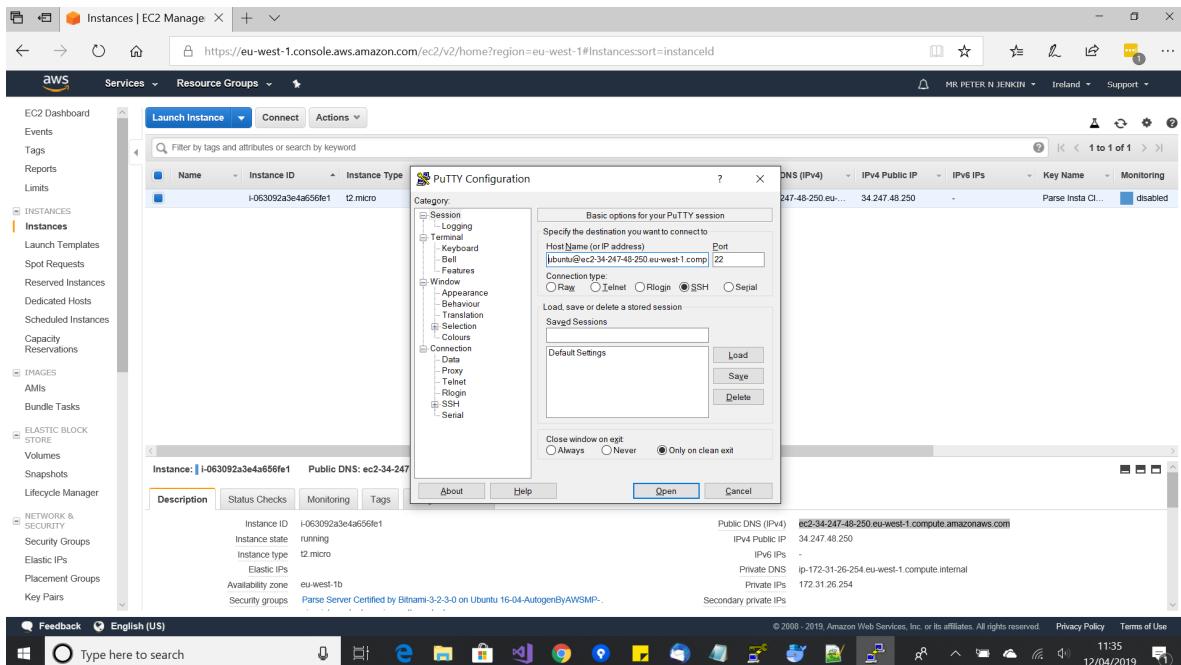
Open **PuTTY**, enter for host eg [ubuntu@ec2-34-247-48-250.eu-](ubuntu@ec2-34-247-48-250.eu)

west-1.compute.amazonaws.com

in PuTTY hierarchy on left, expand/navigate tree to *Connection/SSH/Auth* and in '*Private key file for authentication*' , select the **.ppk** file saved previously. eg '**ParselInstaClone.ppk**'

Click 'Open' in PuTTY - PuTTY window will show Linux box's terminal prompt





TODO: AMI?

16-144

Because url http not https,
in Info.plist, rt-click & 'Add Row', select 'App Transport Security Settings' -
expand (arrow at left of row to down) and click '+' at end of row cell - in newly
opened row, select 'Allow Arbitrary Loads' (1 of 3?) and type in the Value
column (from 'NO') 'YES'

The screenshot shows the Parse Dashboard. The left sidebar has sections for 'Core' (Browser, Role, User, Animals, Webhooks, Jobs, Logs, Config, API Console), 'Push' (with a status of 'Not Secure'), and links to 'Open Source Hub', 'GitHub', and 'Docs'. The main area shows a table for the 'Animals' class. The table has columns: CreatedAt, updatedAt, ACL, age, colour, and name. There is one object listed:

	CreatedAt	updatedAt	ACL	age	colour	name
	12 Apr 2019 at 10:...	12 Apr 2019 at 10:...	Public Read + Write	5	Brown	Fungal Ferret

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

```

// temporary code to check Parse connection working ok - should see
result in Parse web console
let checkObject = PFObject(className: "Animals")      // add a class
// class names visible in Parse dashboard (eg http://34.247.48.250/apps/)
a la tables
checkObject["name"] = "Fungal Ferret"
checkObject["colour"] = "Brown"
checkObject["age"] = 5
checkObject.saveInBackground()    // avoid having to write do ... try
block
{
  {(success, error) in
    if error != nil{
      print(error?.localizedDescription)
    }
    else
    {
      print(success)
    }
  }
}
// Do any additional setup after loading the view.
}

```

TODO: definition fo closures, completions, completion handlers

Bitnami - Seville - stacks - virtual machines - <https://en.wikipedia.org/wiki/Bitnami>

Rockmongo - admin for mongodb as used by Parse (or PostGres) - <https://www.ostechnix.com/rockmongo-graphical-mongodb-administration-tool/>

```

let query = PFQuery(className: "Animals")
//query.whereKey(<#T##key: String##String#>, equalTo: <#T##Any#>)
query.whereKey("name", equalTo: "Fungal Ferret")
query.findObjectsInBackground
{
  (objects, error) in
  if error != nil
  {
    print(error?.localizedDescription)
  }
  else
  {
    print(objects)    // check the debug output console
  }
}

```

```
// Do any additi
```

16-145 User sign-up with PFUser for Parse. (cf 15-124 et seq for Firebase)
ude - easy!

```
@IBAction func signUpBtnClicked(_ sender: Any) {  
    let user = PFUser()      // set up a Parse user  
    user.username = usernameText.text  
    user.password = passwordText.text  
    // user["age"] // example possible usage with Parse (for non-username or  
    non-password)  
    user.signUpInBackground {  
        (success, error) in  
        if error != nil  
        {  
            print(error?.localizedDescription)  
        }  
        else  
        {  
            print("User has been added")  
        }  
    }  
}
```

16-146 User sign-in using Parse server instance (cf 15-124 for Firebase)

NB S3 bucket

IAM user - programmatic access and/or AWS Management Console access\
– as opposed to '**root user**' eg peterjenkin@gmail.com

programmatic access Enables an **access key ID** and **secret access key** for
the AWS API, CLI, SDK, and other development tools.

permissions boundary optional

User

Access key ID

Secret access key

Email login instructions

Access Key and Secret Access Key
peternjenkin
AKIARTMSUSLQLXDBBWWU
WAS+R6HsPTLERd1Gb9nCoPOgQ7wTC+OHeWt+dLMb Hide

used Cyberduck with **Access Key** and **Secret Access Key**

made an iam user account of peternjenkin with full S3 and EC2 policies as far as known (about half a dozen)

install AWS Command Line Interface (using brew)
brew install awscli

```
aws configure
AWS Access Key ID [*****]:
AWS Secret Access Key [*****]:
Default region name [eu-west-1]:
Default output format [None]:
```

list all S3 buckets with

```
aws s3 ls
```

used Cyberduck with Access Key and Secret Access Key

16-147 Sign-In using Parse server instance (on EC2) (cf 15-125 for Firebase)
// NB serverUrl will change if free-tier EC2 instance stopped & re-started - need to SSH in again and find server.js and/or check AWS EC2 console re Public DNS - errors regarding connection failure may be seen

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

    if usernameText.text != "" && passwordText.text != ""
    {
        PFUser.logInWithUsername(inBackground: usernameText.text!,
password: passwordText.text!) {(user, error) in
            if error != nil
            {
                print(error?.localizedDescription)
            }
            else
            {
                print("\(user?.username) is now logged in")
            }
        }
    }
}
```

```
        self.performSegue(withIdentifier: "toTabBarVC", sender: nil)
    }
}
}
}
```

16-148 remember logged-in user via UserDefaults (cf 15-126 for Firebase)

NB if 'entry point' arrow lost, causing error, in Attributes Inspector select '*is Initial View Controller*'

16-149 logging out a user (rememberind and redirecting also)

```
@IBAction func logoutBtnClicked(_ sender: Any) {
    PFUser.logOutInBackground {(error) in
        if error != nil {
            let alert = UIAlertController(title: "Error", message:
error?.localizedDescription, preferredStyle: UIAlertControllerStyle.alert)
                // set up button ready for user acknowledgement on alert
                let button = UIAlertAction(title: "OK", style: UIAlertActionStyle.cancel,
handler: nil)
                alert.addAction(button)
                // show alert with button
                self.present(alert, animated: true, completion: nil)

        }
    } else {
        UserDefaults.standard.removeObject(forKey: "userLoggedIn") // remember that no user is now logged-in

        UserDefaults.standard.synchronize() // update the UserDefaults db

        let signUp = self.storyboard?.instantiateViewController(withIdentifier:
"signUpVC") as! signUpVC

        // refer to AppDelegate.swift code
        let delegate : AppDelegate = UIApplication.shared.delegate as!
AppDelegate
        delegate.window?.rootViewController = signUp

        delegate.rememberLogin()

    }
}
```

```
}
```

16-150 setting up Table view in feed view controller (cf 15-131 for Firebase)

16-151 setting up Prototype feed cells (cf 15-132)

Drag in control and tableView - Set Prototype Cells to '**1**'

NB select Prototype Cell- set **Identifier** to "**Cell**"

NB Identifier = "**Cell**" - use Structure (hierarchy) view -

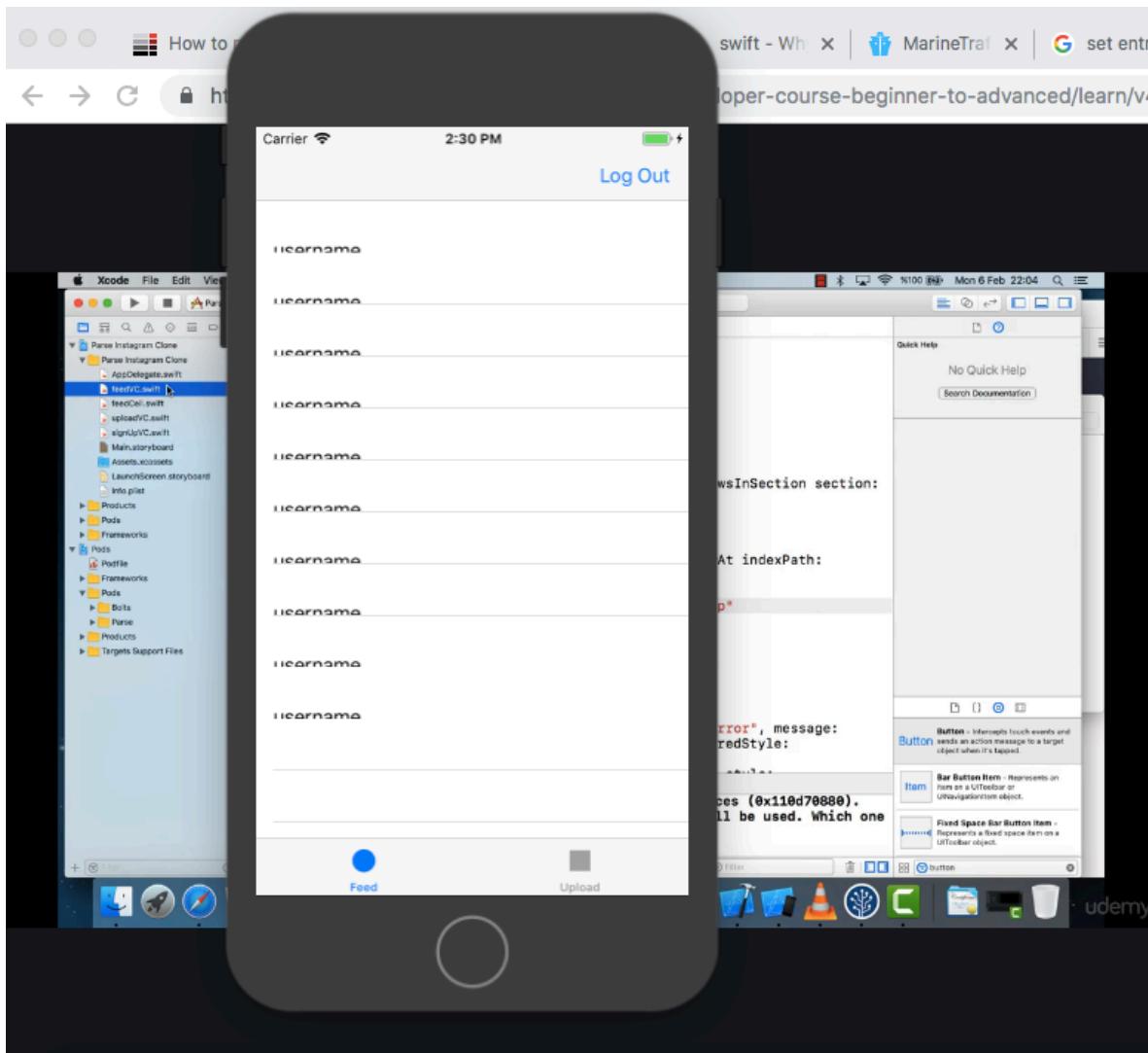
File/New/File - Cocoa Touch Class - feedCell (UITableViewCell) - select Cell in Structure/hierarchy then set **Class = feedCell**

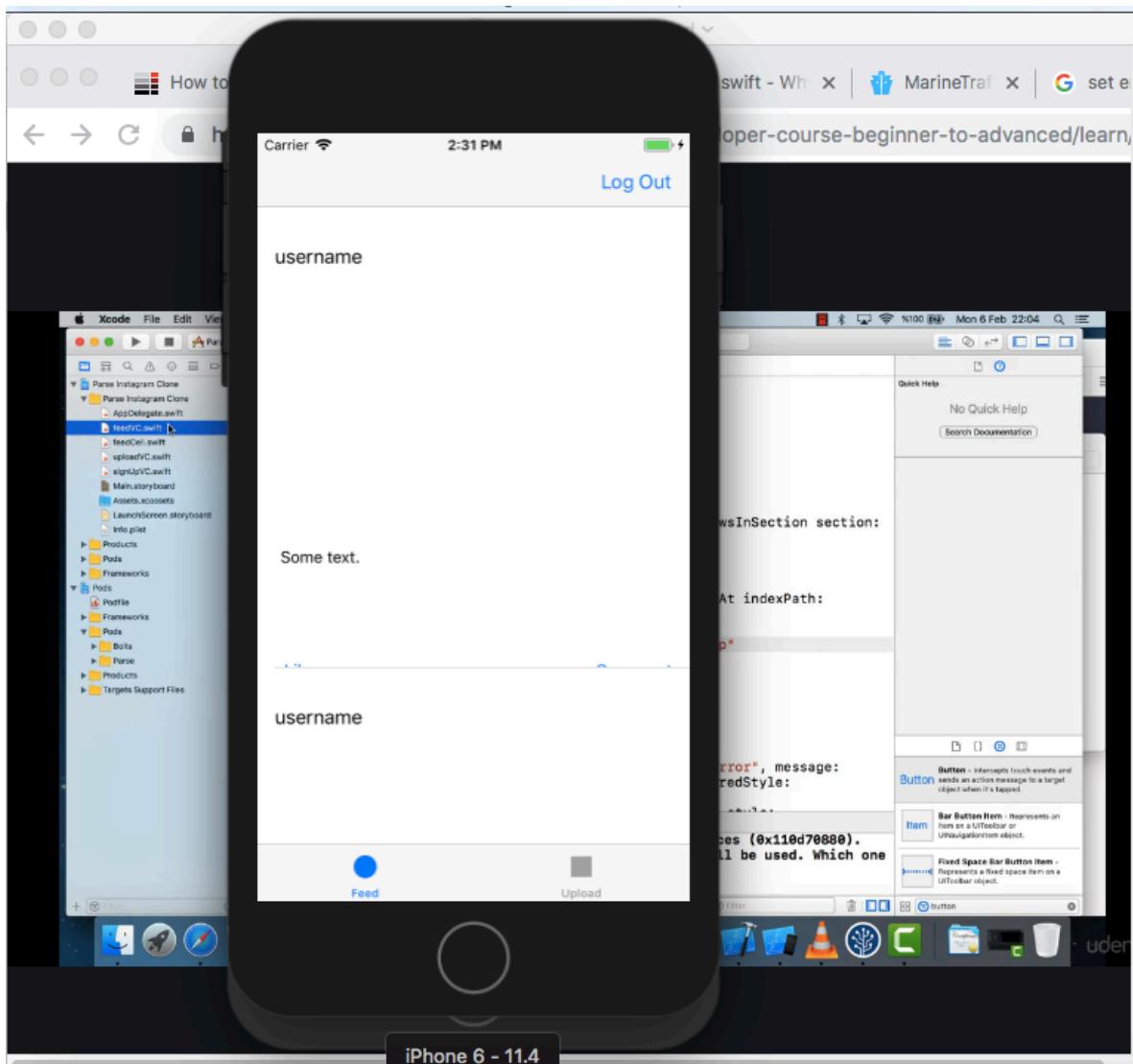
```
let cell = tableView.dequeueReusableCell(withIdentifier: "Cell", for: indexPath)
```

NB. // set **height of prototype cell** programmatically using delegate function

<https://stackoverflow.com/a/46438939>

```
func tableView(_ tableView: UITableView, heightForRowAt indexPath: IndexPath) -> CGFloat {
    return 400
}
```





TODO: fix this in section 15's project

16-152

NB use Structure/hierarchy and select 'Cell' before obtaining Assistant view of code, so as to be able to link properly into feedCell.swift (not feedVC.swift)
 NB ditto for linking, from Structure (rather than storyboard graphic) of control outlets/actions into code

16-153 Upload View Controller laid out (cf 15- for Firebase)

16-154 Image Picking (cf 15-128? for Firebase)

16-155 hiding keyboard (cf 15-128? for Firebase)

16-156 Uploading data from app to Parse
 had a go myself!

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

```

// now upload under way, disable button to prevent impatient click-click
double-uploading
uploadBtn.isEnabled = false

let postObject = PFObject(className: "Posts")      // add a class
// class names visible in Parse dashboard (eg http://34.247.48.250/apps/)
a la tables
postObject["commentText"] = self.commentText.text
postObject["postOwner"] = PFUser.current()!.username!
// UIImageJPEGRepresentation(<#T##image: UIImage##UIImage#>,
<#T##compressionQuality: CGFloat##CGFloat#>)
if let image = UIImageJPEGRepresentation(self.postImage.image!, 0.5)
{
    var parseImageFile = PFFileObject(name: "image.jpg", data: image)
    postObject["postImage"] = parseImageFile
}
else
{
    print("Error processing image file data")
}

let uuid = UUID().uuidString
postObject["postUuid"] = "\(uuid) \(PFUser.current()!.username!)"
postObject.saveInBackground()      // avoid having to write do ... try block
{(success, error) in
    if error != nil
    {
        let alert = UIAlertController(title: "Error",
message:error?.localizedDescription, preferredStyle:
UIAlertControllerStyle.alert)
        let button = UIAlertAction(title: "OK", style:
UIAlertActionStyle.cancel, handler: nil)
        alert.addAction(button)
        self.present(alert, animated: true, completion: nil)
        // NB 1st: the Alert Controller, then 2nd: the button (as an
AlertAction), then 3rd: add button action to the controller, 4th: present the
alert to the user
        print(error?.localizedDescription)
    }
    else
    {
        // assuming successful upload, reset controls of upload view
        self.postImage.image = UIImage(named: "select-picture.png")
        self.commentText.text = ""
        self.tabBarController?.selectedIndex = 0 // 0: viz the initial tab
view controller, to the first ViewController ie feedVC - redirect thither
    }
}

```

```
}
```

	updatedAt	ACL	commentText	postImage	postOwner	postUuid
...	13 Apr 2019 at 17:...	Public Read + Writ...	A waterfall, shrub...	e.jpg	cart man	8C237D91-AE7B-46B1...
...	13 Apr 2019 at 17:...	Public Read + Writ...	Yellow leaf among ...	(undefined)	(undefined)	(undefined)

16-157 Displaying feed results from Parse server database (cf 16- for Firebase)

```
func getDataFromServer()
{
    let query = PFQuery(className: "Posts")
    query.addDescendingOrder("createdAt")           // sort by date, oldest-first
    query.findObjectsInBackground { (posts, error) in
        if error != nil
        {
            let alert = UIAlertController(title: "Error",
message:error?.localizedDescription, preferredStyle:
UIAlertControllerStyle.alert)
            let button = UIAlertAction(title: "OK", style: UIAlertActionStyle.cancel,
handler: nil)
            alert.addAction(button)
            self.present(alert, animated: true, completion: nil)
            // NB 1st: the Alert Controller, then 2nd: the button (as an
AlertAction), then 3rd: add button action to the controller, 4th: present the
alert to the user
            print(error?.localizedDescription)
        }
        else
        {
            print(posts)    // diagnostic
            self.postImageArray.removeAll(keepingCapacity: false)
        }
    }
}
```

```

        self.commentTextArray.removeAll(keepingCapacity: false)
        self.postOwnerArray.removeAll(keepingCapacity: false)
        self.postUuidArray.removeAll(keepingCapacity: false)

        for post in posts!
        {
            self.postImageArray.append(post.object(forKey: "postImage") as!
PFFFileObject)
            self.commentTextArray.append(post.object(forKey:
"commentText") as! String)
            self.postOwnerArray.append(post.object(forKey: "postOwner") as!
String)
            self.postUuidArray.append(post.object(forKey: "postUuid") as!
String)
        }
    }
    self.tableView.reloadData()
}
}
}

```

16-158 NotificationCenter for ensuring sensible navigation when having uploaded

(feedVC tableview 'Selection' from 'single selection' to 'no selection' in Attribute Inspector - no more 'choose')

```

// NB @objc to avoid (at #selector) error: Argument of '#selector' refers to
instance method 'getDataFromServer()' that is not exposed to Objective-C
/// function to get feed data from server (Parse server db in this case)
@objc func getDataFromServer()
{
    let query = PFQuery(className: "Posts")
    query.addDescendingOrder("createdAt")      // sort by date, oldest-first
    query.findObjectsInBackground {(posts, error) in
        if error != nil
        {
            let alert = UIAlertController(title: "Error",
message:error?.localizedDescription, preferredStyle:
UIAlertControllerStyle.alert)
            let button = UIAlertAction(title: "OK", style: UIAlertActionStyle.cancel,
handler: nil)
            alert.addAction(button)
            self.present(alert, animated: true, completion: nil)
            // NB 1st: the Alert Controller, then 2nd: the button (as an
AlertAction), then 3rd: add button action to the controller, 4th: present the
        }
    }
}

```

```

alert to the user
    print(error?.localizedDescription)
}
else
{
    print(posts) // diagnostic

    self.postImageArray.removeAll(keepingCapacity: false)
    self.commentTextArray.removeAll(keepingCapacity: false)
    self.postOwnerArray.removeAll(keepingCapacity: false)
    self.postUuidArray.removeAll(keepingCapacity: false)

    for post in posts!
    {
        // TODO will throw errors if blank fields in db
        self.postImageArray.append(post.object(forKey: "postImage") as!
PFFFileObject)
        self.commentTextArray.append(post.object(forKey:
"commentText") as! String)
        self.postOwnerArray.append(post.object(forKey: "postOwner") as!
String)
        self.postUuidArray.append(post.object(forKey: "postUuid") as!
String)
    }
    self.tableView.reloadData()
}
}
}

```

16-159 skeleton for 'likes and comments' social media style functionality

new class / table for 'Likes' - who liked what - use for uuid for user

code within feedCell - ie within Prototype Cells in tableview in feedVC

comment text not recorded in section code - left as TODO

The screenshot shows the Parse Dashboard interface. On the left, there's a sidebar titled "Core" with a tree view of classes: Role (0), Session (1), User (2), Animals (2), **Comments** (2), Likes (1), Posts (3), Webhooks, Jobs, Logs, Config, and API Console. At the bottom of the sidebar is a "Push" button. The main area is titled "Comments" and shows a table with two rows. The columns are: objectId (String), createdAt (Date), updatedAt (Date), ACL (ACL), to (String), and from (String). The first row has objectId "dcx6JERHzU", createdAt "13 Apr 2019 at 19:29:14", updatedAt "13 Apr 2019 at 19:29:14", ACL "Public Read + Write", to "086F7C8A-0F11-402C...", and from "Kenny". The second row has objectId "xr2AiMmb22", createdAt "13 Apr 2019 at 18:54:14", updatedAt "13 Apr 2019 at 18:54:14", ACL "Public Read + Write", to "086F7C8A-0F11-402C...", and from "Kenny". There are buttons for "Add Row", "Refresh", "Filter", and "Security".

	objectId	createdAt	updatedAt	ACL	to	from
1	dcx6JERHzU	13 Apr 2019 at 19:29:14	13 Apr 2019 at 19:29:14	Public Read + Write	086F7C8A-0F11-402C...	Kenny
2	xr2AiMmb22	13 Apr 2019 at 18:54:14	13 Apr 2019 at 18:54:14	Public Read + Write	086F7C8A-0F11-402C...	Kenny

```

@IBAction func likeBtnClicked(_ sender: Any) {
    let likeObject = PFObject(className: "Likes")
    likeObject["from"] = PFUser.current()!.username!      // person using at
mo is one liking
    likeObject["to"] = postuuidLabel.text

    likeObject.saveInBackground {(success, error) in
        if error != nil
        {
            let alert = UIAlertController(title: "Error",
message:error?.localizedDescription, preferredStyle:
UIAlertControllerStyle.alert)
                let button = UIAlertAction(title: "OK", style: UIAlertActionStyle.cancel,
handler: nil)
                    alert.addAction(button)
                    UIApplication.shared.keyWindow?.rootViewController?.present(alert,
animated: true, completion: nil)
                    // self.present(alert, animated: true, completion: nil)
                    // NB 1st: the Alert Controller, then 2nd: the button (as an
AlertAction), then 3rd: add button action to the controller, 4th: present the
alert to the user
                    // NB in this case as using TableViewCell, cannot present Alert so use
underlying ViewController instead
                    print(error?.localizedDescription)
        }
        else
        {
            print("Like has been saved to server")
        }
    }
}

```

```

        }
    }
}

@IBAction func commentBtnClicked(_ sender: Any) {
    // NB TODO text of comment not actually included here - needs another
control?
    let commentObject = PFObject(className: "Comments")
    commentObject["from"] = PFUser.current()!.username!      // person using
at mo is one commenting
    commentObject["to"] = postuuidLabel.text

    commentObject.saveInBackground {(success, error) in
        if error != nil
        {
            let alert = UIAlertController(title: "Error",
message:error?.localizedDescription, preferredStyle:
UIAlertControllerStyle.alert)
                let button = UIAlertAction(title: "OK", style: UIAlertActionStyle.cancel,
handler: nil)
                alert.addAction(button)
                UIApplication.shared.keyWindow?.rootViewController?.present(alert,
animated: true, completion: nil)
                // self.present(alert, animated: true, completion: nil)
                // NB 1st: the Alert Controller, then 2nd: the button (as an
UIAlertAction), then 3rd: add button action to the controller, 4th: present the
alert to the user
                // NB in this case as using TableViewCell, cannot present Alert so use
underlying ViewController instead
                print(error?.localizedDescription)
        }
        else
        {
            print("Comment has been saved to server")
        }
    }
}

```

logins: 'cart man' and 'kenny' both as 123456

fiddling with leading layout constraints to yield acceptable results is getting a bit easier

NB TODO: back4app.com as similar to Parse

17-161 intro to push notifications

17-162

apple developer - don't recall it costing me anything ??? TODO check this

The screenshot shows the 'Apple Developer' account dashboard. On the left, there's a sidebar with links like 'Welcome', 'Documentation', 'Downloads', 'Forums', 'Bug Reporter', 'Account Help', and 'Contact Us'. The main area is titled 'Getting Started' with the sub-section 'Download Tools'. It features two cards: 'Download Tools' (with an icon of a wrench) and 'Build Your First App' (with an icon of a gear). Below these is a button for 'Join the Apple Developer Program' with an icon of a person. A note below the button says: 'Membership in the Apple Developer Program includes everything you need to develop, distribute, and manage your apps on the App Store. You'll also gain access to beta software, advanced app capabilities, beta testing tools, and app analytics.'

XCode : Project (top left)/Target/Capabilities

eg App Groups, Apple Pay, Associated Domains, **Background Modes**, Class Kit, Data Protection, Game Centre, Health Kit, Home Kit, Hotspot Configuration, iCloud, In-App Purchase, Inter-App Audio, **Keychain Sharing**, Maps, Multipath, Near Field Communication Tag Reading, Network Extensions, Network Extensions, Personal VPN, **Push Notifications**, Siri, Wallet, Wireless Accessory Configuration.

for push: 1st turn on (enable) **Keychain Sharing**, 2nd turn on **Push Notifications**, 3rd turn on **Background Modes** and therein tick checkbox for **Remote Notification**.

NB For enabling Keychain Sharing, had to create a Team - as: Peter Jenkin(Personal Team) role:user - could then select this team in dialogue.

pending device for provisioning..... TODO TODO TODO ?

connecting to actual physical device (i bought an iphone 6)
warning from XCode

This iPhone 6 is running iOS 12.2 (16E227), which may not be supported by this version of Xcode.

try <https://stackoverflow.com/a/55576391/11365317>

NB apps directories from root /

from [https://github.com/iGhibli/iOS-DeviceSupport/blob/master/DeviceSupport/12.2%20\(16E226\).zip](https://github.com/iGhibli/iOS-DeviceSupport/blob/master/DeviceSupport/12.2%20(16E226).zip)

in /Applications/Xcode.app/Contents/Developer/Platforms/iPhoneOS.platform/DeviceSupport folder (with a dozen or so others from 8.0 to 11.4) named (renamed) e.g. 12.2 (16E227) containing 2 files:

DeveloperDiskImage.dmg
DeveloperDiskImage.dmg.signature

Did so,
then ...

XCode seemingly waiting (hanging?) for 5 minutes on message, "**preparing debugger support for iphone**"

<https://stackoverflow.com/a/47448911/11365317>

Did something or other like so,
then ...

XCode warning: **Could not launch "Parse Foursquare Clone"**

Verify the Developer App certificate for your account is trusted on your device.
Open Settings on Peter Jenkin's iPhone and navigate to General -> Device Management, then select your Developer App certificate to trust it.

Did so - on iPhone - app started ok! :)

NB iPhone settings/privacy/location services - turn on to use phone location

(18/4/19 signed up for 1 year of Apple Developer Program membership #79 to enable provisioning of iPhone)

After this had to delete and re-add my account in XCode to see Push notifications in Capabilities, as per <https://stackoverflow.com/a/50474745/11365317>

17-162 (continued) ... went ok

17-163 [onesignal.com](https://documentation.onesignal.com/docs/generate-an-ios-push-certificate) - sign up for limited free login if not yet done - go through steps 1-4 & i-xxvi - 'Add App' eg 'Insta Clone Parse' - (deprecated: from console, app's Settings) - (1) choice of platform: iOS, Android, Google Web Push, Amazon Fire, Windows, MacOS -

(2) Production Certificate .p12 file requested -

<https://documentation.onesignal.com/docs/generate-an-ios-push-certificate>

(i) open Keychain Access on Mac, (ii). Keychain Access/Certificate Assistant/Request Certificate from Certificate Authority... (iii). give email address as user and CA Certificate Authority - choose 'Saved to disk' (iv) Save file (without tags is ok) (v)file

CertificateSigningRequest.certSigningRequest probably made on Desktop (Finder will show) (vi) go to [developer.apple.com](https://developer.apple.com/account/ios/certificate/) and go to page **Certificates IDs & Profiles** <https://developer.apple.com/account/ios/certificate/> (vii) click on link '**App IDs**' and locate the app in question from the list e.g. XC com *peterjenkin Parse-Insta-Clone* (viii) hopefully under the expansion of this app,

'Services' will be marked *Enabled*, *Disabled* or *Configurable* as they were in the Capabilities tab of the Project in XCode - 'Push Notifications' may be 'Configurable'. Click on the 'Edit' button if seen at the bottom. (ix) from the app's services 'Edit' page, scroll down to 'Push Notifications' - note that this is configurable (x) (right-click to open in another tab), in Apple Developer links click on '**Certificates-All**' page, to open in another tab (xi) from Certificates-All, click at the page's top-right '+' button to create a new certificate (xii) "What type of certificate do you need?" is asked - go down to **Production** (not Development) (xiii) from the list of certificate types, choose "**Apple Push Notification service SSL (Sandbox & Production)**" click on 'Continue' (xiv) for "Which App Id would you like to use?" go through the list (as per step vii) and click 'Continue' (xv) a page re Certificate Signing Request CSR file - done in step v - click 'Continue'(xvi) choose

CertificateSigningRequest.certSigningRequest file from step v - click 'Continue' (xvii) if all's well, "**Your certificate is ready**" page will have a confirmation and can click on '**Download**' button for a .cer file eg **aps.cer** (.cer file for use in Keychain Access, saved on computer). (xx) on the computer, in Finder, double-click on the downloaded **xyz.cer/aps.cer** file (xxi) Keychain Store will open asking "**Do you want to add the certificate(s) from the file to a keychain?**" with keychain options set to "login" (xxii) in Keychain Store, on the left click on '**My Certificates**'(makes easier to find own certificates like this one) and locate the new certificate (xxiii) click to expand the new certificate in Keychain Store until the signer's name (eg "Peter Jenkin") is visible in a sub-row - right click on this and select Export "egPeter Jenkin" (xxiv) a dialogue will appear with a suggested filename of "**Certificates.p12**" - (other possible formats may include .pem, .cer or .p7b) - save to a location on the computer (xxv) assign password to certificate (or not) (xxvi) back in onesignal Add App, where

Production Certificate .p12 file was requested - this **Certificates.p12** file can now be uploaded. (supply password if set in step xxv

(NB This is the 'hard way' - one signal has an automated process - XCode can also perform some steps)

TODO see how other processes interact with steps i-xxvi

(2 - continued) in onesignal Add App setup, options *Upload optional sandbox certificate* (Apple's new production certificates work with both sandbox and production environments.) and *Enable iOS 12 Direct To History*.

(3) Select SDK - from eg Native iOS, Corona, Phonegap/Cordova/Ionic, Xamarin, React Native, Unity - select Native iOS **(4)** as below:

Apple iOS SDK Installation

1. Install the SDK

Read the documentation to learn how to install the SDK. Using CocoaPods, a dependency manager, provides the swiftest installation. The longer alternative is to manually download our SDK and add it as a dependency.

Your App ID: c474e572-9466-4f70-9584-565d00e72385

2. After Installing the SDK (Optional Testing)

Click **Check Subscribed Users** to verify you have at least one subscribed user.

The screenshot shows the 'Certificates, Identifiers & Profiles' section of the Apple Developer Account. The left sidebar is set to 'iOS, tvOS, watchOS'. Under 'Identifiers', 'App IDs' is selected. The main area displays a table titled '7 App IDs Total' with columns 'Name', 'ID', 'Status', and 'Enabled'. The table lists various services: Apple Pay Payment Processing, Associated Domains, AutoFill Credential Provider, ClassKit, Data Protection, Game Center, HealthKit, HomeKit, Hotspot, iCloud, and In-App Purchase. Most services are listed as 'Disabled' in the status column, while Game Center and In-App Purchase are marked as 'Enabled'.

Name	ID	Status	Enabled
Apple Pay Payment Processing		Disabled	Disabled
Associated Domains		Disabled	Disabled
AutoFill Credential Provider		Disabled	Disabled
ClassKit		Disabled	Disabled
Data Protection		Disabled	Disabled
Game Center		Enabled	Enabled
HealthKit		Disabled	Disabled
HomeKit		Disabled	Disabled
Hotspot		Disabled	Disabled
iCloud		Disabled	Disabled
In-App Purchase		Enabled	Enabled

The screenshot shows the same 'Certificates, Identifiers & Profiles' section, but the left sidebar is set to 'Identifiers'. The left sidebar includes sections for 'Website Push IDs', 'iCloud Containers', 'App Groups', 'Merchant IDs', 'Music IDs', 'Maps IDs', 'Devices' (listing All, Apple TV, Apple Watch, iPad, iPhone, iPod Touch), and 'Provisioning Profiles' (listing All, Development, Distribution). The main area displays a table with columns 'Identifier', 'Status', and 'Enabled'. The table lists various identifiers: HealthKit, HomeKit, Hotspot, iCloud, In-App Purchase, Inter-App Audio, Multipath, Network Extensions, NFC Tag Reading, Personal VPN, Push Notifications, SiriKit, Wallet, and Wireless Accessory Configuration. Most identifiers are marked as 'Disabled' in the status column, while In-App Purchase and Push Notifications are marked as 'Configurable'.

Identifier	Status	Enabled
HealthKit	Disabled	Disabled
HomeKit	Disabled	Disabled
Hotspot	Disabled	Disabled
iCloud	Disabled	Disabled
In-App Purchase	Enabled	Enabled
Inter-App Audio	Disabled	Disabled
Multipath	Disabled	Disabled
Network Extensions	Disabled	Disabled
NFC Tag Reading	Disabled	Disabled
Personal VPN	Disabled	Disabled
Push Notifications	Configurable	Configurable
SiriKit	Disabled	Disabled
Wallet	Disabled	Disabled
Wireless Accessory Configuration	Disabled	Disabled

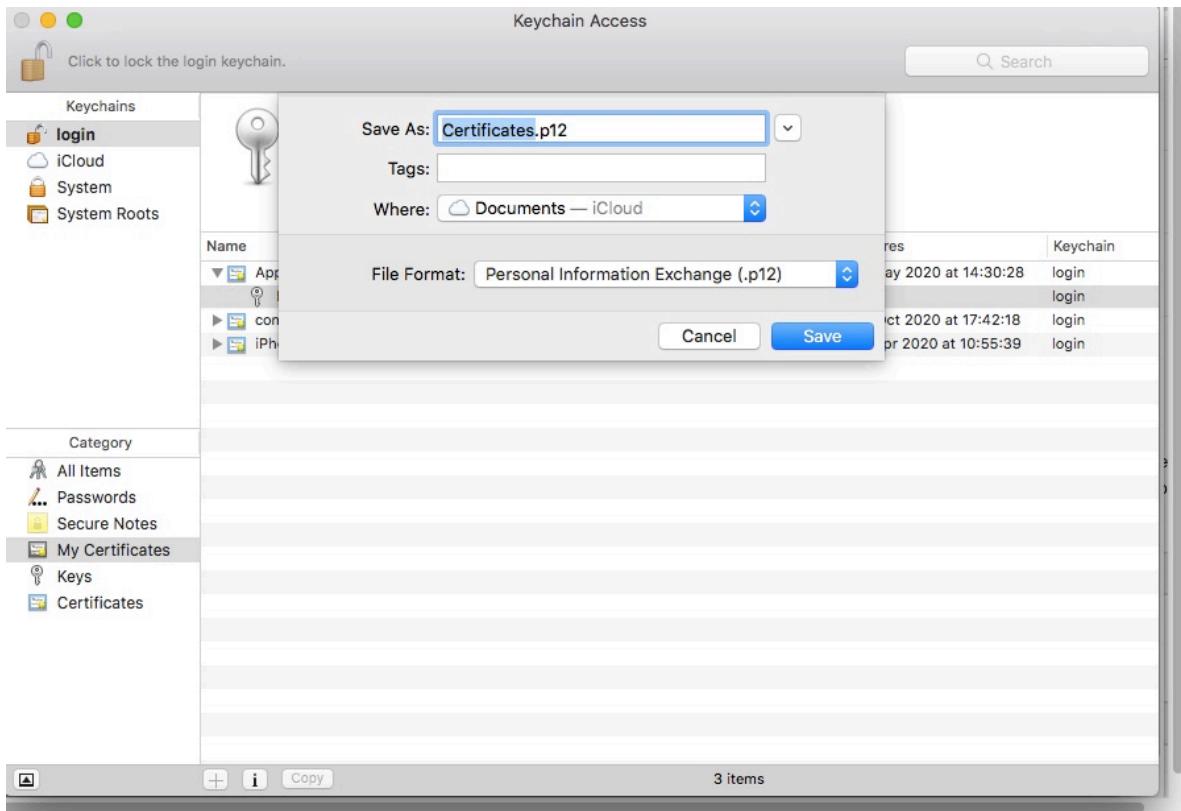
Screenshot of the Apple Developer Account settings page for Push Notifications:

The "Push Notifications" option is selected (Configurable). A note states: "To configure push notifications for this iOS App ID, a Client SSL Certificate that allows your notification server to connect to the Apple Push Notification Service is required. Each iOS App ID requires its own Client SSL Certificate. Manage and generate your certificates below." Below this, there are two sections: "Development SSL Certificate" and "Production SSL Certificate", each with a "Create Certificate" button.

Screenshot of Keychain Access showing certificates:

The "My Certificates" category is selected. A table lists certificates:

Name	Kind	Expires	Keychain
Apple Push Services: com.peternjenkin.Parse-Insta-Clone	certificate	17 May 2020 at 14:30:28	login
Peter Jenkins	private key	--	login
com.apple.idms.appleid.prd....4a3575784c4f6f65795b773d3d	certificate	30 Oct 2020 at 17:42:18	login
iPhone Developer: peternjenkin@icloud.com (3GGLYG3SLZ)	certificate	14 Apr 2020 at 10:55:39	login



17-164

in <https://documentation.onesignal.com/docs/ios-sdk-setup>
are steps for setting up in XCode, different from those in the 'Add App' web dialogue (step 4 in 17-163)

Your App ID: c474e572-9466-4f70-9584-565d00e72385

XCode/ File/New/**Target**.... choose (for **iOS** not macOS &c) from many options (eg Content Blocker, thumbnail, share, Unit Testing Bundle, iMessage)

'Notification Service Extension' - name this

OneSignalNotificationServiceExtension exactly

NB when prompted **Activate "OneSignalNotificationServiceExtension"**

scheme? select '**Cancel**', (*not* 'Activate') , to cancel activation.

Extension should appear within workspace as

"OneSignalNotificationServiceExtension"

copy Swift code from <https://documentation.onesignal.com/docs/ios-sdk-setup>
as per:

```
import UserNotifications
```

```
import OneSignal
```

```
class NotificationService: UNNotificationServiceExtension {
```

```
    var contentHandler: ((UNNotificationContent) -> Void)?
```

```
    var receivedRequest: UNNotificationRequest!
```

```
    var bestAttemptContent: UNMutableNotificationContent?
```

```

override func didReceive(_ request: UNNotificationRequest,
withContentHandler contentHandler: @escaping (UNNotificationContent) ->
Void) {
    self.receivedRequest = request;
    self.contentHandler = contentHandler
    bestAttemptContent = (request.content.mutableCopy() as?
UNMutableNotificationContent)

    if let bestAttemptContent = bestAttemptContent {

OneSignal.didReceiveNotificationExtensionRequest(self.receivedRequest, with:
self.bestAttemptContent)
        contentHandler(bestAttemptContent)
    }
}

override func serviceExtensionTimeWillExpire() {
    // Called just before the extension will be terminated by the system.
    // Use this as an opportunity to deliver your "best attempt" at modified
content, otherwise the original push payload will be used.
    if let contentHandler = contentHandler, let bestAttemptContent =
bestAttemptContent {

OneSignal.serviceExtensionTimeWillExpireRequest(self.receivedRequest, with:
self.bestAttemptContent)
        contentHandler(bestAttemptContent)
    }
}

}

```

and overwrite contents of **OneSignalNotificationServiceExtension/ NotificationService.swift** with this.

Add into Podfile ('pod init' if needed) **only some part** of the docs' recommended script, with **use_frameworks!** added as per:

```

target 'OneSignalNotificationServiceExtension' do
  use_frameworks!
  pod 'OneSignal', '>= 2.6.2', '< 3.0'
end

```

(OneSignal pod installed to workspace)

In **AppDelegate.swift**:

import OneSignal

and in `func application(_ application: UIApplication, didFinishLaunchingWithOptions...)`

```
let onesignalInitSettings = [kOSSettingsKeyAutoPrompt: false]

// Replace 'YOUR_APP_ID' with your OneSignal App ID.
OneSignal.initWithLaunchOptions(launchOptions,
appId: "YOUR_ONESIGNAL_APP_ID",
handleNotificationAction: nil,
settings: onesignalInitSettings)

OneSignal.inFocusDisplayType = OSNotificationDisplayType.notification;

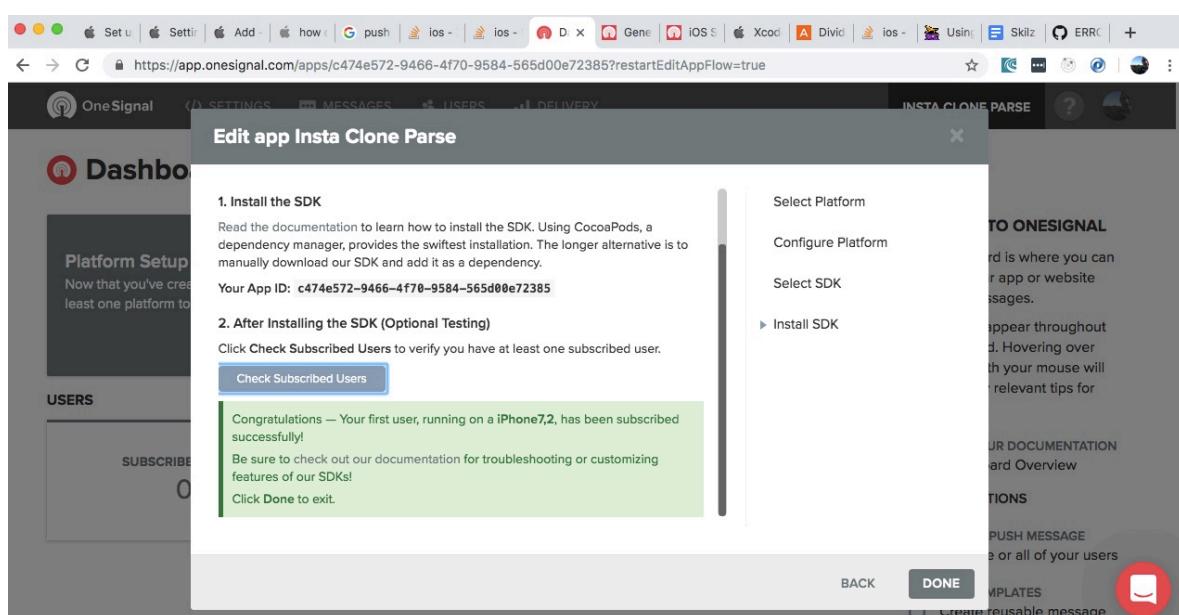
// Recommend moving the below line to prompt for push after informing the
user about
// how your app will use them.
OneSignal.promptForPushNotifications(userResponse: { accepted in
print("User accepted notifications: \(accepted)")
})

// in this case from the OneSignal web page Add App setup (17-163 step 4)
Your App ID: c474e572-9466-4f70-9584-565d00e72385
```

and run on phone...

– despite:

Signal strength query returned error: Error Domain=NSPOSIXErrorDomain Code=13 "Permission denied", descriptor: <CTServiceDescriptor 0x280819740,
:-)



subscribed within phone, ready to send/receive push notifications

NB a target is a build for a certain condition

<https://stackoverflow.com/questions/20637435/xcode-what-is-a-target-and-scheme-in-plain-language>

<https://www.appcoda.com/xcode-targets/>

<https://developer.apple.com/library/archive/featuredarticles/XcodeConcepts/Concept-Targets.html>

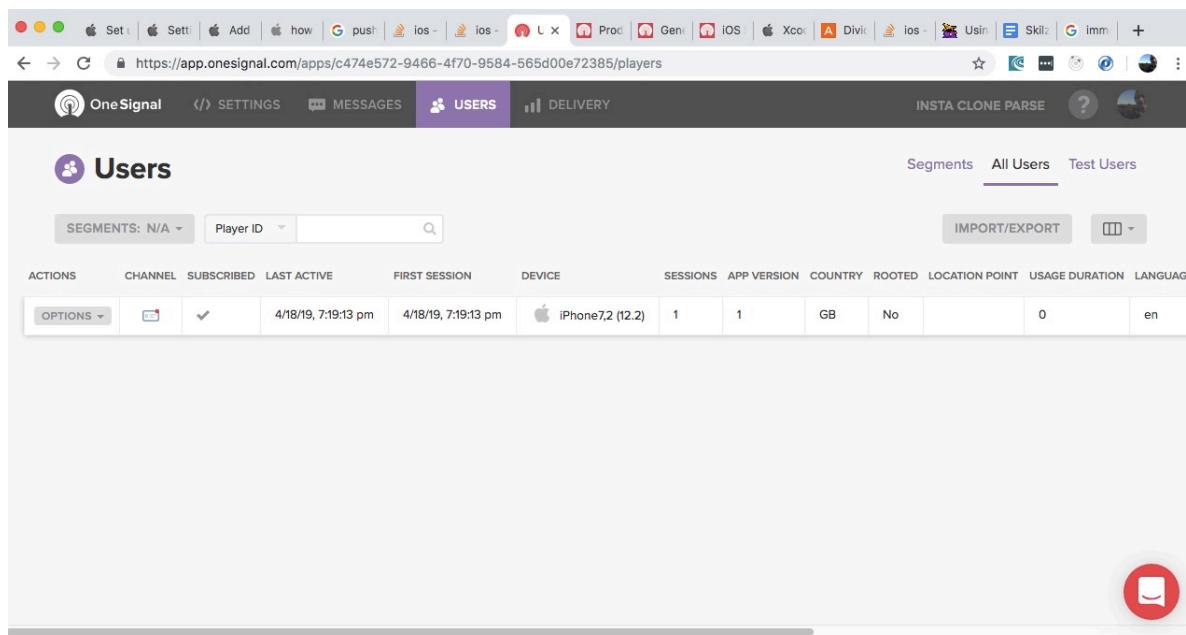
https://pewpewthespells.com/blog/using_xcode_targets.html

TODO: mug up

TODO: does this merge into CI for iOS?

17-165

ignore Segments for now - go to Users (top) / All Users (tab)



ACTIONS	CHANNEL	SUBSCRIBED	LAST ACTIVE	FIRST SESSION	DEVICE	SESSIONS	APP VERSION	COUNTRY	ROOTED	LOCATION POINT	USAGE DURATION	LANGUAGE
OPTIONS		✓	4/18/19, 7:19:13 pm	4/18/19, 7:19:13 pm	iPhone7.2 (12.2)	1	1	GB	No	0	en	

Manual push notification - from OneSignal logged-in web console page, go to app then under 'Common Actions' on right, click on '**Send a Push Message**'. On page New Message, 'Send to Subscribed Users' (other options 'Send to Particular Segment' 'Send to Test Device'). Fill in Title, Subtitle and Message. **Badges** are the number appearing, to show a backlog of whatever (messages &c).

Notification on OS basis eg for iOS.

Sending can be timed/scheduled.

Notification can include: launch URL, collapse ID, time-to-live and action buttons

Realtime notification data

17-166 <https://documentation.onesignal.com/docs/ios-native-sdk> <https://documentation.onesignal.com/docs/ios-native-sdk#section--postnotification>
Automated Push notifications - sending triggered by code in feedVC:

```

override func viewDidLoad() {
    super.viewDidLoad()
    // Do any additional setup after loading the view, typically from a nib.

    // set up table view
    tableView.delegate = self
    tableView.dataSource = self

    // get feed data from server
    getDataFromServer()

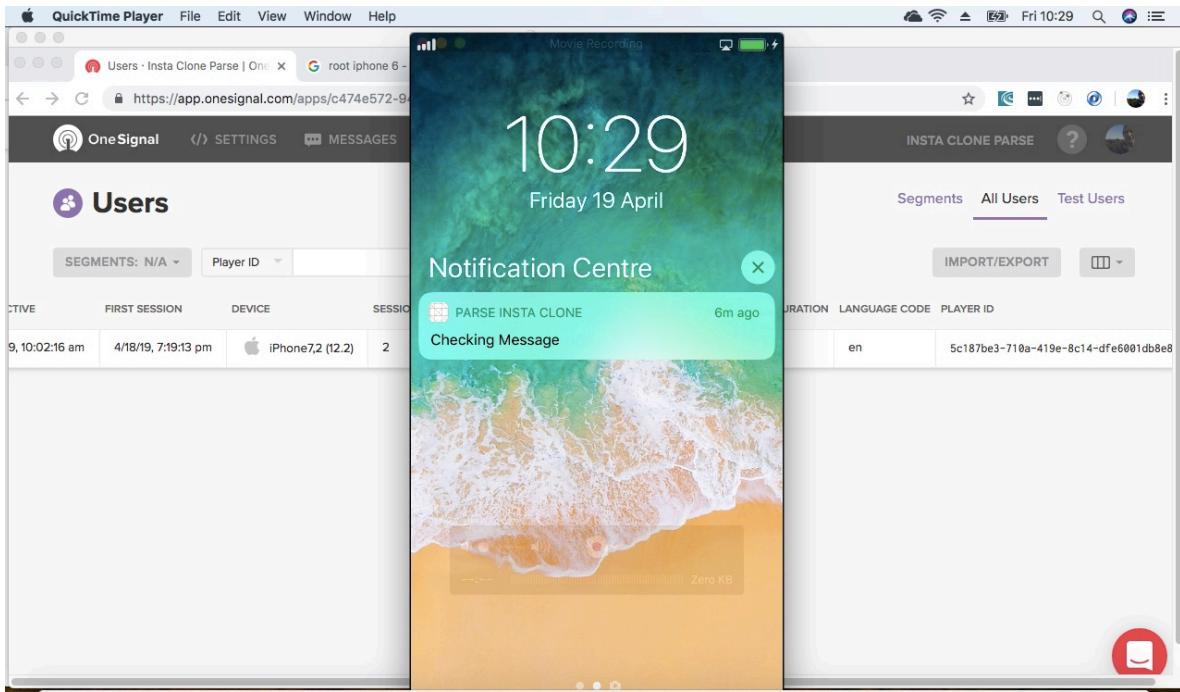
    // Push notification check
    OneSignal.postNotification(["contents": ["en": "Checking Message"],
    "include_player_ids": ["5c187be3-710a-419e-8c14-dfe6001db8e8"]])
    // pasted from https://documentation.onesignal.com/docs/ios-native-
    sdk#section--postnotification-
}

```

Find playerID (effectively a user/device id) listed under **Users** (top menu), **All Users** (right side menu) - paste into code the value listed under **PLAYER ID** column.

ACTIVE	FIRST SESSION	DEVICE	SESSIONS	APP VERSION	COUNTRY	ROOTED	LOCATION POINT	USAGE DURATION	LANGUAGE CODE	PLAYER ID
9, 10:02:16 am	4/18/19, 7:19:13 pm	iPhone7,2 (12.2)	2	1	GB	No	0	0	en	5c187be3-710a-419e-8c14-dfe6001db8e8

- message seen on app start up



NB in Quicktime, New video recording and just to right of red record button, press downward v arrow and select 'iPhone screen'

17-167 Programatically obtain Player IDs

```

override func viewDidLoad() {
    super.viewDidLoad()
    // Do any additional setup after loading the view, typically from a nib.

    // set up table view
    tableView.delegate = self
    tableView.dataSource = self

    // get feed data from server
    getDataFromServer()

    // find whether user yet subscribed (to OneSignal push notifications)
    let status: OSPermissionSubscriptionState =
    OneSignal.getPermissionSubscriptionState()

    let userID = status.subscriptionStatus.userId      // userID playerID

    let pushToken = status.subscriptionStatus.pushToken

    print("userID: \(_userID)")                      // diagnostic
    print("pushToken: \(_pushToken)")                 // diagnostic
/*
}

```

```

userID: Optional("5c187be3-710a-419e-8c14-dfe6001db8e8")
pushToken:
Optional("7fa6ee0ec19c1d10441fb5b898583e19e7dbd73ce76548bcfd65
cab7b805c04f")

ie userID == playerID
... so ...
override func viewDidLoad() {
    super.viewDidLoad()
    // Do any additional setup after loading the view, typically from a nib.

    // set up table view
    tableView.delegate = self
    tableView.dataSource = self

    // get feed data from server
    getDataFromServer()

    // find whether user yet subscribed (to OneSignal push notifications)
    let status: OSPermissionSubscriptionState =
OneSignal.getPermissionSubscriptionState()

let userID = status.subscriptionStatus.userId      // userID playerID

// let pushToken = status.subscriptionStatus.pushToken

print("userID: \(userID)")           // diagnostic
// print("pushToken: \(pushToken)")     // diagnostic

if userID != nil
{
    // use Parse user
    let user = PFUser.current()!.username!
    let object = PFObject(className: "PlayerID")
    object["username"] = user
    object["playerID"] = userID
    object.saveEventually()      // NB not saveInBackground - saves error
messages in case no connection, e.g. - save whenever chance may arise, and
userID will stay constant after registration once with OneSignal
}
/*
    // Push notification check - only sending to 1 phone (defined by PlayerID)
    OneSignal.postNotification(["contents": ["en": "Checking Message"],
"include_player_ids": ["5c187be3-710a-419e-8c14-dfe6001db8e8"]])
    // pasted from https://documentation.onesignal.com/docs/ios-native-
sdk#section--postnotification-

```

```
*/  
}
```

will make a record of this phone/user/player id in the Parse server db

The screenshot shows the Parse Dashboard interface. On the left, there's a sidebar titled "Core" with sections like "Browser", "Create a Class", "Webhooks", "Jobs", "Logs", "Config", and "API Console". Below that is a "Push" button. At the bottom of the sidebar are links to "Open Source Hub", "GitHub", and "Docs". The main area is titled "CLASS" and shows the "PlayerID" class. It indicates "1 object • Public Read and Write enabled". There's a table with columns: objectId, createdAt, updatedAt, ACL, username, and playerID. A single row is listed: objectId is "jdGXDeZrYp", createdAt and updatedAt are "19 Apr 2019 at 10:25:40 UTC", ACL is "Public Read + Write", username is "James", and playerID is "5c187be3-710a-419e...". There are buttons for "+ Add Row", "@ Refresh", "Filter", "Security", and "Edit".

17-168 Sending post notifications programmatically to a specific user (using db of OneSignal registered playerIDs)

in app, use the post's author's label text value to get name to query the Parse server db's list of PlayerIDs. Send a notification if found:

```
@IBAction func likeBtnClicked(_ sender: Any) {  
    let likeObject = PFObject(className: "Likes")  
    likeObject["from"] = PFUser.current()!.username! // person using at  
mo is one liking  
    likeObject["to"] = postuuidLabel.text  
  
    likeObject.saveInBackground {(success, error) in  
        if error != nil  
        {  
            let alert = UIAlertController(title: "Error",  
message:error?.localizedDescription, preferredStyle:  
UIAlertControllerStyle.alert)  
            let button = UIAlertAction(title: "OK", style: UIAlertActionStyle.cancel,  
handler: nil)  
            alert.addAction(button)  
            UIApplication.shared.keyWindow?.rootViewController?.present(alert,  
animated: true, completion: nil)  
            // self.present(alert, animated: true, completion: nil)  
            // NB 1st: the Alert Controller, then 2nd: the button (as an  
AlertAction), then 3rd: add button action to the controller, 4th: present the  
alert to the user
```

```

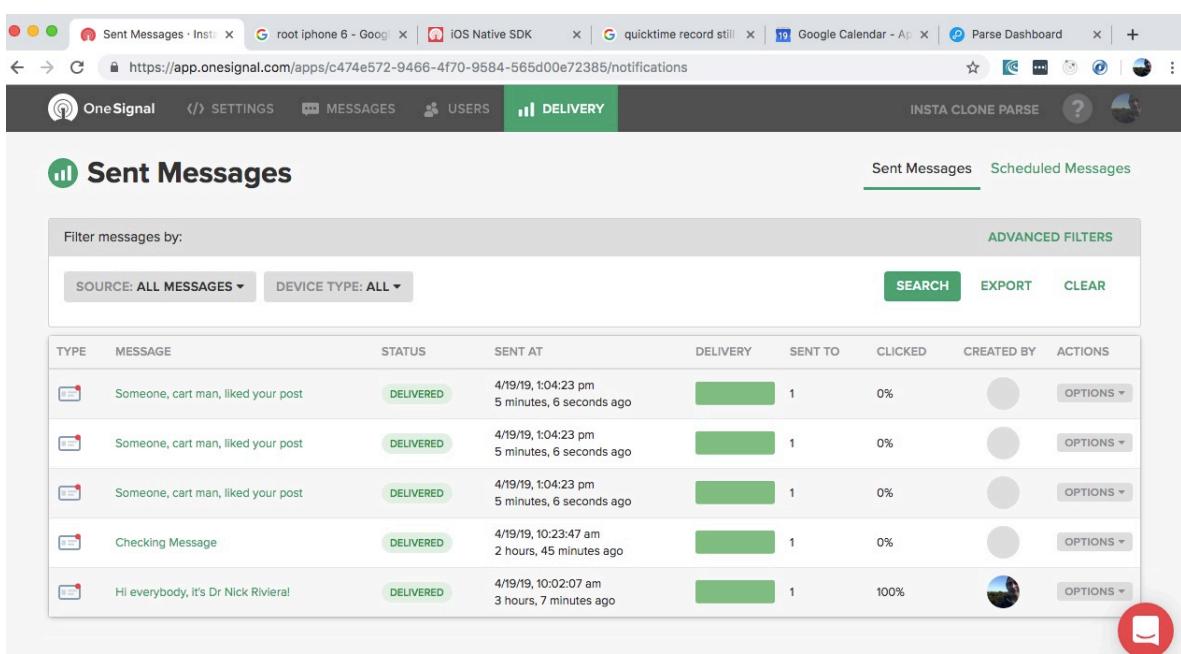
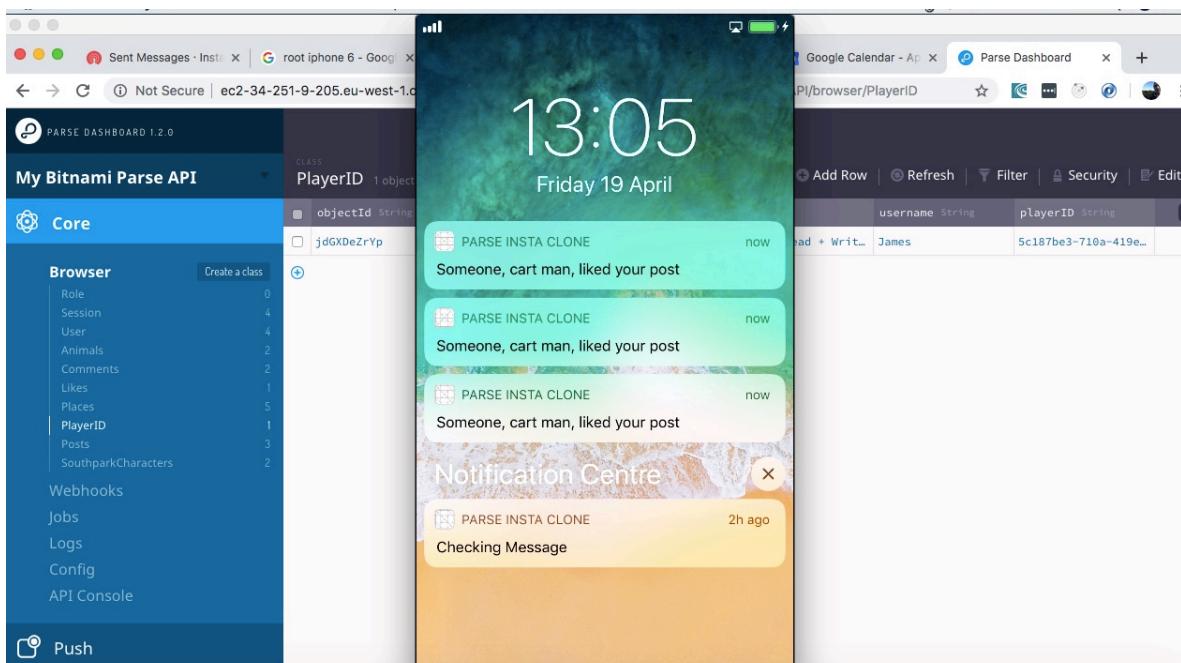
    // NB in this case as using TableViewCell, cannot present Alert so use
underlying ViewController instead
    print(error?.localizedDescription)
}
else
{
    print("Like has been saved to server")

    // send push notification to poster of this post to let them know that
their post was liked by this user
    let query = PFQuery(className: "PlayerID")
        query.whereKey("username", equalTo: self.usernameLabel.text!) // get poster's name
        // NB to work, poster must have been registered via the app with
OneSignal
        query.findObjectsInBackground(block: {(posters, error) in
            if error != nil
            {
                let alert = UIAlertController(title: "Error",
message:error?.localizedDescription, preferredStyle:
UIAlertControllerStyle.alert)
                    let button = UIAlertAction(title: "OK", style:
UIAlertActionStyle.cancel, handler: nil)
                    alert.addAction(button)

UIApplication.shared.keyWindow?.rootViewController?.present(alert, animated:
true, completion: nil)
                    // self.present(alert, animated: true, completion: nil)
                    // NB 1st: the Alert Controller, then 2nd: the button (as an
UIAlertAction), then 3rd: add button action to the controller, 4th: present the
alert to the user
                    // NB in this case as using TableViewCell, cannot present Alert
so use underlying ViewController instead
                    print(error?.localizedDescription) // log
            }
            else // if user found in server's PlayerID table
            {
                self.playerIDArray.removeAll(keepingCapacity: false)
                //for object in objects!
                for poster in posters!
                {
                    // should only be 1 record for this user/device - get the
playerID
                    self.playerIDArray.append(poster.object(forKey: "playerID")
as! String)
                }
            }
        })
    }
}

```

```
//OneSignal.postNotification(<#T##jsonData: [AnyHashable : Any]!##[AnyHashable : Any]!#>)
    OneSignal.postNotification(["contents": ["en": "Someone, \
(PFUser.current()!.username!), liked your post"], "include_player_ids": \
[self.playerIDArray.last]])
        // as per docs in https://documentation.onesignal.com/docs/
ios-native-sdk#section--postnotification-
    }
}
})
}
}
}
```



NB why sent thrice?

17-169

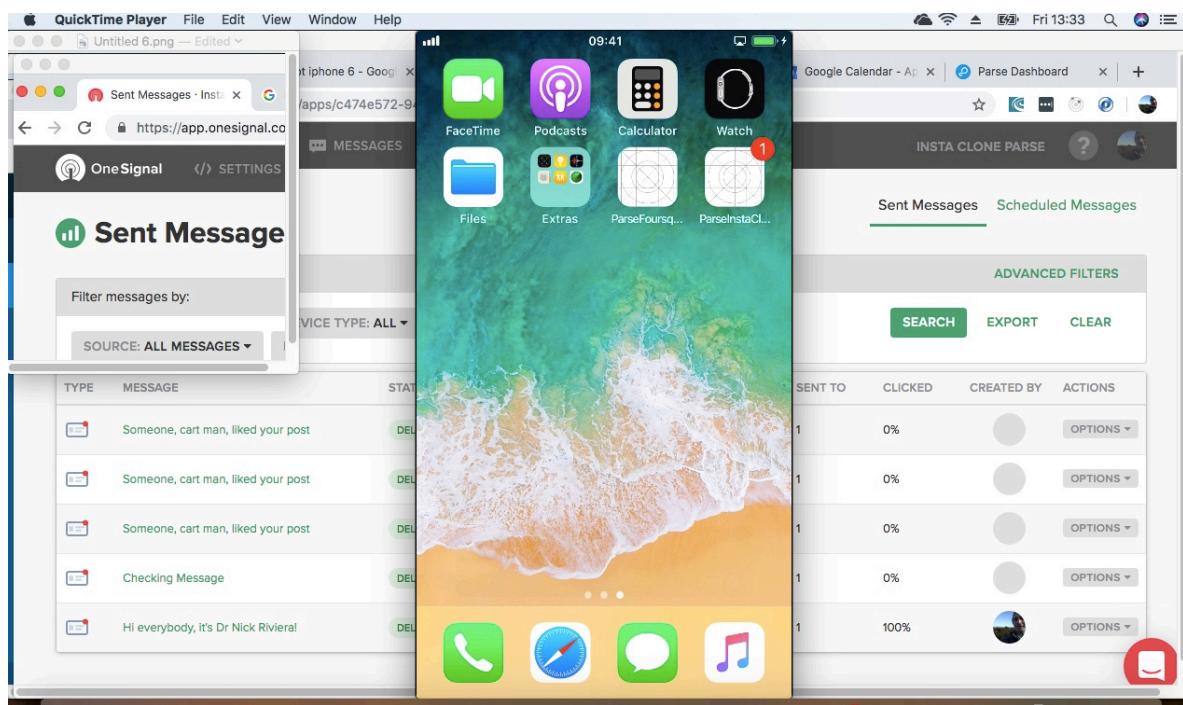
fix: query.limit = 1 // to deal with possible duplicate records when loop following copy/paste for comments to do push notifications as per for likes

17-170 Incrementing Badge number (showing number of backlogged notifications, as shown in top-right of app icon in Home screen)

```
//OneSignal.postNotification(<#T##jsonData: [AnyHashable : Any]!##[AnyHashable : Any]!#>)
```

```
    OneSignal.postNotification(["contents": ["en": "Someone, \(PFUser.current()!.username!), commented on your post"],  
    "include_player_ids": [self.playerIDArray.last], "ios_badgeType" : "Increase",  
    "ios_badgeCount" : "1"])
```

// as per docs in <https://documentation.onesignal.com/docs/ios-native-sdk#section--postnotification->



18-172 sketch intro

18-173 Sketch app specifically meant for iOS apps. (www.sketchapp.com)

Sketch *not* free of charge

using GIMP instead

18-174-18-178 vector graphics manipulations including union/subtract, rotation,

18-179 programmatically manipulating colours eg

```
myLabel.backgroundColor = UIColor(red: 147/255, green: 235/255, blue:  
20/255, alpha: 100)
```

NB Xamarin 15/4/2019

NB if taking a long time to restart after Mac crash, don't panic - either wait (can take 10 mins or more) or try restarting again if definitely stuck. Also NB verbose / &c restart Mac options.

/Users/peterjenkin/Downloads/Xcode.app

Visual Studio Preferences/SDK Locations/Apple. - eg /Users/peterjenkin/Applications/Xcode.app
(also Android) - listings of Android versions downloaded and Found or Not Found for Android SDK, Native NDK and Java JDK

Visual Studio iOS in Xamarin ok on Mac 15/4/2019

Visual Studio Android in Xamarin crashing machine on Mac 15/4/2019

NB Android Studio

Gradle Sync taking several minutes on first build

completed successfully	9 s 952 ms
Run build	9 s 521 ms
Load build	9 ms
Evaluate settings	8 ms
Finalize build cache configuration	
Configure build	5 s 521 ms
Load projects	6 ms
Configure project :	2 s 590 ms
Configure project :app	2 s 590 ms
Calculate task graph	326 ms
Run tasks	3 s 622 ms
:app:preBuild	89 ms
:app:preDebugBuild	2 s 93 ms
:app:compileDebugAidl	
:app:compileDebugRenderscript	481 ms
:app:checkDebugManifest	7 ms
:app:generateDebugBuildConfig	119 ms
:app:prepareLintJar	179 ms
:app:generateDebugSources	1 ms

NB for HAXM Hardware Acceleration with Visual Studio (something similar with

HAXM had to be done with Visual Studio in order for its iOS emulator to work ok) - on Mac

<https://stackoverflow.com/questions/31508240/intel-haxm-installation-failed>

```
PNJ-dev-Mac-09:Hardware_Accelerated_Execution_Manager peterjenkin$  
sudo /Users/peterjenkin/Library/Android/sdk/extras/intel/  
Hardware_Accelerated_Execution_Manager/silent_install.sh  
Silent installation Pass!  
PNJ-dev-Mac-09:Hardware_Accelerated_Execution_Manager peterjenkin$
```

Emulator: Warning: Quick Boot / Snapshots not supported on this machine. A CPU with EPT + UG features is currently needed. We will address this in a future release.

<https://github.com/intel/haxm/issues/30>

The issues have to do with the specific device you're trying to emulate and the processor you have (**intel core 2 or older**)

To solve it:

1. Create a new emulator
2. Select the device you would like to emulate.
3. On the next page with the header "**System Image**" select the tab "other images"
4. Download and select an image that is listed as armeabi-v7a under the ABI column.

did so - no longer crashing, but emulator device not running

<https://stackoverflow.com/a/53193001>

builds.... "...Waiting for target device to come online"

```
if error PNJ-dev-Mac-09:platform-tools peterjenkin$ adb  
-bash: adb: command not found
```

Add the following to `~/.bash_profile`

```
export PATH=~/Library/Android/sdk/tools:$PATH  
export PATH=~/Library/Android/sdk/platform-tools:$PATH
```

then from terminal

```
source ~/.bash_profile
```

<https://stackoverflow.com/a/30574922>

via Nox app player attempt

<https://www.youtube.com/watch?v=cCCW3bh9jNw>

<https://stackoverflow.com/questions/51214825/adb-cant-connect-to-nox>

<https://forum.xda-developers.com/tools/android-studio/how-to-connect-android-studio-nox-app-t3241330>

PNJ-dev-Mac-09:~ peterjenkin\$ **adb connect 127.0.0.1:62001**

connected to 127.0.0.1:62001

PNJ-dev-Mac-09:~ peterjenkin\$

19-180 no coding in this section !!! Only drawing icons &c

19-181

19-182 Sketch has several canvas settings for various iPad & iPhone devices, also iOS icons, Mac icons, Android icons

<https://developer.apple.com/library/archive/documentation/DeviceInformation/Reference/iOSDeviceCompatibility/Displays/Displays.html>

Table 2-1 Screen Geometry

Device	Native Resolution (Pixels)	UIKit Size (Points)	Native Scale factor	UIKit Scale factor
iPhone X	1125 x 2436	375 x 812	3.0	3.0
iPhone 8 Plus	1080 x 1920	414 x 736	2.608	3.0
iPhone 8	750 x 1334	375 x 667	2.0	2.0
iPhone 7 Plus	1080 x 1920	414 x 736	2.608	3.0
iPhone 6s Plus	1080 x 1920	375 x 667	2.608	3.0
iPhone 6 Plus	1080 x 1920	375 x 667	2.608	3.0
iPhone 7	750 x 1334	375 x 667	2.0	2.0
iPhone 6s	750 x 1334	375 x 667	2.0	2.0
iPhone 6	750 x 1334	375 x 667	2.0	2.0

UIKit Size (points) column is used for graphics canvas (in pixels)

TODO: native resolution vs UIKit Size

compare with mock-up while amending 'art-board'

Inkscape

Warning: Your DNS servers incorrectly claim to know the address of nonexistent hosts. This may cause checksum mismatches for some ports. See this page for more information: <<https://trac.macports.org/wiki/MisbehavingServers>>

In -s /opt/local/bin/inkscape /Applications/Inkscape # not pretty, but work

Inkscape - needing XQuartz

TODO: XQuartz, X11 - lookup

<https://inkscape.org/en/doc/tutorials/advanced/tutorial-advanced.html>

<https://inkscape.org/en/doc/tutorials/basic/tutorial-basic.html>

Ctrl+- to subtract. Ctrl++ to union - or Path/Union &c

Star/Polygon for triangle, or Extensions/Render/Triangle

Measure tool - set Units at top to Pixels

Ctrl+G to group Ctrl+U to ungroup.

Objects... to see objects, click on 'Set' after typing 'Name' and/or 'Title'

Ctrl+Shift+M or Object/Transform for Transform dialogue to change height/width &c of object, or select and then use input boxes at top

snap options at right-hand edge

editing vertices - <http://tavmjong.free.fr/INKSCAPE/MANUAL/html/Paths-Editing.html>. and <https://graphicdesign.stackexchange.com/questions/103684/how-to-add-vertices-to-rectangle>

Edit vertices tool on left toolbar to fettle union'd shapes &c
stroke of same colour, rounded corners & many pixels for rounded triangle
graphic design == my passion ;)

Trace bitmap to copy some object shape and use 'Remove background' to
make any traced object background transparent
colour Dropper by D or F7

workarounds for exporting individual objects from Inkscape

<https://graphicdesign.stackexchange.com/a/36577>

or select object, **File/Export PNG Image**, change file name and path in
'Export As' dialogue and click Save and *then* make sure to click '**Export**'.

Inkscape has its own SVG format (there are several) TODO look up SVG formats

NB Mac : Cmd+Option+D to hide/show Dock. or. System Preferences/Dock/Automatically Hide & Show Dock

19-190
XCode tabbed app
Ctrl+drag Segue from entry point view controller, but select 'Relationship segue / view controllers'
Drag in 3 extra view controllers
Name view controllers' Title (in Attribute Inspector) to Home, Explore, Videos, News & Profile
Set initial entry point view controller's 'Bottom Bar' to different shade (opaque black tab bar)

Drag table view and declare 1 prototype cell, to show mocked-up posts. 'No selection' for table view selection. Leave 'separators' as-is.

Ctrl-Drag tableView outlet into first view controller vc code. Type Identifier "Cell" for prototype cell (no separate class as yet).

as per usual: class FirstViewController: UIViewController,
UITableViewDataSource, UITableViewDelegate {

```
...    tableView.delegate = self
      tableView.dataSource = self
```

```
...
func tableView(_ tableView: UITableView, cellForRowAt indexPath:
IndexPath) -> UITableViewCell {
    <#code#>
}
```

```
func tableView(_ tableView: UITableView, numberOfRowsInSection section:
Int) -> Int {
    <#code#>
}
```

then.

```
func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath)
-> UITableViewCell {
    let cell = tableView.dequeueReusableCell(withIdentifier: "Cell", for:
indexPath)
    return cell
}
```

```
func tableView(_ tableView: UITableView, numberOfRowsInSection section:
Int) -> Int {
    return 10
}
```

```
// set height of prototype cell programmatically using delegate function  
https://stackoverflow.com/a/46438939  
func tableView(_ tableView: UITableView, heightForRowAt indexPath:  
IndexPath) -> CGFloat {  
    return 400  
}
```

Labels and image view into prototype cell. put example image.

Export/Save Inkscape objects as PNG files. Drag & copy into project. Add buttons to prototype cells, remove 'Title' and set Image to appropriate icon in each case.

Drag image files into **Assets.xcassets**, and for each, set **Image Set/Render As to 'Original Image'**

else images may appear as grey/blue squares only. - <https://stackoverflow.com/a/37281591/11365317>

NB transparency around edges is a good thing for exported button icon images !!

NB Assets.xcassets can have several sizes of image for different screens/zooms ?? TODO cf 19-193

19-192 Tab bar items/Bar item - 'Image'. : home.png, explore.png,

19-193 Suffix '@2x' to a file name (before extension) eg profile@2x.png, profile@3x.png - Sketch app will do this neatly - **Assets** of various sizes cf 20-201

20-196 Sketch templates for UI dialogues, keyboard, &c (iOS UI Design), backgrounds & templates for iOS icons (iOS App Icon)

20-197

InVision

Zeplin

look out for free-of-charge templates usually with .sketch or .psd format

20-198 shared background style, once defined, handy for duplicating across assets

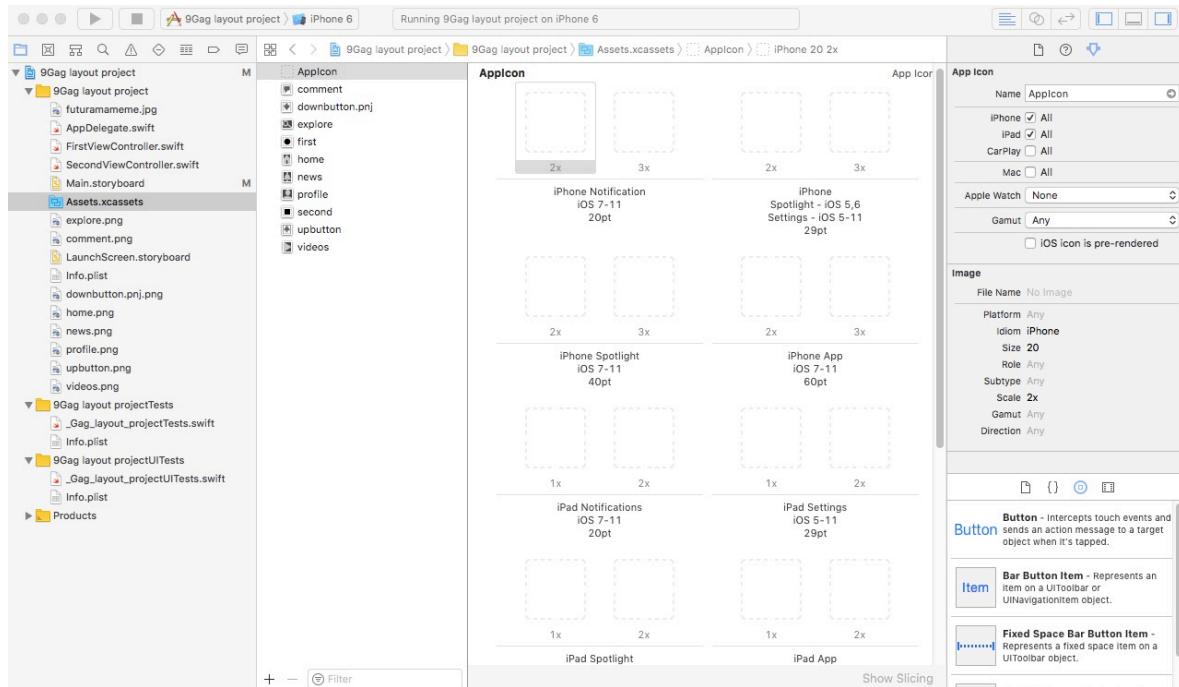
20-199 copy icon from largest template to smaller, and scale down to fit proportionally - indication to sizes given in file name (including for iTunes large icon if wanted)

20-200 Assets named eg App Icon-60@3x, App Icon-60@2x, App Icon-Small-60@3x,

Export icons of all sizes individually as PNG (optionally re-naming according to size in pixels)

20-201 Appicon in Assets.xcassets d'contain several slots for icons of different types & sizes (notification, spotlight, app store) - drag & drop image file into slots - a warning will be given if the wrong size, for example. 2x and 3x categories are listed, so proportionately sized images can be dropped into the appropriate slots. App icon in home screen is included in this functionality, as

well as search (spotlight)bin.



21-204 Single View app in XCode, install Parse with Cocoa Pods, setup UI of 1st view controller

NB bolts <https://cocoapods.org/pods/Bolts> library for tasks (cf js Promise) written by Parse/Fbook

<https://guides.cocoapods.org/using/using-cocoapods.html#should-i-check-the-pods-directory-into-source-control>

NB gitignore for Swift. <https://github.com/github/gitignore/blob/master/Swift.gitignore>

21-205 AWS EC2 setup for Parse

going to keep same EC2 instance as for section 16

in case needed to export/download image of EC2, <https://docs.aws.amazon.com/vm-import/latest/userguide/vmexport.html>

NB slightly worrying that AWS root password the same as amazon.co.uk password

21-206 (as per 16-142 & 16-143) SSH into Parse server to get server.js details for setting up Parse config

copied pem file over to project folder (do not add to git) and chmod 400 ParseInstaClone.pem as recommended by AWS Connect prompt (my pem filename had no spaces)

```
ssh -i "ParseInstaClone.pem" ubuntu@ec2-34-251-9-205.eu-west-1.compute.amazonaws.com
```

use login/password found earlier in 16-142/143 from screen dump

NB clear bash shell command line: Clean up the line: Ctrl + E Ctrl + U to wipe the current line in the terminal. Clean up the line: Ctrl + A Ctrl + K to wipe the current line in the terminal. Cancel the current command/line: Ctrl + C . Recall the deleted command: Ctrl + Y (then Alt + Y)

```
ssh -i "ParseInstaClone.pem" ubuntu@ec2-34-251-9-205.eu-west-1.compute.amazonaws.com  
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.4.0-1079-aws x86_64)
```

```
*** Welcome to the Bitnami Parse Server 3.2.3-0 ***
*** Documentation: https://docs.bitnami.com/aws/apps/parse/ ***
***           https://docs.bitnami.com/aws/ ***
*** Bitnami Forums: https://community.bitnami.com/ ***
Last login: Sat Apr 13 09:01:21 2019 from 86.148.178.237
```

```
bitnami@ip-172-31-26-254:~$ ls
apps bitnami_credentials htdocs stack
bitnami@ip-172-31-26-254:~$ cd apps
bitnami@ip-172-31-26-254:~/apps$ ls
bitnami parse rockmongo
bitnami@ip-172-31-26-254:~/apps$ cd parse
bitnami@ip-172-31-26-254:~/apps/parse$ ls
bnconfig conf htdocs licenses scripts updateip
bitnami@ip-172-31-26-254:~/apps/parse$ cd htdocs
bitnami@ip-172-31-26-254:~/apps/parse/htdocs$ cat server.js
var express = require('express');
var ParseServer = require('parse-server').ParseServer;
var app = express();
// Specify the connection string for your mongodb database
// and the location to your Parse cloud code
var api = new ParseServer({
  databaseURI: "mongodb://root:ammfOJ3HCNgj@127.0.0.1:27017/
bitnami_parse",
  cloud: "./node_modules/parse-server/lib/cloud-code/Parse.Cloud.js",
  appId: "4db3417ac4d79c34640760d3c13c3728cd26604a",
  masterKey: "571c1e762705f9975f4qafdf4f91cb373d6e23d9f".
```

```

    fileKey: "a1fce6651b8726a4ef991570632615baf128d9f3",
    serverURL: "http://34.251.9.205:80/parse"
  });
  // Serve the Parse API on the /parse URL prefix
  app.use('/parse', api);

  var port = 1337;
  app.listen(port, function() {
    console.log('parse-server running on port ' + port);
  });

  //Parse Dashboard
  var ParseDashboard = require('parse-dashboard');
  var dashboard = new ParseDashboard({
    apps: [
      {
        appName: "My Bitnami Parse API",
        appId: "4db3417ac4d79c34640760d3c13c3728cd26604a",
        masterKey: "571c1e762705f9975f4aaf4f91cb373d6e23d9f",
        fileKey: "a1fce6651b8726a4ef991570632615baf128d9f3",
        production: true,
        serverURL: "http://34.251.9.205:80/parse"
      }
    ],
    users: [
      {
        user: "user",
        pass:
        "$2a$10$fW6IQJ2rZL5dTMFwq277Ou0En2YjMfpwSxB6BupSHYtt7V./Yf0iO"
      }
    ],
    useEncryptedPasswords: true
  });
}

var allowInsecureHTTP = true;

// Serve the Parse Dashboard on the /parsedashboard URL prefix
app.use('/', dashboard);

var portdash = 4040;
app.listen(portdash, function() {
  console.log('parse-dashboard running on port ' + portdash);
});bitnami@ip-172-31-26-254:~/apps/parse/htdocs$
```

```

func application(_ application: UIApplication,
didFinishLaunchingWithOptions launchOptions:
```

```

[UIApplicationLaunchOptionsKey: Any]?) -> Bool {
    // Override point for customization after application launch.

    let myConfig = ParseClientConfiguration
    { (ParseMutableClientConfiguration) in
        ParseMutableClientConfiguration.applicationId =
"4db3417ac4d79c34640760d3c13c3728cd26604a"
        ParseMutableClientConfiguration.clientKey =
"571c1e762705f9975f4aaf4f91cb373d6e23d9f"
        ParseMutableClientConfiguration.server = "http://34.251.9.205:80/
parse"
    }

    Parse.initialize(with: myConfig)

    let defaultACL = PFACL()
    defaultACL.hasPublicReadAccess = true
    defaultACL.hasPublicWriteAccess = true

    PFACL.setDefault(defaultACL, withAccessForCurrentUser: true)
    // giving permission to edit objects

    return true
}
//in server.js - applicationId: appId ; clientKey: masterKey; server: serverURL

```

Since using HTTP not HTTPS, also in Info.plist 'Add row', ' Transport Security Settings', first click left triangle til pointing down instead of to right then click '(+)' to add a sort of sub-row, and select 'Allow Arbitrary Loads' and amend value to 'YES'

for ec2-34-251-9-205.eu-west-1.compute.amazonaws.com/apps **Parse web interface** use login/password found earlier in 16-142/143 from screen dump - user / **ammfOJ3HCNgj**

NB might skimp on notes from hereon except where not duplicating earlier sections

20-208 (after making record) ACL visible in Parse web interface for records e.g. 'Public Read + Write' as per , in code, defaultACL.hasPublicReadAccess = true

defaultACL.hasPublicWriteAccess = true

NB git add -u to add tracked files (can avoid serious omissions)

james 123456
cart man 123456

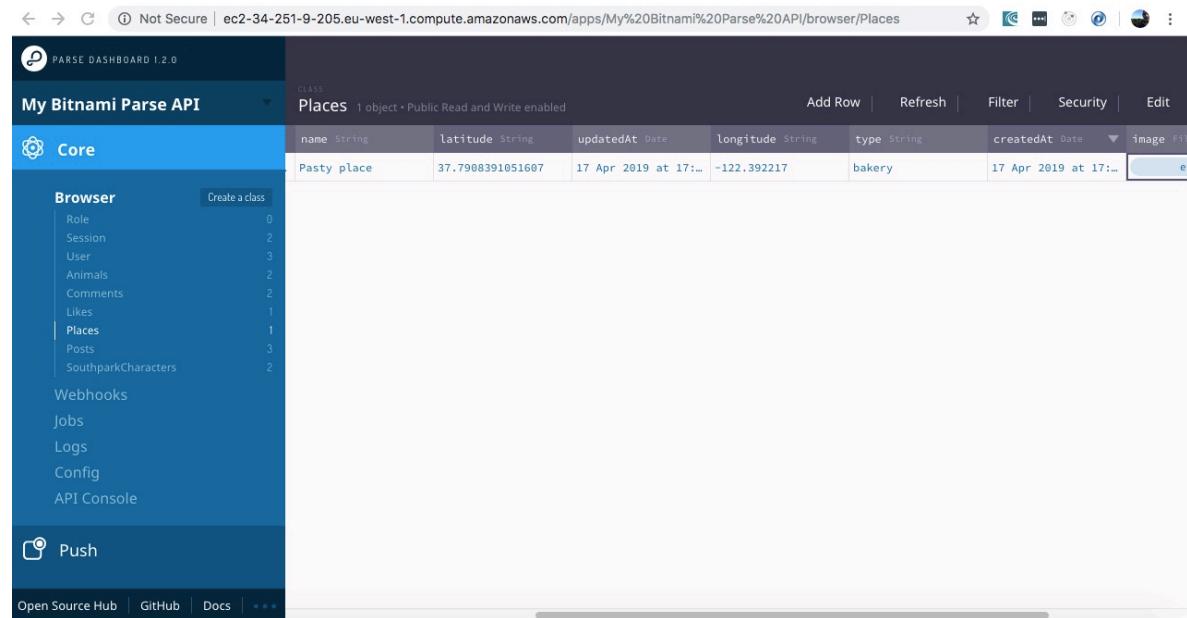
Parse very easy for logins and data writing/reading
this time building single tabbed but with multiple view controllers

remember login reset entry point - // set & use Storyboard ID for entry point View Controller(s) - initial entry and navigation VC for reset-to VC - in order to redefine entry points

use new way of annotations, as per my code in section 12

```
import UIKit
```

```
// global variables - accessible from any view controller
var globalName = ""
var globalType = ""
var globalAtmosphere = ""
var globalImage = UIImage()
```



The screenshot shows the Parse Dashboard interface. The top navigation bar indicates "Not Secure" and the URL "ec2-34-251-9-205.eu-west-1.compute.amazonaws.com/apps/My%20Bitnami%20Parse%20API/browser/Places". The main content area displays the "Places" class table. The table has columns: name (String), latitude (String), updatedAt (Date), longitude (String), type (String), createdAt (Date), and image (File). A single row is shown: "Pasty place" with latitude 37.798391051607, updated at 17 Apr 2019 at 17:21:17, longitude -122.392217, type bakery, and createdAt 17 Apr 2019 at 17:21:17. The dashboard sidebar on the left lists various core components: Browser, Push, Open Source Hub, GitHub, and Docs.

name	latitude	updatedAt	longitude	type	createdAt	image
Pasty place	37.798391051607	17 Apr 2019 at 17:21:17	-122.392217	bakery	17 Apr 2019 at 17:21:17	

The screenshot shows the Parse Dashboard interface. On the left, there's a sidebar titled 'Core' with sections like 'Browser', 'Places', and 'Push'. The 'Places' section is expanded, showing a table of objects with columns: objectId, atmosphere, ACL, name, latitude, updatedAt, and longitude. There are 5 objects listed.

	objectId	atmosphere	ACL	name	latitude	updatedAt	longitude
0	BVeqAPhwYR	casual	Public Read + Write	Fred's Fiery Food	37.7834103535819	17 Apr 2019 at 18:00:00 UTC	-122.4
1	YN4oVd33Aa	smart	Public Read + Write	Ricky Restaurant	37.7780885902237	17 Apr 2019 at 18:00:00 UTC	-122.3
2	0fdLwIKTbw	friendly	Public Read + Write	PJ Pasty Place	37.7868350481522	17 Apr 2019 at 17:59:59 UTC	-122.3
3	Gtlfw3FB8d	friendly	Public Read + Write	James Student Bar	37.7974242533453	17 Apr 2019 at 17:59:59 UTC	-122.3
4	4YOqvKw34D	casual	Public Read + Write	Pasty place	37.7908391051607	17 Apr 2019 at 17:59:59 UTC	-122.3

Parse tending to have Initial Capital table/object names and lower case 'field' names

los of copy/paste'ing of alert code

TODO: how to do map with several pinview pins?