iOS alapú szoftverfejlesztés - Labor 12

A laborsegédletet összeállította: Blázovics László, Kántor Tibor és Kelényi Imre

A labor témája:

* + Third-party Swift komponens integrálása
  + Objective-C kód integrálása
  + A Cocoapods használata

A laborhoz tartozó nagyobb kódrészletek a következő url-en érhetők el "copy-paste barát" formában:

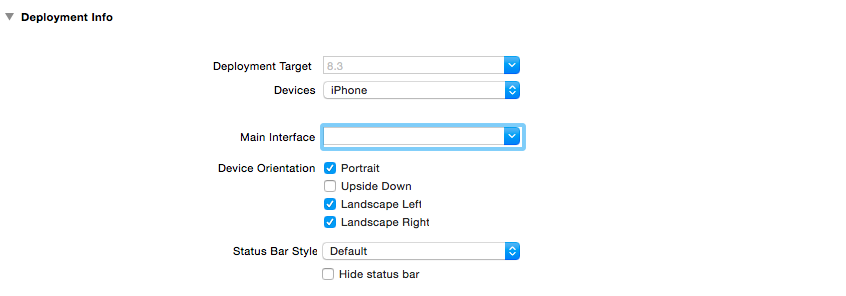
https://gist.github.com/blazovics/416a6c5d431ffdd161a9

# ImageDownloader

Alkalmazásunk a fortepan.hu-ról fog letölteni közepes, illetve nagy felbontású képeket. Utóbbi esetén azt is szeretnénk megmutatni a felhasználónak, hogy hogyan halad a letöltés.

Kiindulásnak hozzunk lére egy SingleView applicationt **PictureDownload** néven.

A létrehozott projektben töröljük ki a Main.Storyboard-ot és ezt vezessük át project beállításaiban is, nevezetesen töröljük ki a Main interface mező tartalmát a Deployment Info résznél.



## ViewController inicializálása kódból

Adjuk hozzá az alábbi kódrészletet az AppDelegate didFinishLaunchingWithOptions metódusához:

func application(application: UIApplication, didFinishLaunchingWithOptions launchOptions: [NSObject: AnyObject]?) -> Bool {

self.window = UIWindow(frame: UIScreen.mainScreen().bounds)

let viewController = ViewController()

viewController.imageUrl =

NSURL(string: "http://www.fortepan.hu/\_photo/display/28268.jpg")

viewController.contentUrl =

NSURL(string:

"http://www.fortepan.hu/\_photo/download/fortepan\_28268.jpg")

self.window!.rootViewController = viewController

self.window!.makeKeyAndVisible()

return true

}

https://gist.github.com/blazovics/416a6c5d431ffdd161a9#file-appdelegate-didfinishlaunchingwithoptions-swift

Ezután menjünk át a ViewController.swift-re és kezdjük el kiegészíteni.

Hozzunk létre egy **imageView** nevű UIImageView property-t, hogy szükség esetén le tudjuk cserélni a benne lévő képet, illetve egy **contentUrl** és egy **imageUrl** property-t

class ViewController: UIViewController {

var contentUrl:NSURL?

var imageUrl:NSURL?

var imageView:UIImageView?

Majd cseréljük le a viewDidLoad metódust a következő kódrészlettel:

override func viewDidLoad() {

super.viewDidLoad()

imageView = UIImageView(frame: CGRectMake(self.view.bounds.origin.x,

self.view.bounds.origin.y, self.view.bounds.size.width,

self.view.bounds.size.height/2))

imageView!.contentMode = UIViewContentMode.ScaleAspectFit

imageView!.backgroundColor = UIColor.blackColor()

self.view.addSubview(imageView!)

let downloadButton =

UIButton.buttonWithType(UIButtonType.System) as! UIButton

downloadButton.frame = CGRectMake(0, 0, 100, 40)

downloadButton.center = CGPointMake(self.view.bounds.width/2,

self.view.bounds.size.height\*5/8)

downloadButton.setTitle("Download", forState: UIControlState.Normal)

downloadButton.setTitleColor(

UIColor.blackColor(), forState: UIControlState.Normal)

downloadButton.addTarget(

self, action: "downloadFile",

forControlEvents: UIControlEvents.TouchUpInside)

self.view.addSubview(downloadButton)

let displayButton =

UIButton.buttonWithType(UIButtonType.System) as! UIButton

displayButton.frame = CGRectMake(0, 0, 100, 40)

displayButton.center =

CGPointMake(self.view.bounds.width/2, self.view.bounds.size.height\*6/8)

displayButton.setTitle("Present", forState: UIControlState.Normal)

displayButton.setTitleColor(

UIColor.blackColor(), forState: UIControlState.Normal)

self.view.backgroundColor = UIColor.whiteColor()

self.view.addSubview(displayButton)

}

https://gist.github.com/blazovics/416a6c5d431ffdd161a9#file-viewcontroller-viewdidload-swift

Teszteljük az alkalmazást!

Látható, hogy minden UI elem megjelenik, de a kép nem töltődik be, és a letöltés gomb megnyomásakor is hibát kapunk.

A kép letöltéséhez használjuk a következő kódrészletet:

override func viewWillAppear(animated: Bool) {

if imageUrl != nil{

var request: NSURLRequest = NSURLRequest(URL: imageUrl!)

let queue:NSOperationQueue = NSOperationQueue()

NSURLConnection.sendAsynchronousRequest(

request, queue: queue, completionHandler:{

[weak imageView] response, data, error -> Void in

if let image = UIImage(data: data) {

imageView?.image = image

}

})

}

}

https://gist.github.com/blazovics/416a6c5d431ffdd161a9#file-viewcontroller-viewwillappear-swift

Most, hogy letöltődik a közepes méretű kép, töltsük le a nagyobb képet is. Ehhez először implementáluk az NSURLConnection-höz szükséges delegateket és további három property-t, amik a letöltött nagy képet , illetve a letöltés állapotát fogják nyilvántartani.

class ViewController: UIViewController, NSURLConnectionDelegate, NSURLConnectionDataDelegate{

var data: NSMutableData?

var contentLength: Int64 = 0

var downloadedContentLength: Int64 = 0

var contentUrl:NSURL?

var imageUrl:NSURL?

var imageView:UIImageView?

Ezután implementáljuk a downloadFile() nevű függvényt, amit a Download gomb eseménykezelője hív meg:

func downloadFile(){

data = NSMutableData()

var request: NSURLRequest = NSURLRequest(URL: contentUrl!)

var connection: NSURLConnection = NSURLConnection(

request: request, delegate: self, startImmediately: true)!

connection.start()

}

Végül implementáljuk a szükséges delegate metódusokat:

func connection(connection: NSURLConnection,

didReceiveResponse response: NSURLResponse) {

if let httpResponse = response as? NSHTTPURLResponse {

contentLength = httpResponse.expectedContentLength

downloadedContentLength = 0

println(httpResponse.expectedContentLength)

}

}

func connection(connection: NSURLConnection, didReceiveData data: NSData){

downloadedContentLength += data.length

if contentLength > 0 {

let percentage =

Double(downloadedContentLength)/Double(contentLength) \* 100.0

println(percentage)

}

self.data?.appendData(data)

}

func connectionDidFinishLoading(connection: NSURLConnection)

{

}

func connection(connection: NSURLConnection,

didFailWithError error: NSError) {

print(error.description)

}

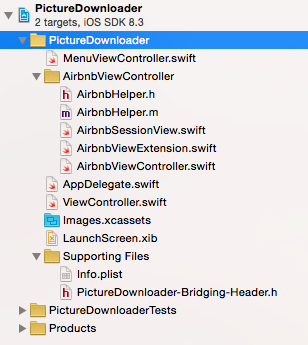
## AirbnbViewController

Most, hogy van már rendes tartalmunk adjuk hozzá az alkalmazáshoz egy menüt.

Ehhez töltsük le az AirbnbViewController thrid-party komponenst a következő linkről:

<https://github.com/pixyzehn/AirbnbViewController/archive/master.zip>

Csomagoljuk ki a fájlt és keressük meg az AirbnbViewController mappát, amiben négy forrásfájl taláható. Az egész mappát adjuk hozzá a projekthez.



Ezután keressük meg a AirbnbViewController-Sample mappában a MenuViewController.swift és a AirbnbViewExtension.swift fájlokat és azt is adjuk hozzá a kódhoz.

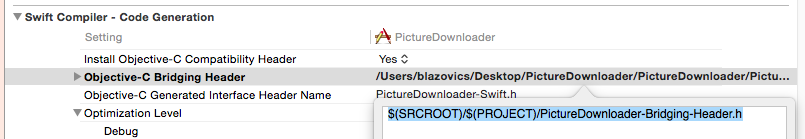
Hozzunk lére egy úgynevezett Bridging header fájlt **PictureDownloader-Bridging-Header.h** névvel, hogy használni tudjuk az Objective-C-s kód komponenseket. Miután létrehoztuk a fájlt, töröljük ki a tartalmát és adjuk hozzá a következő sort:

#import "AirbnbHelper.h"

Majd menjünk át a projektbeállításokhoz, azon belül is Build settings-hez és állítsuk be a Brigding headert.

Ha a projekt gyökérmappájába tettük a fájlt, akkor ezt a sort kell belemásolni:

$(SRCROOT)/$(PROJECT)/PictureDownloader-Bridging-Header.h



Fordítsuk le az alkalmazást!

Ha minden rendben ment, akkor módosítsuk az AppDelegate didFinishLaunching metódusát, hogy a MenuViewController jelenjen meg.

func application(application: UIApplication, didFinishLaunchingWithOptions launchOptions: [NSObject: AnyObject]?) -> Bool {

// Override point for customization after application launch.

self.window = UIWindow(frame: UIScreen.mainScreen().bounds)

let viewController = ViewController()

viewController.imageUrl =

NSURL(string: "http://www.fortepan.hu/\_photo/display/28268.jpg")

viewController.contentUrl =

NSURL(string:

"http://www.fortepan.hu/\_photo/download/fortepan\_28268.jpg")

let controller: UINavigationController =

UINavigationController(rootViewController: viewController)

let menuController: MenuViewController =

MenuViewController(viewController: controller,

atIndexPath: NSIndexPath(forRow: 0, inSection: 0))

self.window!.rootViewController = menuController

//self.window!.rootViewController = viewController

self.window!.makeKeyAndVisible()

return true

}

Futtassuk az alkalmazást!

Egészítsük ki ViewController.swift viewDidLoad metódusát, hogy meg tudjuk nyitni az oldalsó menüt. Ehhez a következő kódrészlet kell csak beilleszteni:

var button: UIButton =

UIButton.buttonWithType(UIButtonType.Custom) as! UIButton

button.frame = CGRectMake(0, 0, 50, 35)

button.setTitle("Menu", forState: UIControlState.Normal)

button.setTitleColor(UIColor(red:0.3, green:0.69, blue:0.75, alpha:1), forState: UIControlState.Normal)

button.addTarget(self, action: "leftButtonTouch",

forControlEvents: UIControlEvents.TouchUpInside)

Továbbá írjuk meg a gomb eseménykezelőjét:

func leftButtonTouch() {

self.airViewController.showAirViewFromViewController(

self.navigationController, complete: nil)

}

Már van menünk, de a többi menüelemhez nincs képünk. Ezt orvosolandó, módosítsuk a MenuViewControllert!

func viewControllerForIndexPath(indexPath: NSIndexPath) -> UIViewController {

let viewController: ViewController = ViewController()

let controller: UINavigationController =

UINavigationController(rootViewController: viewController)

switch indexPath.row {

case 0:

viewController.view.backgroundColor =

UIColor(red:0.13, green:0.14, blue:0.15, alpha:1)

viewController.imageUrl =

NSURL(string: "http://www.fortepan.hu/\_photo/display/28268.jpg")

viewController.contentUrl =

NSURL(string: "http://www.fortepan.hu/\_photo/download/fortepan\_28268.jpg")

case 1:

viewController.view.backgroundColor =

UIColor(red:0.95, green:0.95, blue:0.95, alpha:1)

viewController.imageUrl =

NSURL(string: "http://www.fortepan.hu/\_photo/display/74394.jpg")

viewController.contentUrl =

NSURL(string: "http://www.fortepan.hu/\_photo/download/fortepan\_74394.jpg")

case 2:

viewController.view.backgroundColor =

UIColor(red:0.8, green:0, blue:0.48, alpha:1)

viewController.imageUrl =

NSURL(string: "http://www.fortepan.hu/\_photo/display/16022.jpg")

viewController.contentUrl =

NSURL(string: "http://www.fortepan.hu/\_photo/download/fortepan\_16022.jpg")

default:

break

}

return controller

}

https://gist.github.com/blazovics/416a6c5d431ffdd161a9#file-menuviewcontroller-viewcontrollerforindexpath-swift

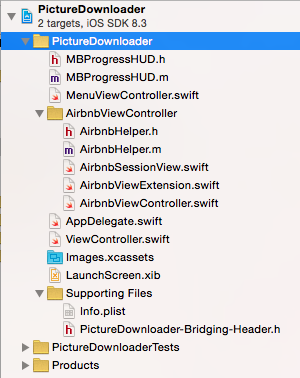
## MBProgressHUD

Most, hogy már van mit letölteni adjuk hozzá a projekthez az MBProgressHUD third-party plugint, amivel különböző progress bar-okat tudunk megjeleníteni.

Ehhez először töltsük le a forrást majd csomagoljuk is ki.

<https://github.com/jdg/MBProgressHUD/archive/master.zip>

Keressük meg és adjuk hozzá a projekthez az MBProgressHUD.h és az MBProgressHUD.m fájlokat!



Mivel ezek Objective-C-ben írt állományok így egészítsük ki a PictureDownloader-Bridging-Header.h-t az #import “MBProgressHUD.h” direktívával!

#import "AirbnbHelper.h"

#import "MBProgressHUD.h"

Ezután egészítsük ki a ViewController-t, hogy letöltés közben kijelződjön a letöltés állapota.

Hozzunk létre egy MBProgressHUD típusú property-t, hogy később bárhonnan el tudjuk érni:

var contentUrl:NSURL?

var imageUrl:NSURL?

var imageView:UIImageView?

var HUD:MBProgressHUD?

override func viewDidLoad() {

Ezután inicializáljuk a viewDidLoad metódus végén.

self.navigationItem.leftBarButtonItem =

UIBarButtonItem(customView: button)

self.view.addSubview(displayButton)

HUD = MBProgressHUD(view: self.view)

self.view.addSubview(HUD!)

}

Amikor elindul a letöltés jelenítsük meg. Ehhez a downloadFile()-t kell kiegészíteni:

func downloadFile(){

data = NSMutableData()

var request: NSURLRequest = NSURLRequest(URL: contentUrl!)

var connection: NSURLConnection = NSURLConnection(request: request,

delegate: self, startImmediately: true)!

connection.start()

HUD?.show(true)

}

Ha sikerült kitalálnia az NSURLConnectionnek, hogy mekkora a fájl, amit le akarunk tölteni, akkor váltsunk megjelenítési módot:

func connection(connection: NSURLConnection, didReceiveResponse response: NSURLResponse) {

if let httpResponse = response as? NSHTTPURLResponse {

contentLength = httpResponse.expectedContentLength

downloadedContentLength = 0

println(httpResponse.expectedContentLength)

HUD?.mode = MBProgressHUDMode.Determinate

}

}

Az egyes adatcsomagok beérkezésekor frissítsük a HUD-ot.

func connection(connection: NSURLConnection, didReceiveData data: NSData){

downloadedContentLength += data.length

if contentLength > 0 {

let percentage =

Double(downloadedContentLength)/Double(contentLength) \* 100.0

println(percentage)

HUD?.progress =

Float(downloadedContentLength)/Float(contentLength)

HUD?.labelText = NSString(format: "%.0f%%", percentage) as String

}

self.data?.appendData(data)

}

Végül, ha minden letöltődött, rejtsük el.

func connectionDidFinishLoading(connection: NSURLConnection)

{

HUD?.hide(true)

}

## CocoaPods

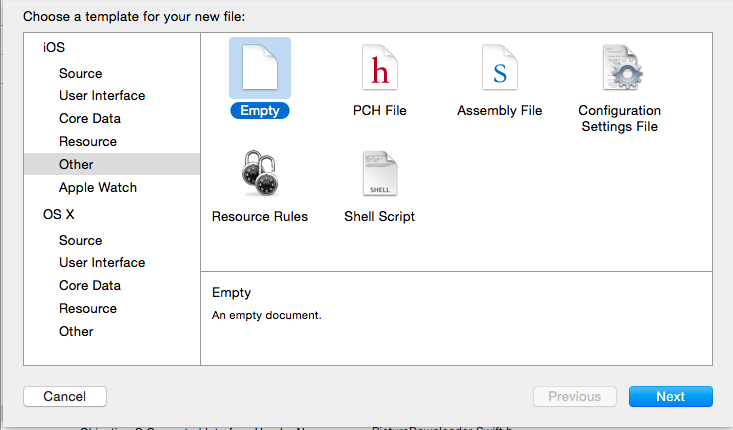
Végül szeretnénk megoldani, hogy ha nem elérhető semmilyen hálózat, akkor ne induljon el a letöltés. Ehhez a Reachability nevű third-party modult fogjuk használni, amit a CocoaPods nevű keretrendszerrel fogunk hozzáadni a projekthez.

Először telepítsük/frissítsük a CocoaPods-ot. (**A laborgépeken nem lehetséges, mert adminisztrátori jogosultság kell hozzá!**)

Nyissuk meg a terminált és írjuk be a következő parancsot:

$ sudo gem install cocoapods

Majd adjunk hozzá egy üres fájt a projektünk gyökérkönyvtárába Podfile néven. Ez a legegyszerűbben úgy tudjuk megtenni, hogy a projekthez magához adunk hozzá egy üres fájlt.



A tartalma legyen a következő:

platform :ios, '8.0'

target 'PictureDownloader' do

pod 'Reachability', '~> 3.2'

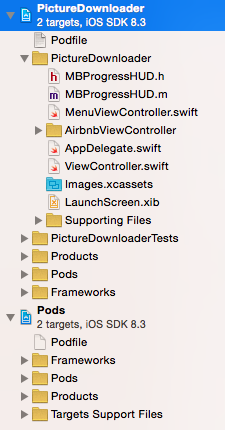
end

Mentsük el, majd menjünk vissza terminálhoz navigáljunk el a project gyökérkönyvtárába és írjuk be a következő parancsot:

$ pod install

Ha sikerült, zárjuk be a projektet és a Finderben keressük meg a projektünket. Látható, hogy a PictureDownloader.xcodeproj projektfájl mellett megjelent egy PictureDownloader.xcworkspace fájl is. Utóbbit nyissuk meg.

Valami ilyesmit kell látnunk.

Mivel az általunk választott Reachability modul szintén Objective-C-ben íródott, így adjuk hozzá a megfelelő header fájlt a bridging headerünkhöz.

#import "AirbnbHelper.h"

#import "MBProgressHUD.h"

#import <Reachability/Reachability.h>

Végül egészítsük ki a ViewController letöltés részét úgy, hogy csak akkor kezdődjön el a letöltés, amikor van internethozzáférésünk. Ehhez adjunk hozzá egy új függvényt az osztályhoz:

func startDownload()

{

let reach = Reachability(hostName: "google.com")

reach.reachableBlock = { reach in

dispatch\_async(dispatch\_get\_main\_queue(),{ [weak self] in

self!.downloadFile()

})

}

reach.startNotifier()

}

Legvégül módosítsuk a download gombot a ViewDidLoad-ban, hogy ezt a függvényt és ne közvetlenül a downloadFile-t hívja meg.

downloadButton.addTarget(self, action: "startDownload", forControlEvents: UIControlEvents.TouchUpInside)

## Önálló feladatok

1. Jelenítsük meg a letöltött képet valami kreatív módon!