

## Creating App on iTunes Connect

Add New App

Consumable

Non Consumable

Auto-Renewable Subscription

Non-Renewing Subscription

Reference Name

One Meal

Product ID

com.iapcourse.meal

# Sandbox Testers

↳ Users and Roles → Add Sandbox Tester

Hold shift and select all in storyboard to put in stack view

## IAPService.swift (Services)

import Foundation

import StoreKit

protocol IAPServiceDelegate {

func iapProductsLoaded()

}

class IAPService : NSObject, SKProductsRequestDelegate {

static let instance = IAPService()

var delegate : IAPServiceDelegate?

var products = [SKProduct]()

var productIds = Set<String>()

var productRequest = SKProductsRequest()

func loadProducts() {

productIDToStringSet()

requestProducts(forIDs: productIds)

}

func productIDToStringSet() {

productIds.insert(IAP\_MEAL\_ID)

}

func requestProducts(forIDs ids: Set<String>) {

productRequest.cancel()

productRequest = SKProductRequest(productIdentifiers: ids)

productRequest.delegate = self

productRequest.start()

}

func productsRequest(\_ request: SKProductsRequest, didReceive response: SKProductsResponse) {

self.products = response.products

if products.count == 0 {

requestProducts(forIDs: productIds)

} else {

delegate?.iapProductsLoaded()

}

3

## //FoodItems (Helpers)

```
import Foundation
```

```
let defaultPrice = 9.99
```

```
let salmon = Item (image: UIImage(named: "food1")!, name: "Salmon", price: defaultPrice)
```

↓

```
let foodItems: [Items] = [salmon, ...]
```

## // Constants.swift (Helpers)

```
import Foundation
```

```
let IAP_MEAL_ID = "com.iapcourse.meal"
```

## Item.swift (Model)

```
import UIKit
```

```
class Item {  
    public private(set) var image  
    public private(set) var name  
    public private(set) var price
```

```
init (image: UIImage, name: String, price: Double) {
```

```
    self.image = image
```

```
    self.name = name
```

```
    self.price = price
```

```
}
```

```
3
```

## // ItemCell.swift (view)

import UIKit

```
class ItemCell: UICollectionViewCell {
    @IBOutlet weak var itemImageView: UIImageView!
    @IBOutlet weak var itemNameLbl: UILabel!
    @IBOutlet weak var itemPriceLbl: UILabel!

    func configureCell(forItem item: Item) {
        itemImageView.image = item.image
        itemNameLbl.text = item.name
        itemPriceLbl.text = String(describing: item.price)
    }
}
```

## 11 StorefrontVC.swift (Controller)

```
import UIKit
```

```
class StorefrontVC: UIViewController, UICollectionViewDelegate, UICollectionViewDataSource {
```

```
@IBOutlet weak var collectionView: UICollectionView!
```

```
override func viewDidLoad() {
```

```
    super.viewDidLoad()
```

```
}
```

```
func numberOfSections(in collectionView: UICollectionView) -> Int {
```

```
    return 1
```

```
}
```

```
func collectionView(_ collectionView: UICollectionView, numberOfItemsInSection section: Int) -> Int {
```

```
    return foodItems.count
```

```
}
```

```
func collectionView(_ collectionView: UICollectionView, cellForItemAt indexPath: IndexPath) -> UICollectionViewCell {
```

```
    guard let cell = collectionView.dequeueReusableCell(withReuseIdentifier: "itemCell", for: indexPath)  
        as? ItemCell else { return UICollectionViewCell() }
```

```
    let item = foodItems[indexPath.row]
```

```
    cell.configureCell(forItem: item)
```

```
    return cell
```

```
}
```

```
func collectionView(_ collectionView: UICollectionView, didSelectItemAt indexPath: IndexPath) {
```

```
    guard let detailVC = storyboard?.instantiateViewController(withIdentifier: "DetailVC") as? DetailVC else {  
        return  
    }
```

```
    let item = foodItems[indexPath.row]
```

```
    detailVC.initData(forItem: item)
```

```
    present(detailVC, animated: true, completion: nil)
```

```
}
```

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

```
}
```

## // Detail VC. Swift (controller)

\* set the stack view to fill alignment

```
import UIKit
```

```
class DetailVC: UIViewController {  
    @IBOutlet weak var itemImageView: UIImageView!  
    @IBOutlet weak var itemNameLabel: UILabel!  
    @IBOutlet weak var itemPriceLabel: UILabel!  
    @IBOutlet weak var uglyAdView: UIView!  
    @IBOutlet weak var buyItemBtn: UIButton!  
    @IBOutlet weak var hideAdsBtn: UIButton!
```

```
public private(set) var item: Item!
```

```
func initData(forItem item: Item) {  
    self.item = item  
}
```

```
override func viewDidLoad() {  
    super.viewDidLoad()  
  
    itemImageView.image = item.image  
    itemNameLabel.text = item.name  
    itemPriceLabel.text = String(describing: item.price)  
    buyItemBtn.setTitle("Buy this item for $\((item.price))", for: .normal)  
}
```

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

```
}
```

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

```
}
```

```
@IBAction func closeBtnWasPressed(_ sender: Any) {  
    dismiss(animated: true, completion: nil)
```

```
}
```