

## 内容回顾

### 一. 初始化项目, 并且进行基本设置

- 部署版本
- 支持屏幕方向
- 应用程序图标/启动图片
- 分目录结构

### 二. 给项目添加GIT管理

- `git init` -> 初始化一个本地仓库
- `touch .gitignore` -> 添加忽略文件
- 将项目添加到git管理之下

### 三. 搭建项目的框架

#### 1. 启动控制器是一个UICollectionViewController

#### 2. 自定义布局

- `HomeCollectionViewLayout : UICollectionViewFlowLayout`
- 重写 `prepareLayout`
- `itemSize`
- `min/min`
- 设置内边距

### 四. 封装AFNetworking

#### 1. 集成Cocoapods, 并且导入AFNetworking

#### 2. 封装工具类, 并且将工具栏设计成单例对象

- 两种单例
- 通过某种方式一直拿到一个实例
- 不管如何创建拥有只有一个实例

#### 3. 封装请求方法

- `request(type, urlString, parameters, 闭包)`

### 五. 发送网络请求, 请求首页数据

- 在控制器中封装请求方法
- 将请求方法, 写在NetworkTools
- 将请求到数据回调到控制器

### 六. 解析请求到数据

- 解析到的一个字典
- 封装Shop模型: 1. 添加属性 2. 自定义构造函数
- 字典转模型, 将模型对象放入到shops数组
- 刷新表格

## 七.展示数据

### 1.自定义Cell

- 在cell添加shop属性

### 2.将shop对象传给cell

### 3.给cell添加imageView子控件

### 4.监听shop属性的改变

- 给imageView设置图片

## 八.上拉加载更多

### 1.在cellForItemAtIndexPath方法中监听是否最后一个cell出现方请求最新的数据

### 2.请求最新的数据时,需要传入offset

## 九.弹出图片浏览器

- 监听Cell点击
- 创建图片浏览器,并且将它弹出

## 十.布局图片浏览器

### 1.布局collectionView

- 创建collectionView
- 设置frame
- 设置数据源
- 注册cell
- 自定义布局

### 2.布局关闭和保存按钮

- 便利构造函数: 1.前面加载convenience 2.需要self.init()
- 监听按钮的点击: Selector : 1. Selector("") 2."" 3.#Selector(方法名称)

### 3.给图片浏览器设置数据

- 将shops数组/indexPath传递到控制器中
- 自定义Cell
- 在cell中添加imageView
- 监听shop的改变: 1.获取要显示的小图 2.根据小图计算imageView的尺寸 3.设置大图

## 十一.细节调整

### 1.调整图片之间的间距

- view.frame.size.width += 15

### 2.保存图片

- 拿到正在显示的Cell `let cell = collectionView.visibleCells[0] as! Cell`
  - `cell.imageView.image`
  - `UIImageWriteToDisk(image)`
3. 点击cell, 关闭控制器
- 监听cell的点击
4. 让图片浏览器滚到到对应的位置

## 十二. 自定义弹出的动画

- `modalTransitionStyle`