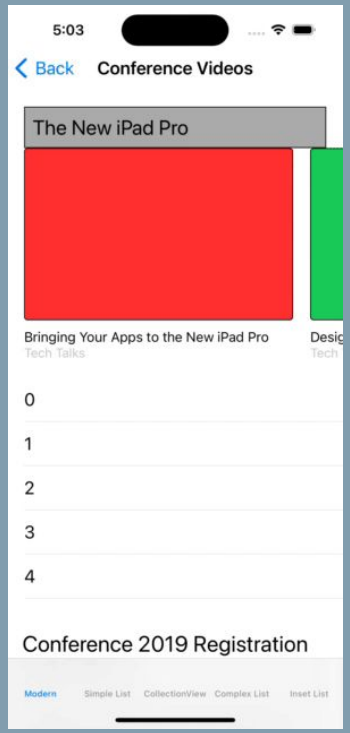
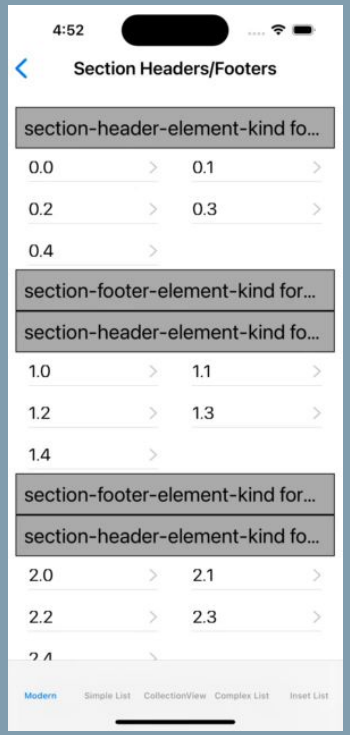
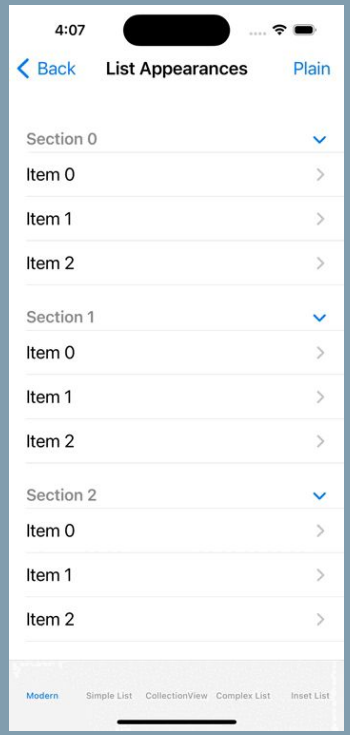
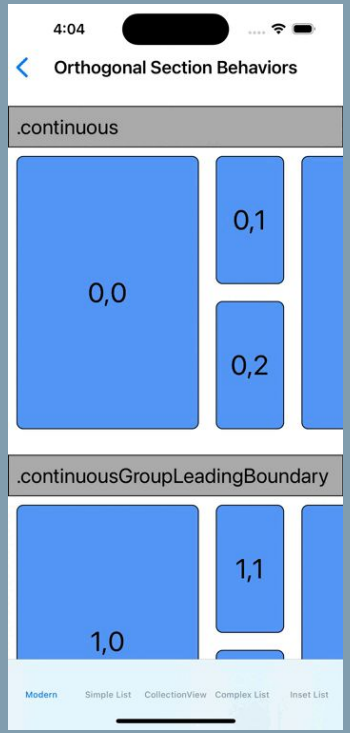
IQListKit

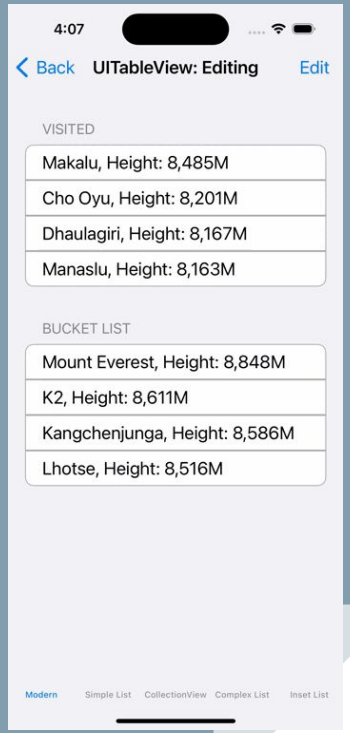
Modern CollectionView Layouts





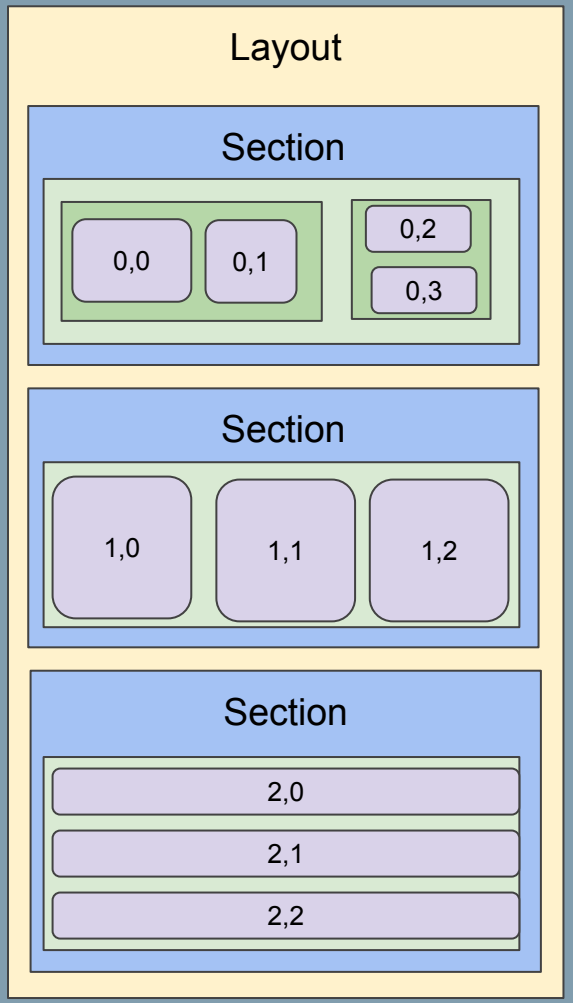






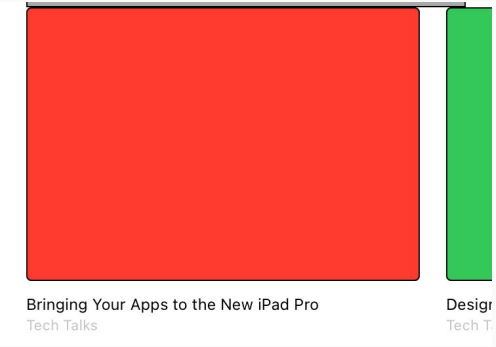
Compositional Layout

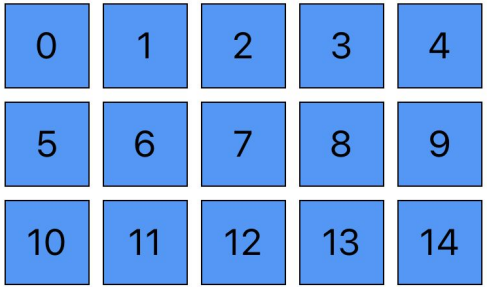
- Layout
  - Section
    - Group
      - (sub Groups)
        - Item

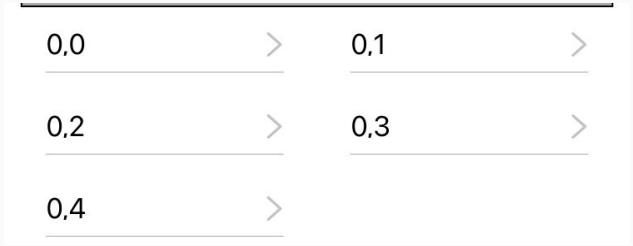


Item (NSCollectionLayoutItem)

- You can think it like a cell. UICollectionViewCell or UICollectionViewListCell (UITableViewCell style)
- It has a size width and height (NSCollectionLayoutSize)
  - **`fractionalWidth(0.5)`** proportionally
  - **`fractionalHeight(0.5)`** proportionally
  - **`absolute(44)`** fixed
  - **`estimated(50)`** an estimate of size
- **`contentInsets: NSDirectionalEdgeInsets`**
- **`edgeSpacing: NSCollectionLayoutEdgeSpacing`**

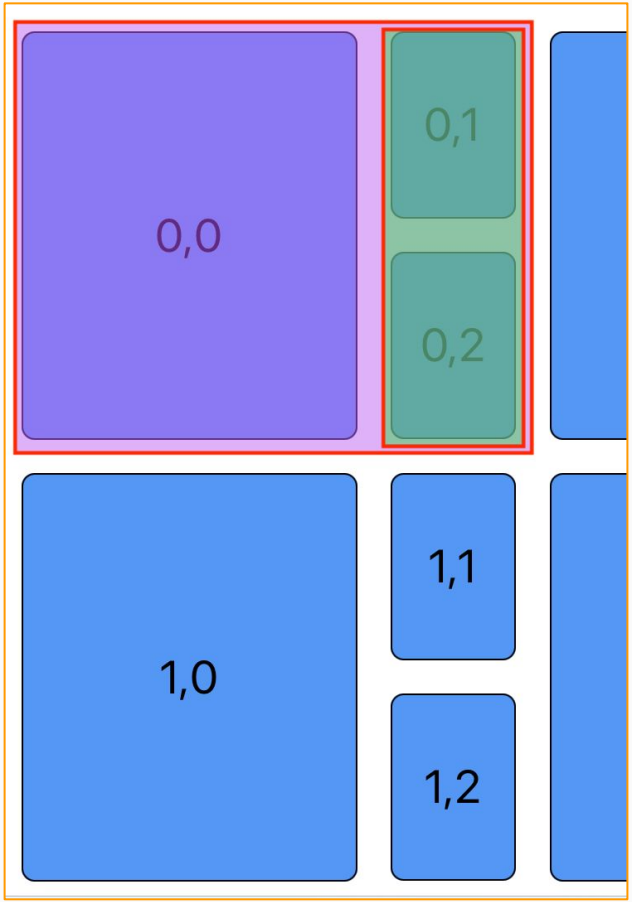






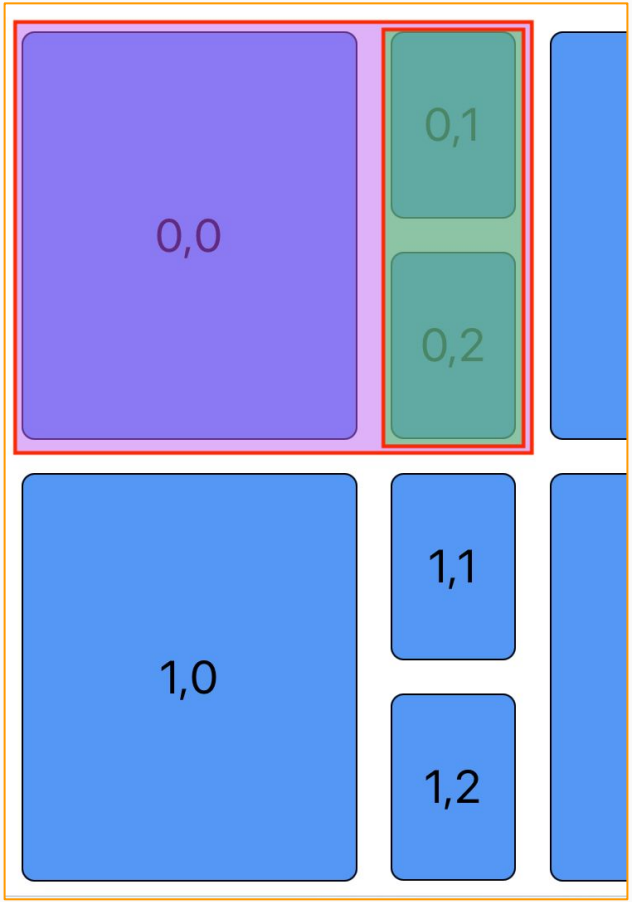
Group (NSCollectionLayoutGroup: NSCollectionLayoutItem)

- It’s a direct subclass of NSCollectionLayoutItem
- Group can collection multiple items and another group as well.
  - **`subitems`**
- It’s either horizontal or vertical
- **`layoutSize: NSCollectionLayoutSize`**
- **`contentInsets: NSDirectionalEdgeInsets`**
- **`edgeSpacing: NSCollectionLayoutEdgeSpacing`**
- **`interItemSpacing: NSCollectionLayoutSpacing`**
-



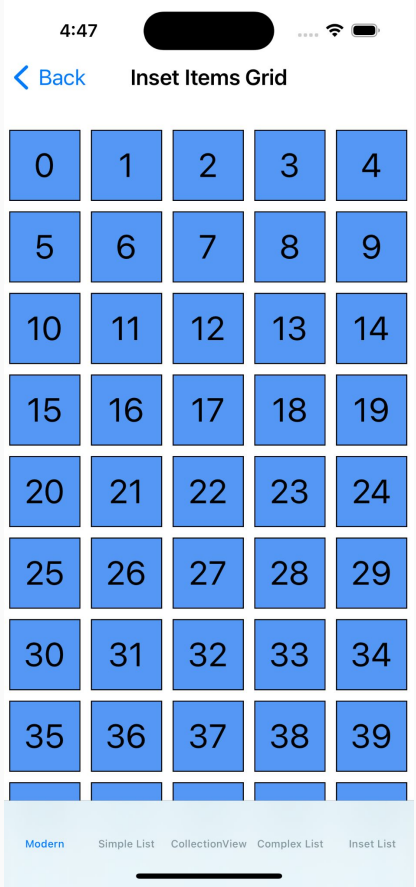
Section (NSCollectionLayoutSection)

- Section contains single group which will be repeated according to the number of items in a section.
- `contentInsets: NSDirectionalEdgeInsets`
- `interGroupSpacing: CGFloat`
- `orthogonalScrollingBehavior: UICollectionLayoutSectionOrthogonalScrollingBehavior`
  - `none`
  - `continuous`
  - `continuousGroupLeadingBoundary`
  - `paging`
  - `groupPaging`
  - `groupPagingCentered`



Layout (UICollectionViewLayout)

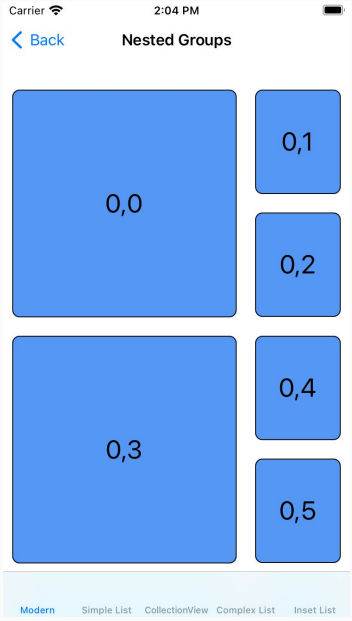
- It’s the base layout which knows how exactly the layout needs to be rendered or displayed in screen.
- One of the default layout we already know is
  - **`UICollectionViewFlowLayout`**
- Now we’ll be discussing another subclass introduced in iOS 13 is **`UICollectionViewCompositionalLayout`**

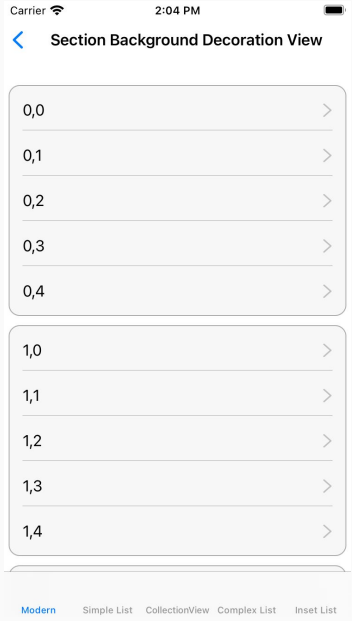


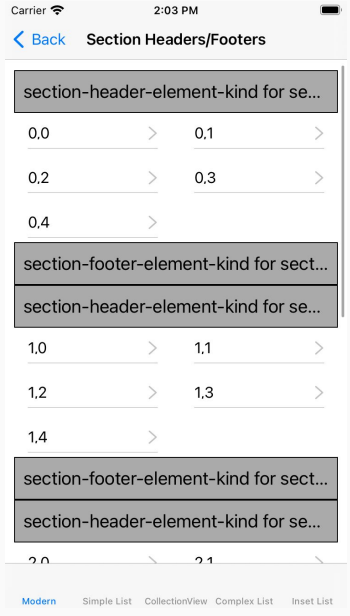
Layout (UICollectionViewCompositionalLayout)

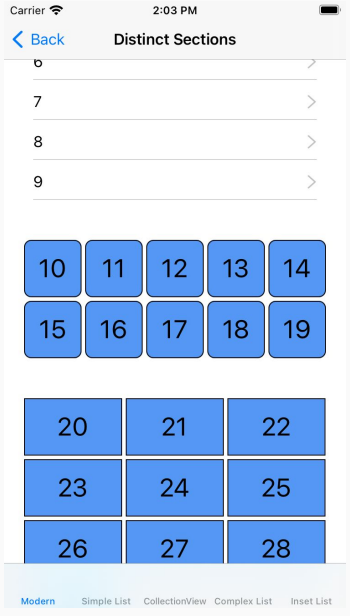
- There are 2 ways we can compose the compositional layout

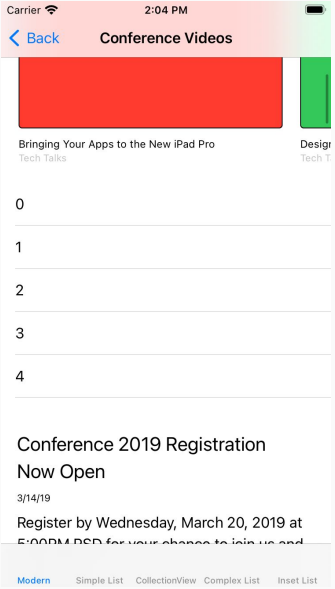
1) Provide a **`NSCollectionLayoutSection`** to repeat in each section of collectionView
2) Using a section provider completion handler using which we can return different **`NSCollectionLayoutSection`** for different section of collectionView







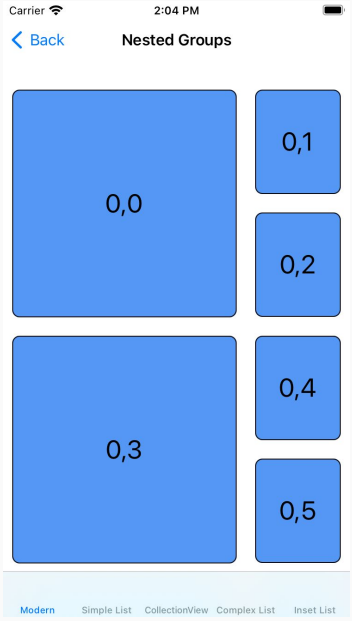


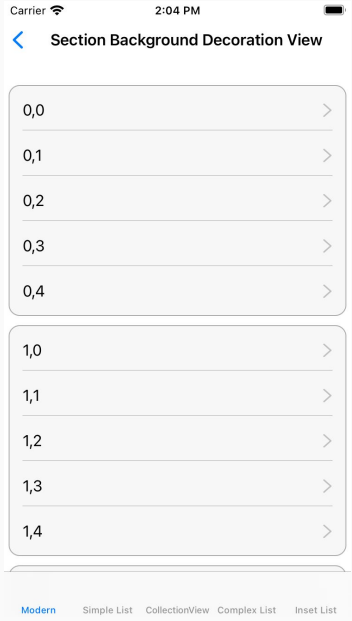


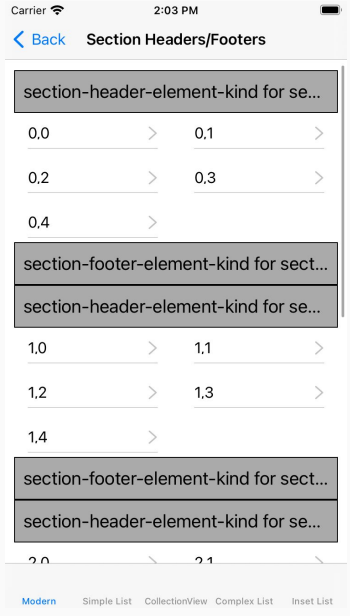
Layout (UICollectionViewCompositionalLayout)

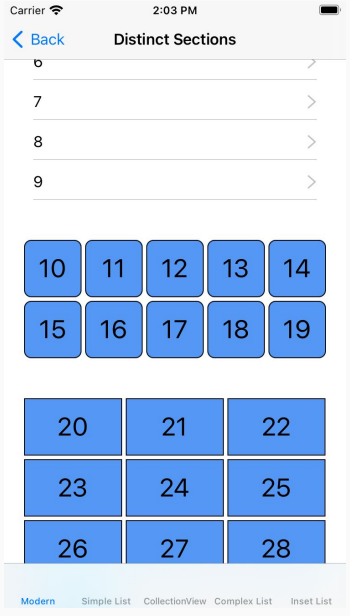
- There are 2 ways we can compose the compositional layout

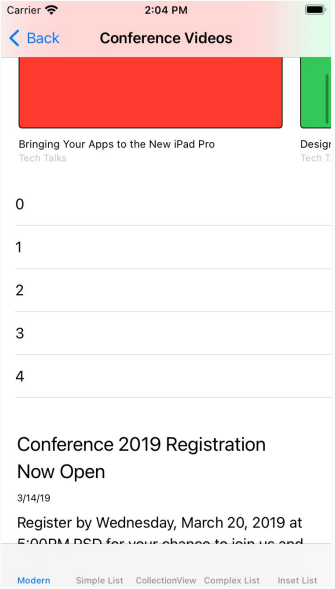
1) Provide a **`NSCollectionLayoutSection`** to repeat in each section of collectionView
2) Using a section provider completion handler using which we can return different **`NSCollectionLayoutSection`** for different section of collectionView









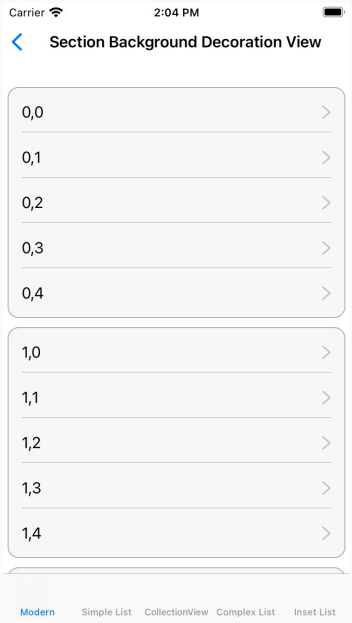


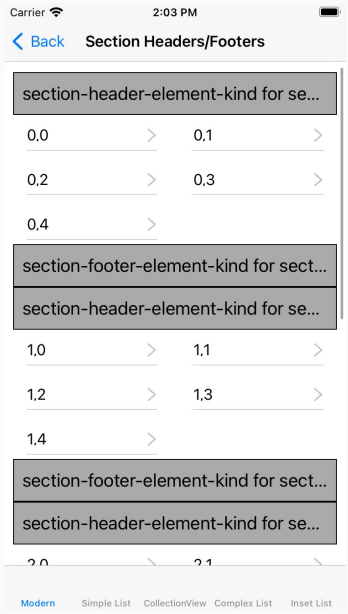
Special List Layout (UICollectionViewCompositionalLayout)

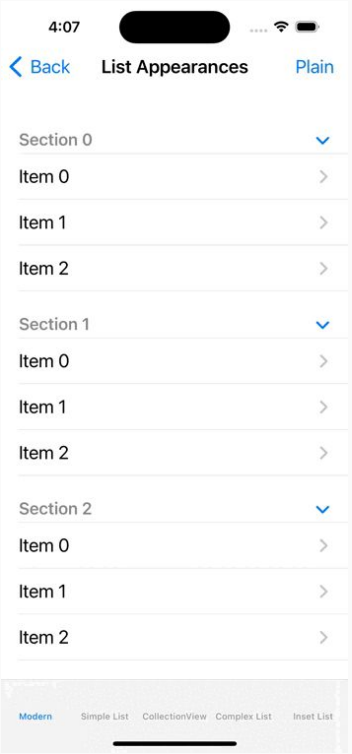
- There is one special kind of layout called list available since iOS 14 which looks and behaves like UITableView but composed using a UICollectionView and a special layout.

```
let config = UICollectionLayoutListConfiguration(appearance: .plain)
let layout = UICollectionViewCompositionalLayout.list(using: config)
```

There is special kind of Cell **`UICollectionViewListCell`**







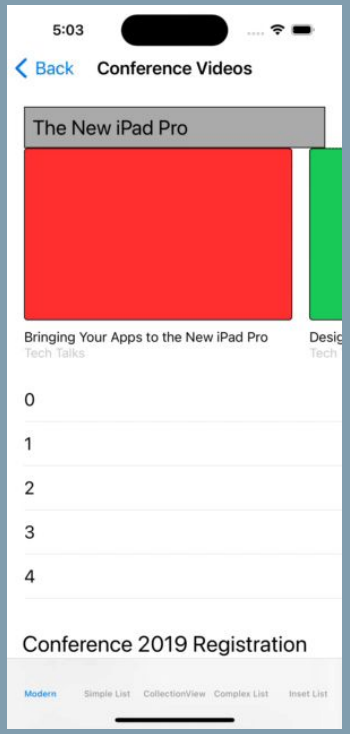
But I came here for IQListKit

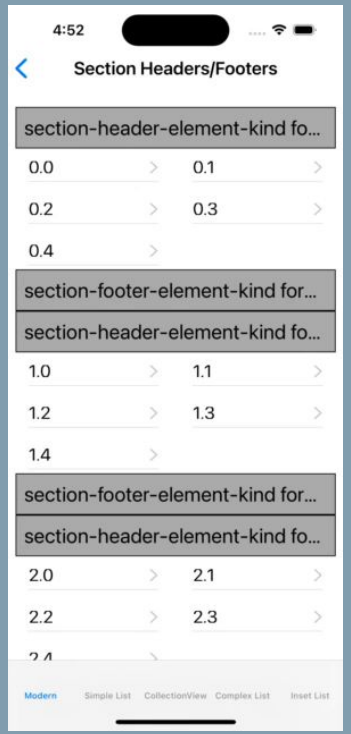
What the hell is going on with these layouts.
Show me the code?

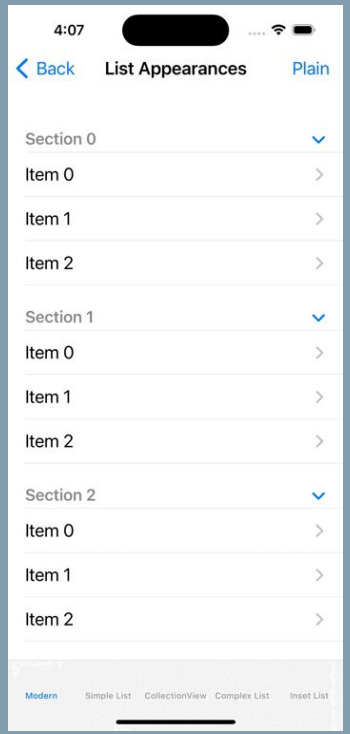
Congratulations

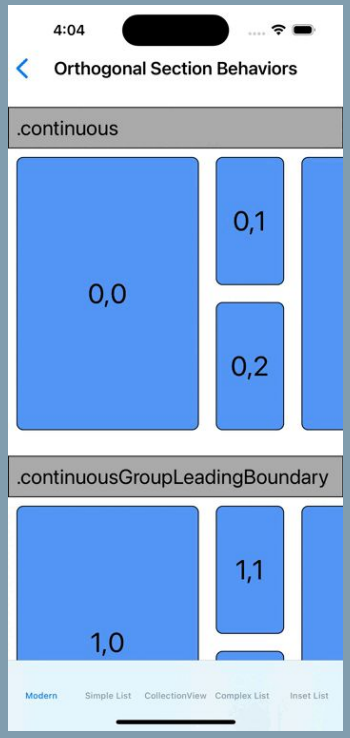
IQListKit still works out of the box in the same way with these new layouts.

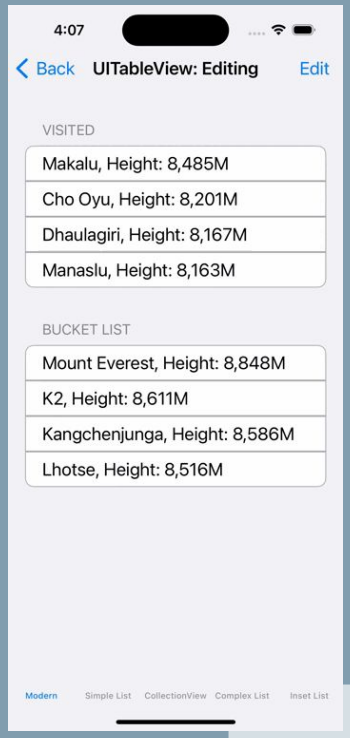
You still just have to provide Cell Type and it's Model like before











In 90% of the cases, you don't have to modify or change any existing IQListKit code to switch to modern layouts.

So what exactly I need to do?

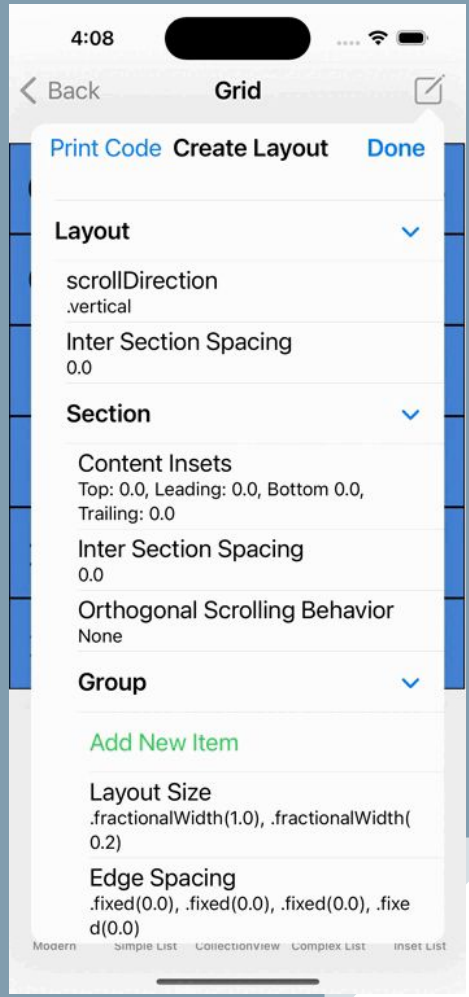
Well, you need to provide correct layout you would like to render.

But I still don’t understand much about how these layout works?

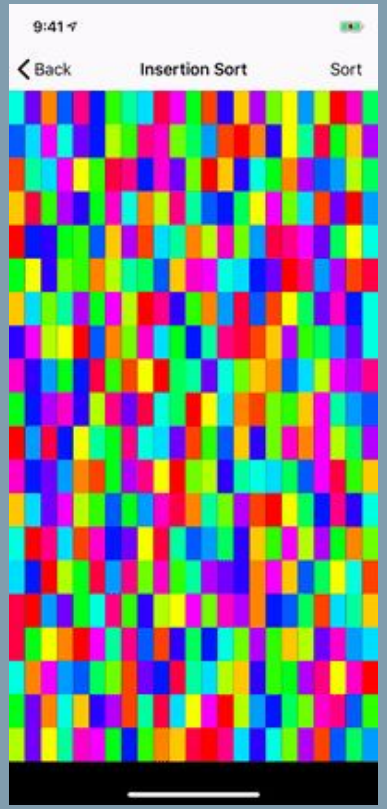
To solve this problem, I have created a Live Layout Editor to help you out on this matter.

Layout Editor

```
----------Layout Code Start----------

var itemLayoutSize: NSCollectionLayoutSize = NSCollectionLayoutSize(widthDimension:
.fractionalWidth(0.2), heightDimension: .fractionalHeight(1.0))
let item = NSCollectionLayoutItem(layoutSize: itemLayoutSize)
var groupLayoutSize: NSCollectionLayoutSize =
NSCollectionLayoutSize(widthDimension: .fractionalWidth(1.0), heightDimension:
.fractionalWidth(0.2))
let group: NSCollectionLayoutGroup = NSCollectionLayoutGroup.vertical(layoutSize:
groupLayoutSize,
                                                                    subitem: item,
                                                                    count: 6)


let section = NSCollectionLayoutSection(group: group)
section.contentInsets = .init(top: 5.0, leading: 5.0, bottom: 5.0, trailing: 5.0)
section.orthogonalScrollingBehavior = .paging

let configuration: UICollectionViewCompositionalLayoutConfiguration = .init()
configuration.scrollDirection = .horizontal
let layout = UICollectionViewCompositionalLayout(section: section, configuration:
configuration)

----------Layout Code End----------
```



Let’s Have a Demo



Download the github project to see it in action

Questions, suggestions and improvements can be contributed through github issues

https://github.com/hackiftekhar/IQListKit