

添加一个新类型cell

实现新的cell
NewCell

解析新的数据，添加到
tableView的数据源里面

修改tableView的代理方法
cellForRow
heightForRow
...

如果添加的是一个
section，还得处理
numberOfRowsInSection

清理cell的时
候也要都清理
一遍

添加一个新类型cell

实现新的cell
NewCell
实现对应的接口

解析新的数据，生成新的
view model添加到
tableView的数据源里面

清理的时候只
要清理数据源

好处

- 适应经常变化的首页UI
- 用数据驱动UI
- 可重用性提升
- cell跟view controller解耦

几个点

- cell注册
 - 在vc中引入cell头文件注册。
 - 在cellForRow方法中注册，存在反复注册的情况，每次调cellForRow方法都会执行注册代码。
- cell高度返回
- 现有cell支持，cell添加category实现接口或直接cell实现接口，给cell添加一个view model。
- section header 的解决方式
 - 做成cell的形式
 - 在header的代理方法里面处理

- 分隔线的解决方法

协议说明

- YTCellProtocol

table view的cell都要实现这个接口，接口提供配置数据的方法和注册cell的方法。

```
1.  /**
2.   cell 的接口
3.   提供一个配置cell的ViewModel的方法
4.   */
5.   @protocol YTCellProtocol <NSObject>
6.
7.   @required
8.
9.
10.  /**
11.   通过 view model 来配置cell, table view 的数据源里面装的都会是 view model
12.
13.   @param viewModel: cell 对应的 view model
14.   */
15.  - (void)configCellWithViewModel:(id)viewModel;
16.
17.
18.  /**
19.   给table view 注册cell 方便复用，不用知道cell是用xib的形式，还是代码的形式实现的
20.
21.   @param table 要注册的table view
22.   */
23.  + (void)registerFor:(UITableView *)table;
24.
25.
26.  @end
```

- YTTableViewCellViewModelProtocol

table view cell 对应的view model 都要实现这个接口，实现返回cell高度的方法和cell的复用id

```

1.  /**
2.   UITableViewCell 对应的ViewModel 的协议
3.   */
4.  @protocol YTTableCellViewModelProtocol <NSObject>
5.
6.  @required;
7.  /**
8.   返回cell的高度，这个方法是在view model中实现，view model中有cell的全部数据，所以这里可以通过数
9.
10.   @return cell的高度
11.   */
12.  - (CGFloat)cellHeight; // 返回cell的高度
13.
14.
15.  /**
16.   返回cell的复用id
17.
18.   @return cell的复用id
19.   */
20.  + (NSString *)identifier;
21.
22.  @end

```

Section类说明

```

1.  @interface TableViewSection : NSObject
2.
3.
4.  /**
5.   用来标识section的类型，要保证每个section的都不同
6.   代理方法里面如果要做特殊处理会用到，所以不能相同
7.   */
8.  @property (nonatomic, copy) NSString *sectionKey;
9.
10.
11.  /**
12.   section 里面 row 的 view model 集合
13.   */
14.  @property (nonatomic, copy) NSArray *viewModels;
15.
16.
17.  /**
18.   返回这个section有多少row，这个不用设置，在设置viewModels时，会自动设置
19.   */
20.  @property (nonatomic, readonly) NSInteger numberOfRows;
21.
22.  @end

```

```
1. @implementation TableViewSection
2.
3. - (instancetype)initWithSectionKey:(NSString *)key viewModels:(NSArray *)viewModels;
4.     self = [super init];
5.     if (self) {
6.         _sectionKey = key;
7.         _viewModels = viewModels;
8.         _numberOfRows = viewModels.count;
9.     }
10.
11.     return self;
12. }
13.
14. @end
```