MVC大家族

数据绑定 / MVC / MVP / MVVM / MVW

汪志成 (wangzhicheng@admaster.com.cn)

2014年5月19日

MVC是什么?

- 它不是架构!
- 它不是框架!
- 它是软件架构模式!

软件架构是什么?

软件架构是有关软件整体结构与组件的抽象描述

- 超高的视角有其不可替代的价值
- 人类用了很久很久才知道地球有七大洲
- 但假如有月球人的话，他们早就知道了

架构对系统的长期演化有着深刻的影响

- 根据需求，谨慎的选择架构

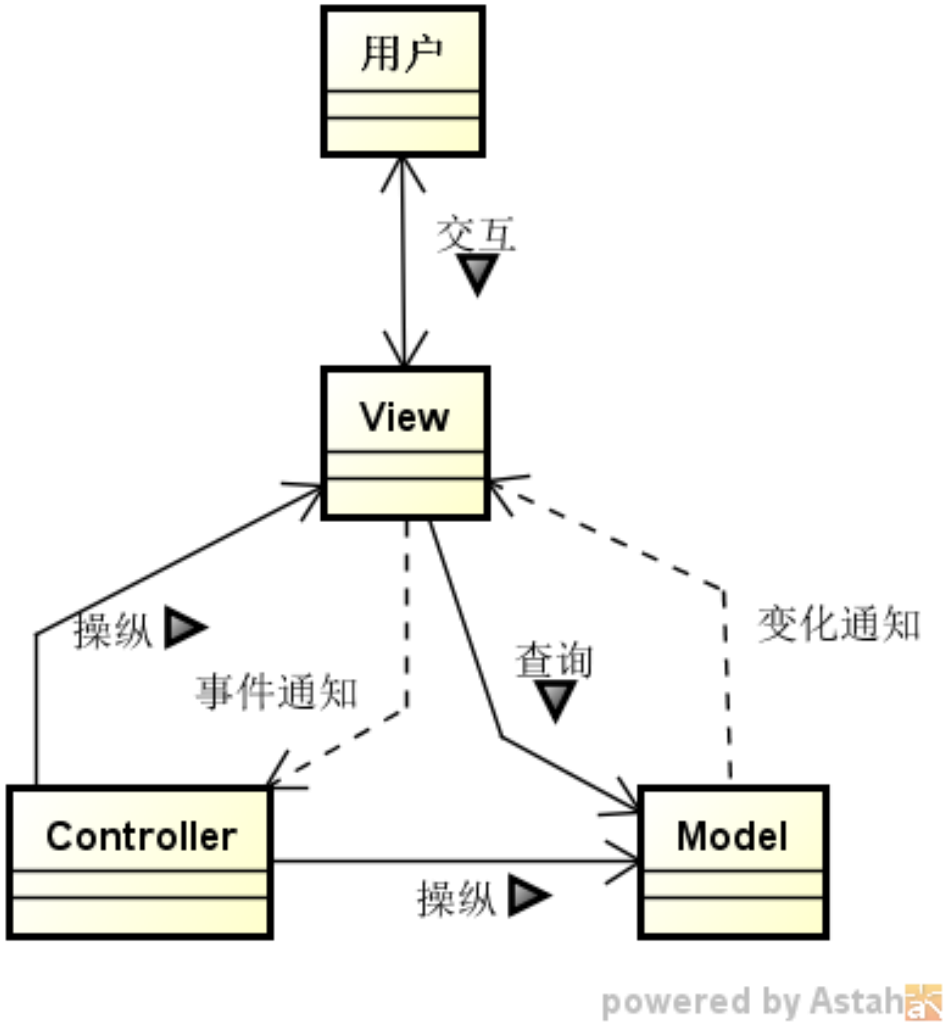
架构主要用来解决非功能需求

- 扩展性、延展性、安全性、用户友好性等等

想象一个场景场景

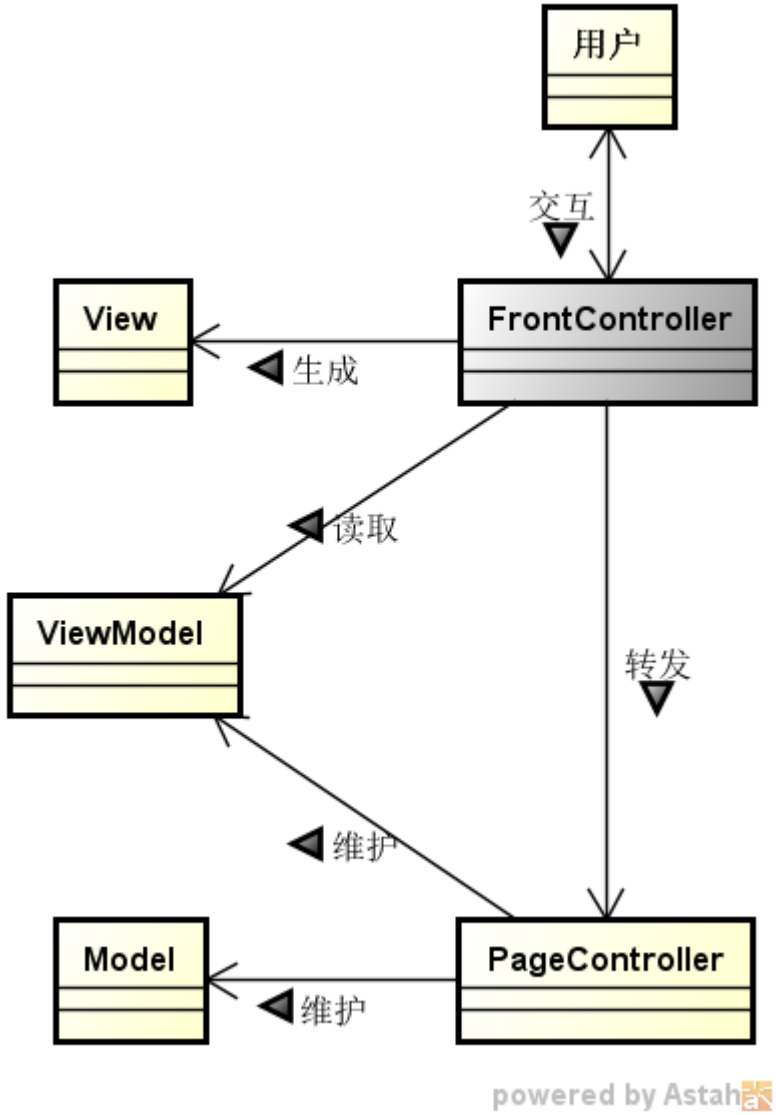
- 显示商品列表
- 如果单价>1500元则显示为红色
- 这个显示规则短期内不会变化，但是将来肯定会变化

经典MVC



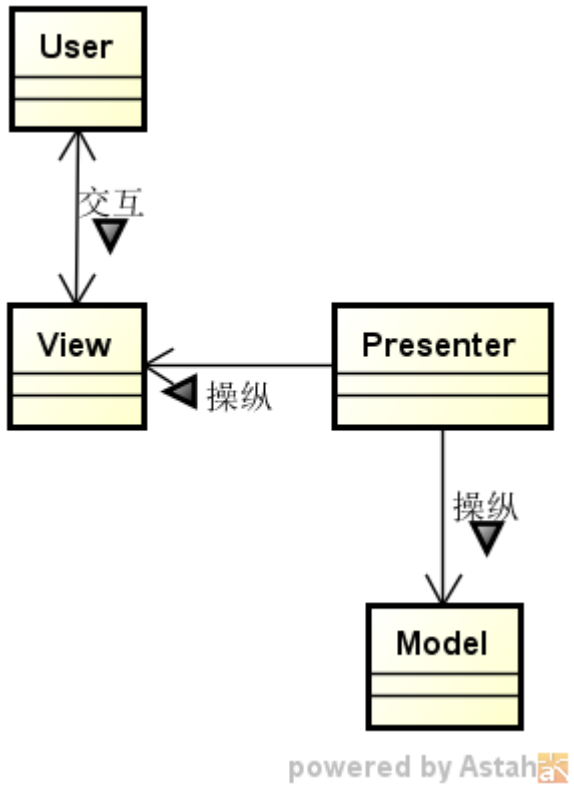
- 视图需要访问Model，而Model的数据和视图需要的数据之间往往不匹配，该改谁呢？
- Controller既需要操纵View又需要操纵Model，内聚性不够
- View中如果写查询逻辑，如何对这部分代码进行自动化测试？

Web MVC（Spring MVC为例）



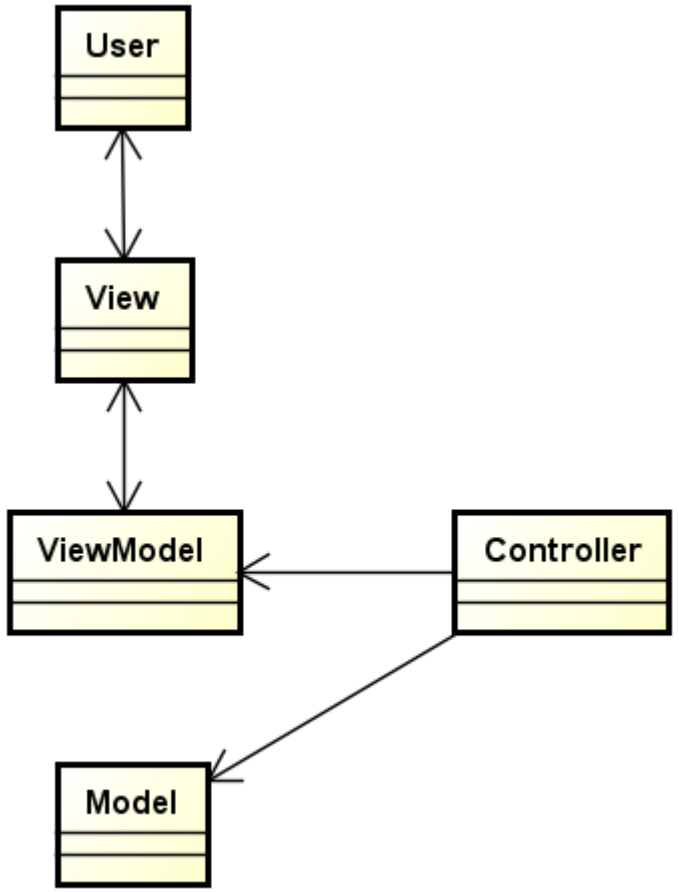
- 引入了ViewModel
- 分成了两种Controller
- View变“瘦”，测试压力减小

MVP（被动模式）



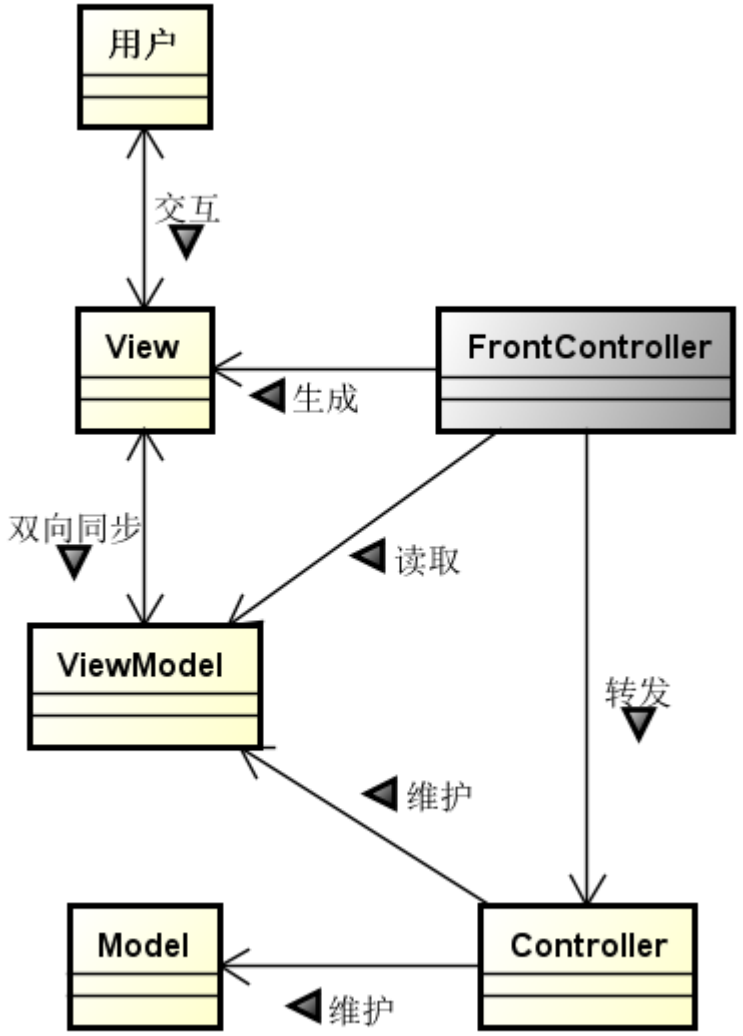
- 便于从过程式编程跨越过来
- View变瘦，测试压力减小
- Presenter内聚性不够

MVVM



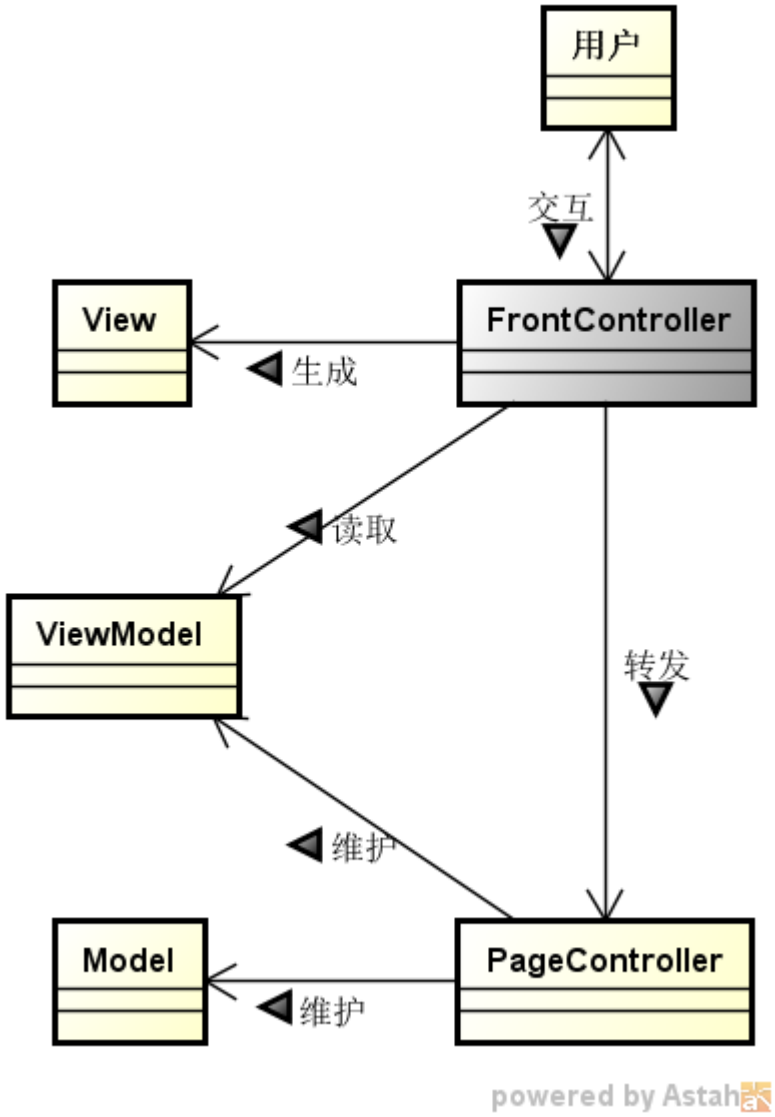
- View和ViewModel自动同步
- Controller再也看不到View了
- Model和View之间的数据不匹配也解决了

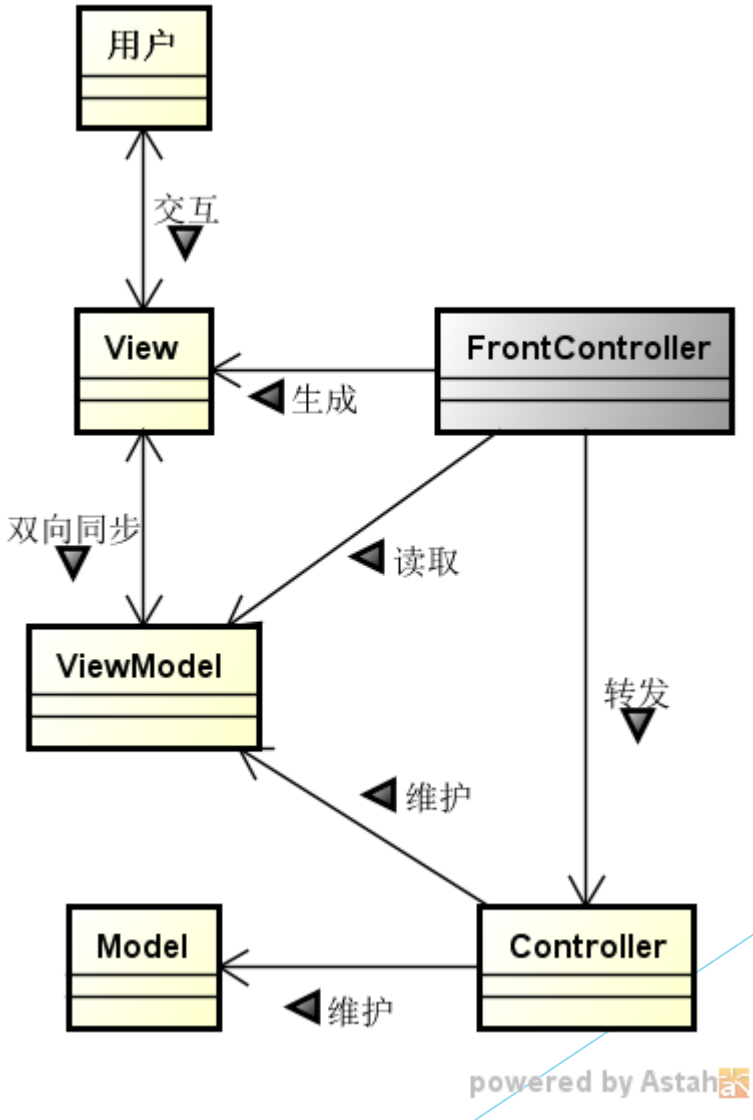
Angular的架构概览（隐藏掉了细节）



- View很“瘦”/定义式界面
- Controller看不到View
- 内部实现了前端控制器
- 前端控制器负责双向同步

Spring MVC vs Angular





MVC的发展趋势（个人看法）

提高内聚性	• 把操纵View的逻辑和操纵Model的逻辑分开 • 公用的逻辑尽量移入框架来实现
被动式View	• 描述式界面，测试的需求变小了 • 对编程技能要求低，便于分工
明确引入ViewMode	• 解决数据适配的问题 • 解耦视图和数据访问层

MVW —— Google式恶搞

- MVW = Model-View-Whatever
- 万变不离其宗
- 高内聚，低耦合
- 信息隐蔽原则

分享完毕，谢谢！