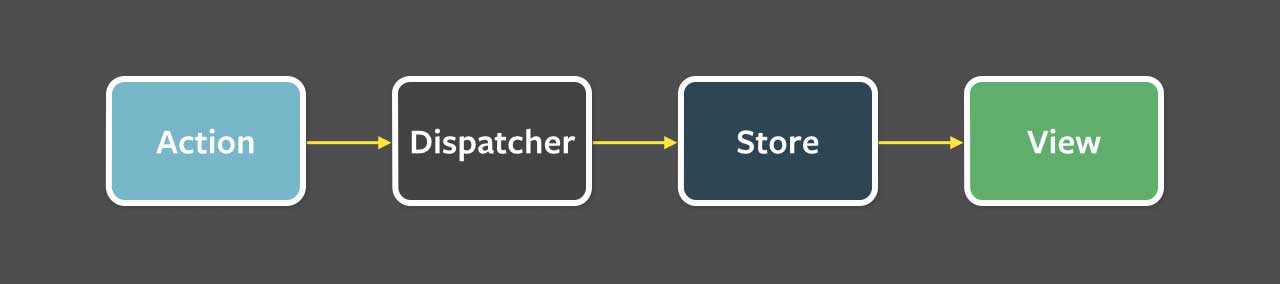
**React Flux入门的一些理解**

**题记**：Flux是Facebook用于构建客户端web应用程序的应用架构。它通过单项数据流模式补充了React单纯的view组件化。Flux更多的市一中模式而不是一个正式的框架。

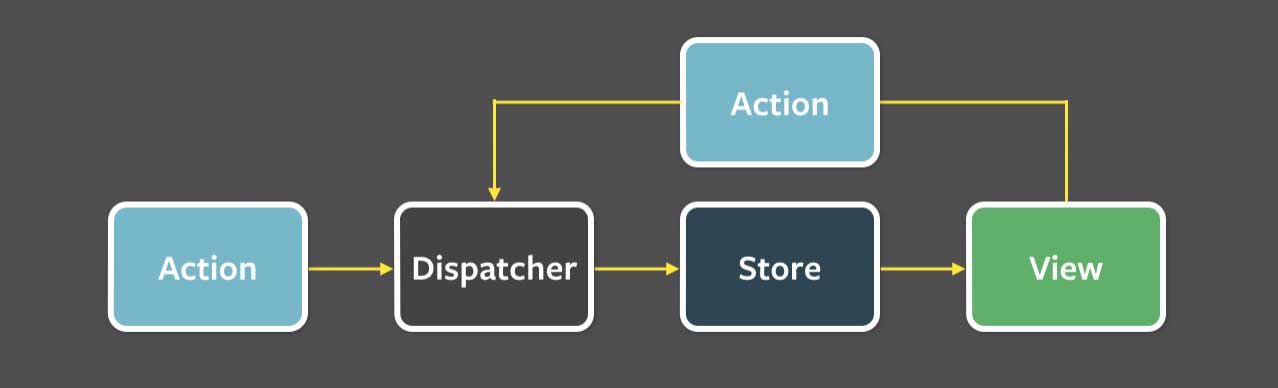
Flux是一种模式，来描述**单向数据流**。

Flux主要包括三个部分：dispatcher（调度器），store（商店）和view（React的组件）。

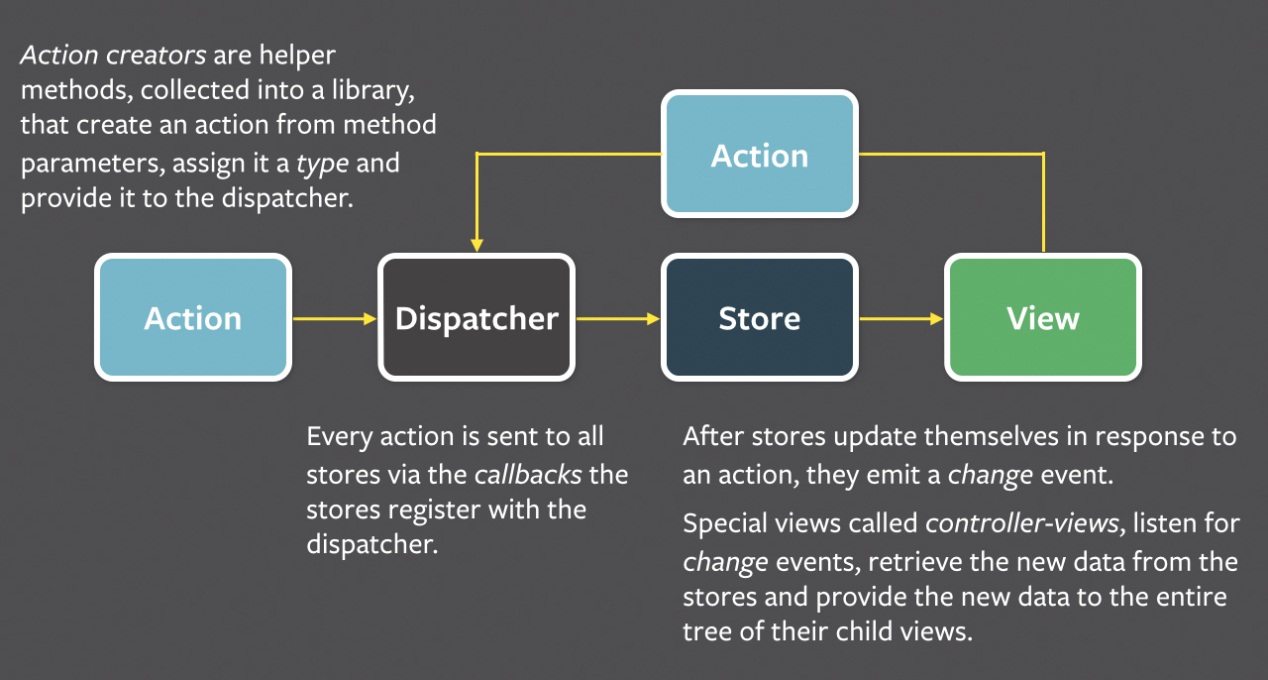


上面这张图是使用Flux的首要思考模型。ispatcher，store和view是有着特定输入输出的独立节点。

view根据用户的交互动作有可能会产生一个新的action：



Dispatcher像一个中心枢纽的调度器。所有数据的流动都要通过这里。Action来源于用户与views的交互行为，Action触发Dispatcher。Dispatcher派发这个事件给对应的Store（通过之前注册好的回调函数callback）。Store在修改State后触发一个“change”事件通知controller-views数据发生变化了。controller-views监听这些“change”事件并且从stores暴露的函数中获取（新）数据，然后调用自己的setState()方法，reRender自己和它的子组件。



## A Single Dispatcher

Dispatcher是Flux模式中管理所有数据流的中心枢纽。它本质上就是一些Store回调函数的注册器，它本身没有其他逻辑。只是提供了把Action分发给Store的机制。（我的理解是：dispatcher根据action type调用对应的回调函数）。每一个Store都在Dispatcher注册并提供回调函数。

随着应用的发展，Dispatcher会变得越来越重要。例如Dispatcher可以用来管理Stores之间的依赖关系，通过特定的顺序来调用注册了的回调函数就可以办到。Stores可以等到其他Stores完成更新再进行自己的更新操作。

## Stores

Store内有这个应用的State和对应的逻辑。他的角色在一定程度上和传统MVC结构中的model相似。

一个store陈列了多种模式集合和单一逻辑范围模式的特性。

在应用中，我们需要特定的处理逻辑和数据集合来存储列表

https://xinranliu.me/2015-01-30-some-understanding-about-flux/