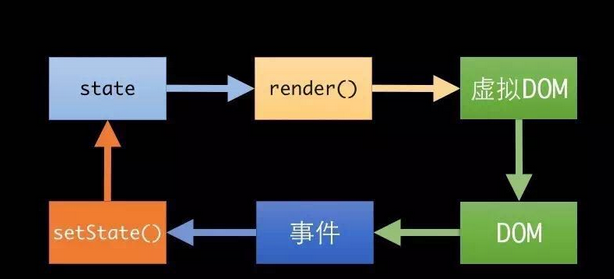
React和Backbone的对比分析

## 对于DOM节点的操作对比

### React

React的本质是一个带有作用域的自定义的DOM元素。React首创的虚拟DOM，封装了对于真实DOM的操作，react框架帮助我们在组件的props或者state发生变化时，对比重新构造的虚拟DOM和真实DOM的差别，对有变化的DOM节点进行更新操作，完成渲染功能。对于开发者来说，不用涉及到对真实DOM节点的操作，提升了开发效率，只需要对需要更新的组件进行值的传递。React采用了html和js结合的方式，将html和js封装成组件类，由render返回最终的组件虚拟DOM节点。



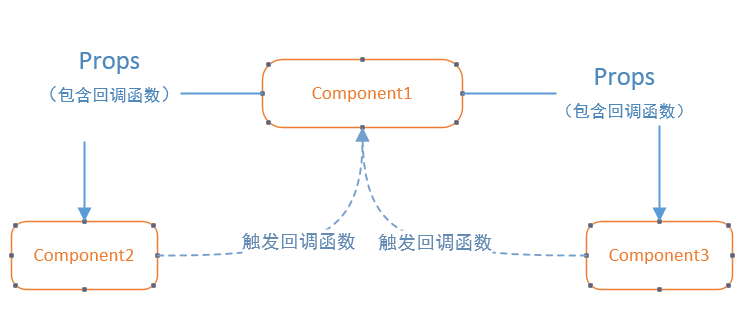
### Backbone

Backbone遵循 MVC的设计模式，有Model负责业务数据的管理，由View进行视图的渲染。比价好的隔离了数据和视图。在视图操作中，采用构建模板和模板填充的方式，由View对Model进行监听变化，借助模板引擎调用View层的render方法就可实现对界面的重新渲染。从这个方面来看，Backbone也采用了数据驱动式的DOM更新操作，避免了原有人工操作DOM的繁琐。对比React的虚拟DOM，Backbone对真实DOM进行操作。

## 数据流的传递

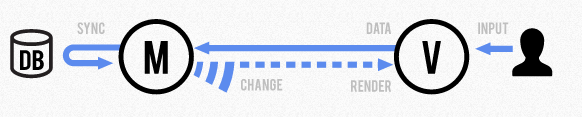
### React

React采用的是单向数据流，父组件通过向子组件传递props的方式完成对子组件的更新操作，而无法由子组件向父组件直接通信。针对子组件对父组件通信、兄弟组件通信，采用的是由父组件向子组件传递回调函数（以props的形式），由子组件触发该函数，进而触发父组件进行更新；兄弟组件（具有同一个父组件）在子父组件通信的基础上由父组件的回调函数传递新的props给子组件的兄弟组件，进而完成兄弟组件的状态更新。这种回调函数的做法本质上也是由父组件向子组件传递props的单向数据传递。这一做法的好处是确保了数据驱动，使得数据的走向一目了然，方便追踪问题。



### Backbone

Backbone中数据流的传递是双向的，主要体现在View模块和Model进行关联，监听Model的变化来实时更新渲染Html；通过自身注册的事件将UI的变化同步更新到Model。View就像一个控制器，同步界面UI和数据的实时变化。而关于数据的获取、更新等操作则统一交由Model模块进行。



以上React和Backbone在数据流传递方式的区别，根本原因在于Backbone比较完整地涵盖了MVC的设计模式，而React则是只包含了MVC中的V。Backbone框架比较完整地包含了网站搭建从前端界面到后台数据；而React则是专注于界面UI的用于创建可复用、可聚合的web component的js库，并不是完整地MVC框架。

## 渲染方式

### React

用户输入 🡪 从API获取数据 🡪 将数据传给顶层组件 🡪 React将每个组件渲染出来

### Backbone

浏览器请求界面 🡪 服务器请求数据库 🡪 将数据传递给模板 🡪 模板渲染界面

## 第三方的模块集成

### React

React不用jQuery，因此现成的jQuery组件无法直接被React的组件调用（也不符合React的设计模式），对于需要的组件，需要全部按照React的方式重新设计。

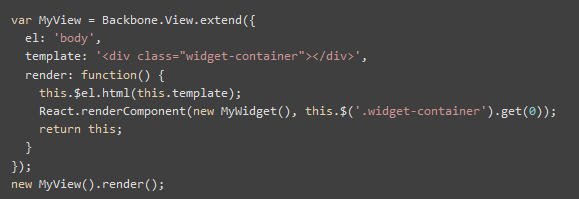
### Backbone

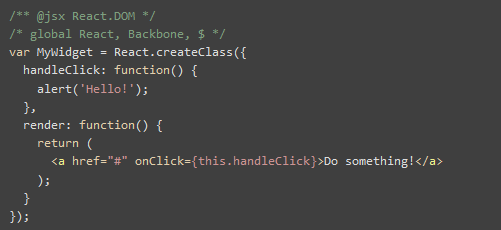
Backbone采用的模板填充方式，本质上在填充的时候仍然采用的是jQuery的方式，因此对于现成的jQuey插件可以方便的直接调用。

## 总结

Backbone和React并不是冲突的框架，Backbone是完整的MVC框架，而React则是专注于View层的框架。React需要搭配其他框架使用；Backbone的View层比较薄弱，可以考虑Backbone + React的方案。

### 由React完成Backbone中的组件封装工作，进而由这些组件完成backbone中View层的渲染。





### React组件->backbone通信

