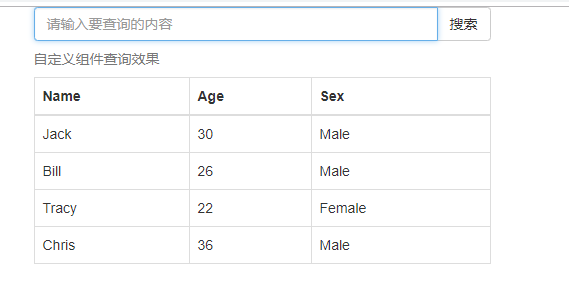
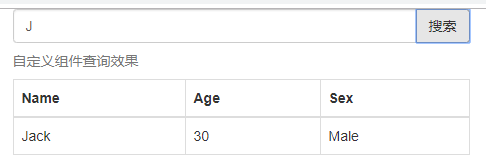
**关于写一个搜索的案例来了解Vue2.0自定义组件中事件的分发和接收机制**

案例效果图：



输入值点击搜索按钮后：



<**html** lang="en">  
<**head**>  
 <**meta** charset="UTF-8">  
 <**title**>自定义组件--CURD</**title**>  
 <**link** href="../../static/bootstrap3/css/bootstrap.min.css" rel="stylesheet">  
 <**script** src="../../static/bootstrap3/css/bootstrap-theme.min.css"></**script**>  
 <**script** src="../../static/assets/js/jquery-2.1.4.min.js"></**script**>  
 <**script** src="../../static/bootstrap3/js/bootstrap.min.js"></**script**>  
 <**script** language="JavaScript" src="../../static/js/vue.js"></**script**>  
</**head**>  
<**body**>  
 <**div** class="container">  
 <**div** class="row">  
 <**div** class="col-md-5 col-md-offset-3">  
 <**div** id="app">  
 <**div** class="input-group">  
 <**input** type="text" class="form-control" placeholder="请输入要查询的内容" v-model="queryValue">  
 <**span** class="input-group-btn">  
 <**button** class="btn btn-default" type="button" @click="notifySearch">搜索</**button**>  
 </**span**>  
 </**div**>  
 <**grid-table** :data="gridData"></**grid-table**>  
 </**div**>  
 </**div**>  
 </**div**>  
 </**div**>  
  
 <**template** id="grid-table">  
 <**table** class="table table-bordered">  
 <**caption**>自定义组件查询效果---自定义事件及事件的发送和接收

</**caption**>  
 <**thead**>  
 <**tr**>  
 <**th** v-for="col in columns">  
 {{col | capitalize}}  
 </**th**>  
 </**tr**>  
 </**thead**>  
 <**tbody**>  
 <**tr** v-for="entry in data">  
 <**td** v-for="col in columns">  
 {{entry[col]}}  
 </**td**>  
 </**tr**>  
 </**tbody**>  
 </**table**>  
 </**template**>  
  
 <**script** language="JavaScript">  
  
 **var *vm***=**new** Vue();  
  
 Vue.filter("capitalize",**function**(value){  
 **if**(!value) **return** "";  
 value=value.toString();  
 **return** value.charAt(0).toUpperCase()+value.slice(1);  
 });  
  
 Vue.component("grid-table", {  
 template: "#grid-table",  
 data:**function**(){  
 **return**{  
 columns:['name', 'age', 'sex']  
 }  
 },  
 props: {  
 data: **Array** },  
 computed: {  
 filterColumns: **function** () {  
 **var** self = **this**;  
 **return** self.columns.filter(**function** (col) {  
 **if** (!col) **return false**;  
 **return true**;  
 });  
 }  
 },  
 created:**function**(){  
 **var** self = **this**;  
 ***vm***.$on("change-gridData",**function**(msg){  
 self.data.filter(**function**(obj){  
 **if**(!obj)**return false**;  
 **return** obj.name.indexOf(msg)!==-1;  
 });  
 });  
 }  
 });  
  
 **var *v*** = **new** Vue({  
 el: "#app",  
 data: {  
 queryValue: "",  
 gridData: [{  
 name: 'Jack',  
 age: 30,  
 sex: 'Male'  
 }, {  
 name: 'Bill',  
 age: 26,  
 sex: 'Male'  
 }, {  
 name: 'Tracy',  
 age: 22,  
 sex: 'Female'  
 }, {  
 name: 'Chris',  
 age: 36,  
 sex: 'Male'  
 }]  
 },  
 methods: {  
 notifySearch: **function** () {  
 **var** s=***v***;  
 **var** self = **this**;  
 ***vm***.$emit("change-gridData", self.queryValue);  
 }  
 }  
 });  
 </**script**>  
</**body**>  
</**html**>

**改上面的错误写法：**

**定义一个全局变量var vmCpt;**

**然后之前标注的代码改为：**

created:**function**(){  
  **var vmCpt=this;**  
 *vm*.$on("change-gridData",**function**(msg){  
 vmCpt.data=vmCpt.data.filter(**function**(item){  
 **if**(!item) **return false**;  
 **return** item.name.indexOf(msg)!==-1;  
 });  
 });  
}

**总结：**

**Vue2.0中是使用一个Vue的实例来分发和接收事件的**

关于事件的修饰符：

* **.stop**：等同于JavaScript中的 event.stopPropagation() ，防止事件冒泡
* **.prevent**：等同于JavaScript中的 event.preventDefault() ，防止执行预设的行为（如果事件可取消，则取消该事件，而不停止事件的进一步传播）
* **.capture**：与事件冒泡的方向相反，事件捕获由外到内
* **.self**：只会触发自己范围内的事件，不包含子元素
* **.once**：只会触发一次