Vue.js Ajax(vue-resource) Vue.js 实例

Vue.js 响应接口

Vue 可以添加数据动态响应接口。

例如以下实例，我们通过使用 $watch 属性来实现数据的监听，$watch 必须添加在 Vue 实例之外才能实现正确的响应。

实例中通过点击按钮计数器会加 1。setTimeout 设置 10 秒后计算器的值加上 20 。

实例

```
<div id = "app">
<p style = "font-size:25px;">计数器: {{ counter }}</p>
<button @click = "counter++" style = "font-size:25px;">点我</button>
</div>
<script type = "text/javascript">
var vm = new Vue({
el: '#app',
data: {
counter: 1
}
});
vm.$watch('counter', function(nval, oval) {
alert('计数器值的变化 :' + oval + ' 变为 ' + nval + '!');
});
setTimeout(
function(){
vm.counter += 20;
},10000
);
</script>
```

尝试一下 »

Vue 不允许在已经创建的实例上动态添加新的根级响应式属性。

Vue 不能检测到对象属性的添加或删除，最好的方式就是在初始化实例前声明根级响应式属性，哪怕只是一个空值。

如果我们需要在运行过程中实现属性的添加或删除，则可以使用全局 Vue，Vue.set 和 Vue.delete 方法。

Vue.set

Vue.set 方法用于设置对象的属性，它可以解决 Vue 无法检测添加属性的限制，语法格式如下：

```
Vue.set( target, key, value )
```

参数说明：

- target: 可以是对象或数组
- key : 可以是字符串或数字
- value: 可以是任何类型

实例

```
<div id = "app">
<p style = "font-size:25px;">计数器: {{ products.id }}</p>
<button @click = "products.id++" style = "font-size:25px;">点我</button>
</div>
<script type = "text/javascript">
var myproduct = {"id":1, name:"book", "price":"20.00"};
var vm = new Vue({
el: '#app',
data: {
counter: 1,
products: myproduct
}
});
vm.products.qty = "1";
console.log(vm);
vm.$watch('counter', function(nval, oval) {
alert('计数器值的变化 :' + oval + ' 变为 ' + nval + '!');
});
</script>
```

尝试一下 »

在以上实例中，使用以下代码在开始时创建了一个变量 myproduct：

```
var myproduct = {"id":1, name:"book", "price":"20.00"};
```

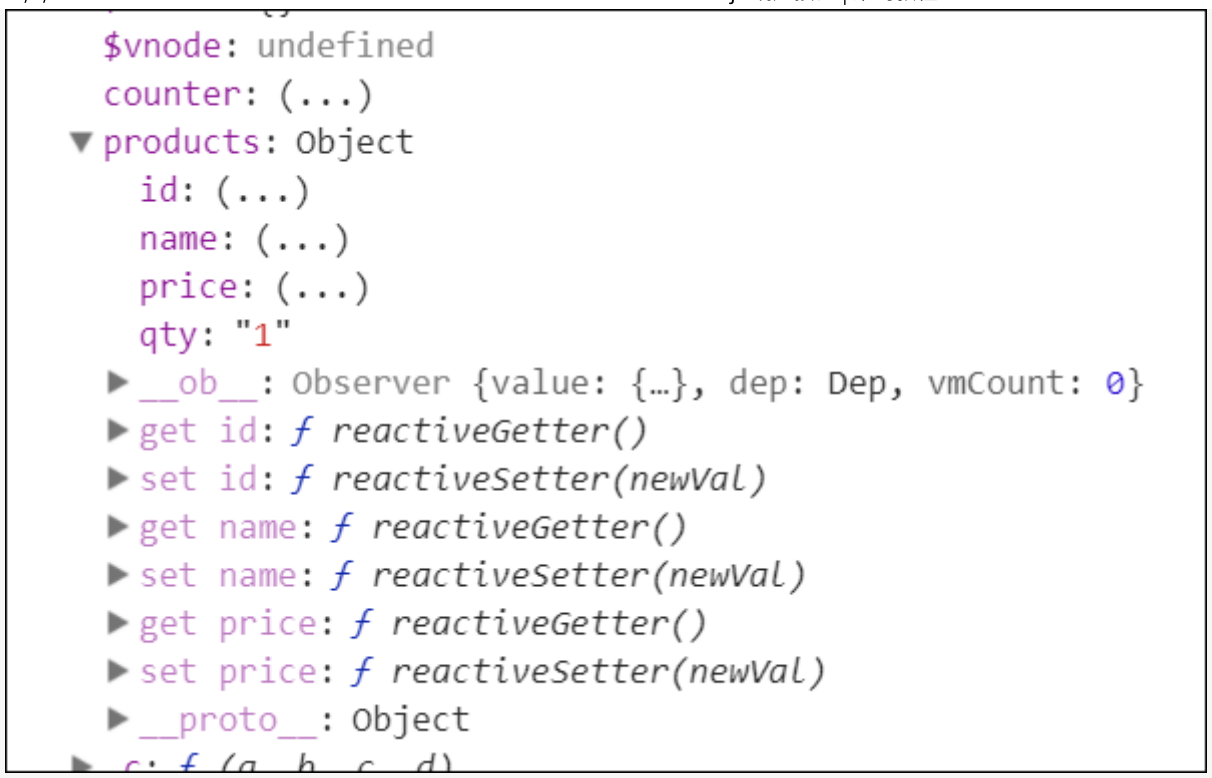
该变量在赋值给了 Vue 实例的 data 对象：

var vm = new Vue({ el: '#app', data: { counter: 1, products: myproduct } });

如果我们想给 myproduct 数组添加一个或多个属性，我们可以在 Vue 实例创建后使用以下代码：

```
vm.products.qty = "1";
```

查看控制台输出：



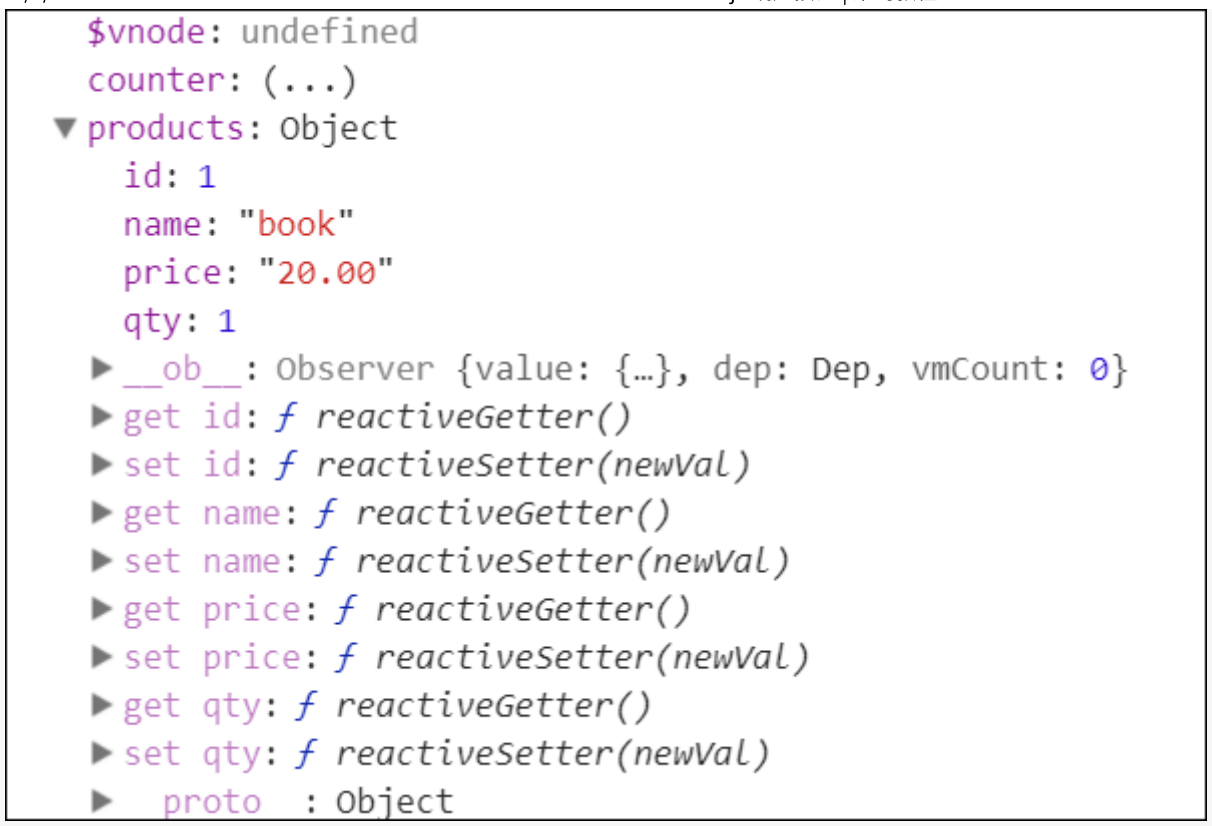
如上图看到的，在产品中添加了数量属性 qty，但是 get/set 方法只可用于 id，name 和 price 属性，却不能在 qty 属性中使用。

我们不能通过添加 Vue 对象来实现响应。 Vue 主要在开始时创建所有属性。 如果我们要实现这个功能，可以通过 Vue.set 来实现：

实例

```
<div id = "app">
<p style = "font-size:25px;">计数器: {{ products.id }}</p>
<button @click = "products.id++" style = "font-size:25px;">点我</button>
</div>
<script type = "text/javascript">
var myproduct = {"id":1, name:"book", "price":"20.00"};
var vm = new Vue({
el: '#app',
data: {
counter: 1,
products: myproduct
}
});
Vue.set(myproduct, 'qty', 1);
console.log(vm);
vm.$watch('counter', function(nval, oval) {
alert('计数器值的变化 :' + oval + ' 变为 ' + nval + '!');
});
</script>
```

尝试一下 »



从控制台输出的结果可以看出 get/set 方法可用于qty 属性。

Vue.delete

Vue.delete 用于删除动态添加的属性 语法格式：

```
Vue.delete( target, key )
```

参数说明：

- target: 可以是对象或数组
- key : 可以是字符串或数字

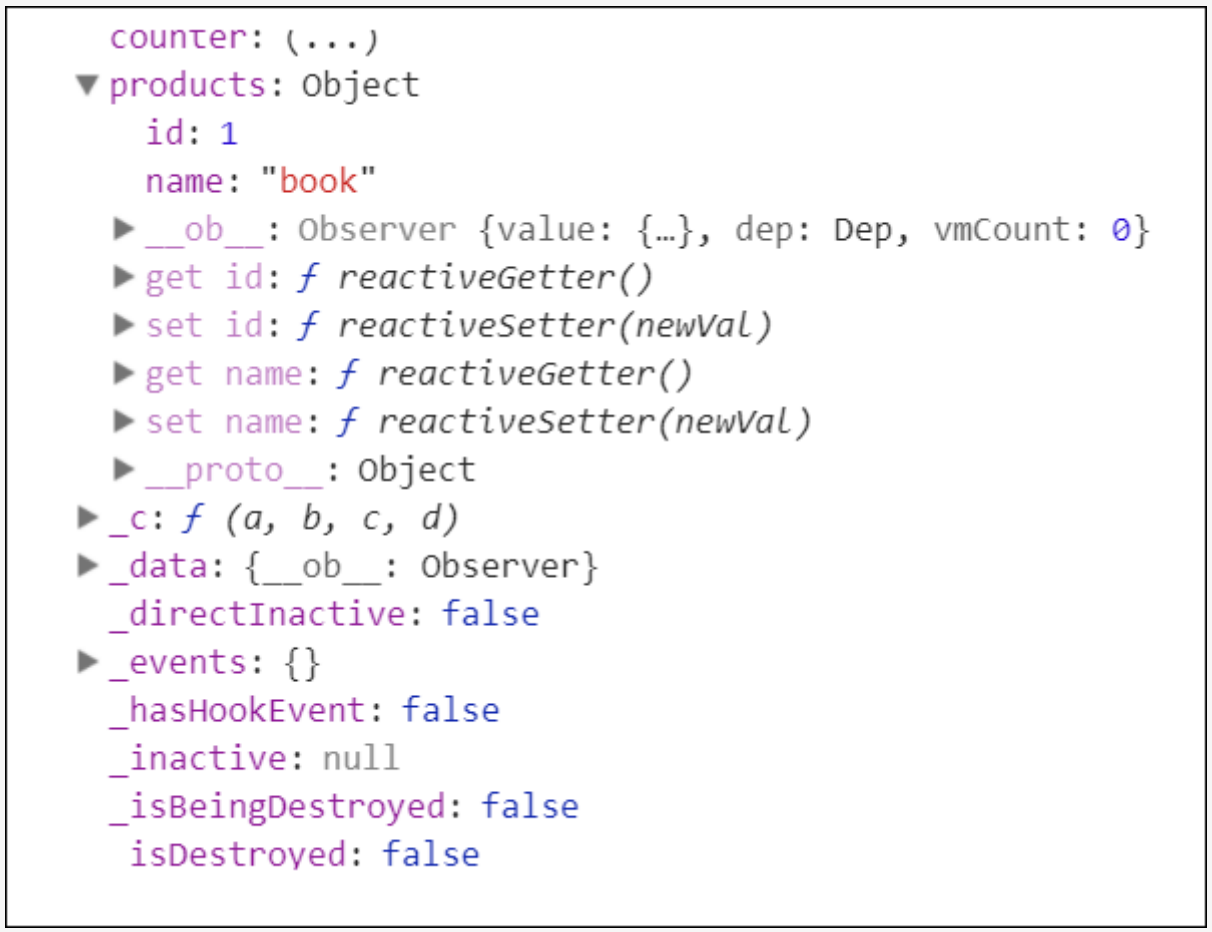
实例

```
<div id = "app">
<p style = "font-size:25px;">计数器: {{ products.id }}</p>
<button @click = "products.id++" style = "font-size:25px;">点我</button>
</div>
<script type = "text/javascript">
var myproduct = {"id":1, name:"book", "price":"20.00"};
var vm = new Vue({
el: '#app',
data: {
counter: 1,
products: myproduct
}
});
Vue.delete(myproduct, 'price');
```

```
console.log(vm);
vm.$watch('counter', function(nval, oval) {
alert('计数器值的变化 :' + oval + ' 变为 ' + nval + '!');
});
</script>
```

尝试一下 »

以上实例中我们使用 Vue.delete 来删除 price 属性。以下是控制台输出结果：



从上图输出结果中，我们可以看到 price 属性已删除，只剩下了 id 和 name 属性，price 属性的 get/set 方法也已删除。

← Vue.js Ajax(vue-resource) Vue.js 实例 →