

持有数据的组件

让组件持有数据的能力是RCRE最有特色的功能. 灵活使用组件的数据功能, 开发者就能从使用一种简单, 易于维护, 直观的方式来表述单页应用复杂的业务逻辑. 在前端开发过程中, 不管使用何种框架, 复杂的业务逻辑都无法避免, 而框架能做的, 就是让开发者使用最小的成本, 最低的代码改动量, 来实现业务所需要的功能.

组件的概念

在理解持有数据的组件之前, 需要理解在RCRE中, 什么是一个组件.

例如上一个章节的helloworld代码.

```
1  var config = {
2    "body": [
3      {
4        "type": "text",
5        "text": "this is a simple demo"
6      }
7    ]
8  };
9
10 var engine = React.createElement(RCRE, {
11   code: JSON.stringify(config)
12 });
```

通过RCRE来实现复杂的页面结构和业务逻辑, 都将是开发者编写JSON结构的描述, 然后传入到RCRE引擎组件内部.

JSON结构中的一个对象, 就称作是一个组件.

```
1  {
2    "type": "text",
3    "text": "this is a simple demo"
4  }
```

这样一个JSON结构, 就代表是一个Text文本组件.

`type` 属性是一个必选的属性值, `type: text` 就代码这是一个text组件.

Text组件提供一个 `text` 属性, 通过 `text` 属性, 就能直接让Text组件展示 `text` 属性值的内容.

所以上面这个JSON结构就会得到下面的运行结果.

```
this is a simple demo
```

Container组件

RCRE中提供一个特殊的组件 —— `container` 组件.

`container` 组件就是整个RCRE中的组件数据源, 它可以为字级的所有组件提供数据源, 同时 `container` 组件可以给 `container` 组件提供数据源. `container` 组件也可以通过一种可扩展的机制来扩容获取数据的方式. 后续的章节, 将会初步围绕 `container` 组件来为读者依依介绍.

数据源Key

RCRE中会有很多个 `container` 组件, 同时RCRE会持有每个 `container` 组件的数据模型. 所以就需要一个不能重复的数据源Key来指代每个 `container` 组件. 所以就需要 `model` 属性.

通过 `model` 属性, 就可以给 `container` 组件设置数据源Key

```
1 {  
2   "type": "container",  
3   "model": "demo"  
4 }
```

字级组件

空的 `container` 组件不会在页面上有任何效果. 而且还会报一个 `children props must be specific in Container Component` 这样的异常. `container` 组件在设计上就是为其他组件提供数据操作能力的. 因此它必定会有自己的字级组件.

通过 `children` 属性, 就可以在 `container` 组件内部, 放置一些其他类型的组件.

```
1 {  
2   "type": "container",  
3   "model": "demo",  
4   "children": [  
5     {  
6       "type": "text",  
7       "text": "helloworld"  
8     },  
9     {  
10      "type": "text",  
11      "text": "another text"  
12    }  
13  ]  
14 }
```

数据对象

`container` 组件可以在初始化的时候, 对数据模型写入初始值.

通过 `data` 属性, 就为 `container` 组件写入初始值.

```
1 {
2   "type": "container",
3   "model": "demo",
4   "data": {
5     "name": "andycall",
6     "age": 21
7   },
8   "children": [
9     {
10      "type": "text",
11      "text": "helloworld"
12    },
13    {
14      "type": "text",
15      "text": "another text"
16    }
17  ]
18 }
```

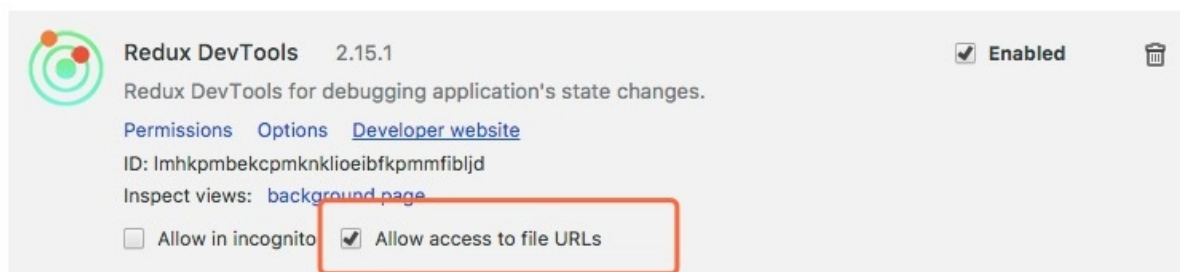
这个例子. 会让RCRE在渲染这个 `container` 组件的时候, 将 `name: andycall` 和 `age: 21` 写入到数据源Key为 `demo` 的 `container` 组件内部. 下个章节中, 将会介绍如何利用数据模型中的数据, 来动态展示一些值.

查看数据模型中的值

RCRE的数据模型是保存在Redux的store中, 所以我们可以通过redux devtools工具来查看目前数据模型所带有的值.

可以在Chrome浏览器商店上, 通过这个[下载地址](#)进行下载安装.

安装完成之后, 需要在 `chrome://extensions/` 中, 把Redux DevTools的访问本地文件的权限打开, 这样才能在直接打开单个html文件情况下, 使用redux devtools.



之后Redux将会出现在Chrome的控制台中. 这样我们就可以通过Redux这个工具来查看当前数据模型中的值

State

ActionStateDiffTest

TreeChartRaw

container (pin)

▼ demo (pin)

name (pin): "andycall"

age (pin): 21