



vfront组件开发指南

开发模式

在vue体系中，组件开发的模式和页面的开发模式几乎一样。因为我们可以把页面当作是更大的组件。但是我们仍然要注意区别于页面的部分：

- 组件是不允许包含业务逻辑的
- 组件是复用性更强的

因此，组件中，大致需要包含三种数据：

- props 外部属性
- data 内部属性
- slot 插槽

考虑到vfront架构中，使用的是单文件组件，因此本指南也将继续传递这种开发模式。

步骤

我们将模仿官方组件的案例，来看构建一个组件的过程。我们准备做一个自己的按钮。

构建文件结构

在 `src/compoments` 目录下新建目录 `my-link`，然后在 `my-link` 目录下新建 `my-link.vue` 文件。如果自定义组件中仍然有其他的视图逻辑、功能逻辑以及子组件，也应该存放在该文件夹中。最终，我们只会引用到 `my-link.vue` 文件组件。其他的文件夹和子组件都将是本文件夹组件私有独占的。

编写模版

接着，我们就可以开始编写模版文件了：

```
<template>
<div>
  <a :class="classes" v-on:click="onClick"> </a>
  <slot> </slot>
</div>
</template>
```

导出组件

```
<script>
export default {
  name: 'MyLink',
  props: {
    name: {
      type: String,
      default: '默认按钮'
    },
    size: {
      type: String,
      validator (value) {
        return ~['small', 'large', 'default'].indexOf(value);
      }
    }
  },
  computed: {
    classes () {
      return [
        `my-link-${this.size}`]: !!this.size
      ]
    }
  },
  methods: {
    onClick () {
      this.$emit('click', this.name);
    }
  }
};
</script>
```

在导出组件过程中，我们使用了props属性，用于接受外部的参数。data和computed中可以直接使用props属性。

使用和测试

同页面的套用一样，用 `import` 语法来导入组件并使用。目前并没有相关测试的好的方案，以后会逐步追加。

更多

更多资料及语法，请[点击这里查阅官网](#)