



开发拓展篇3：尝试使用外部数据

当你应用的UI 层面已经趋于完善的时候，接下来你就需要去获取动态的数据来实现真实的应用场景。那么动态数据从哪里来呢？前端主要还是通过接口的形式获取。

如果有专业的接口开发人员和我们一起完成一个应用，那么你只需要和他定义好接口的入参和出参，然后进行调用就好了，这也是公司中前端与后台的常见合作方式。

而在以下场景下，你可能并不需要或者并没有后台工程师提供接口给你：

- 你个人开发的项目
- 你的应用数据来源于外部

这时候你就需要通过调用**第三方接口**来实现应用的数据交互与展现。那么接下来我们就来介绍下第三方接口的使用。

介绍

大多数情况下，我们调用的第三方接口都是完全或者部分开源及免费的，因此只要在合理的范围内使用我们便可以实现一些简单的数据交互。本文将以[聚合数据](#)提供的第三方接口API为例进行讲解。

聚合数据

聚合数据是国内领先的基础数据服务商，以自有数据为基础，各种便捷服务整合以及第三方数据接入，为互联网开发全行业提供标准化API 技术支撑服务的DaaS 平台。

在官网API首页（www.juhe.cn/docs）我们可以选择你想要的接口类型进行调用，如下图所示：



其中包含了免费及付费的接口类型，如果只是用于单个的调用或测试，建议大家使用免费接口就可以了（非会员只能申请一个免费接口，比较坑），但如果你的应用准备上架或发布，你最好付费以获得更多的调用和使用接口的次数。

点击你想使用的接口后你便可以查看该接口的API 文档，包括接口地址、请求方式、请求示例及返回示例等，比如这里我点击“笑话大全”：



获取到 AppKey 后我们便可以对接口进行测试了，这里我们可以直接通过对应接口的测试按钮进行测试，当然你也可以通过 `postman` 工具进行调试：

测试完毕后，我们便可以在 Vue 项目中接入我们的第三方接口，实现动态数据和功能。

其他第三方接口

除了整合型的第三方接口聚合数据外，其他比较实用的第三方接口还有：

- [高德地图](#)：访问高德地图的 Web API



- [和风天气](#)：中国天气信息
- [阿凡达数据](#)：提供中国可用的 API

实例

接下来，我们便可以在 Vue 项目中接入第三方接口来实现数据的交互。这里我们以接入聚合数据的 [历史上的今天](#) 接口为例。

首先我们来看一下该接口的请求详情：

请求地址：<http://api.juheapi.com/japi/toh>
请求参数：`v=1.0&month=10&day=31&key=你的AppKey`
请求方式：GET

上方详情中我们可以看到请求参数主要有 4 个，分别为：

- v: string 类型。版本，当前：1.0
- month: int 类型。月份，如：10
- day: int 类型。日，如：1
- key: string 类型。你的 AppKey

通过接口入参的说明我们可以知道唯一可以变的数据便是月份和日期，所以我们页面中得有选择月日的功能。

另外我们再来看一下该接口的出参示例：

```
json
{
  "error_code": 0,
  "reason": "请求成功!",
  "result": [
    {
      "day": 1,
      "des": "1907年11月1日 电影导演吴永刚诞生 &emsp;&emsp;吴永刚，1907年11月1日生于江",
      "id": 9000,
      "lunar": "丁未年九月廿六",
      "month": 11,
```



```
      },  
    },  
  }  
}
```

通过上方示例，我们可以获取**历史上的今天的**标题、详情、图片等，这些数据可以用于页面展示。

这样经过接口入参和出参的分析，我们可以使用 Vux 构建一个简单的页面，如下图所示：

通过点击上图中“查看历史上的今天”按钮，便可以调用接口获取数据列表并渲染。主要代码如下：



```
<datetime-view v-model="value" ref="datetime" :format="format"></datetime-view>
<p class="info">选中值: {{ value }}</p>
<div class="btn-padding">
  <x-button type="primary" @click.native="watchHistory">查看历史上的今天</x-button>
</div>
<panel header="历史上的今天" :list="list" type="1"></panel>
</div>
</template>
<script>
import { DateTimeView, XButton, Panel } from 'vux'
import { getHistory } from '_ser/moduleB'

export default {
  data() {
    return {
      value: '10-31',
      format: 'MM-DD',
      list: []
    }
  },
  methods: {
    watchHistory() {
      let data = this.value.split('-')

      this.list = [];

      getHistory({
        v: '1.0',
        month: data[0],
        day: data[1],
        key: 'd6ceaf9be9f116ae45e7699845d87056'
      }).then(response => {
        if (!response.error_code) {
          response.result.map(e => {
            this.list.push({
              title: e.title,
              desc: e.desc,
              src: e.pic,
            })
          })
        }
      })
    }
  }
}
```

```
        XButton,  
        Panel  
      },  
    }  
  }  
</script>
```

上方我们通过调用封装的 `getHistory` 接口名称进行数据的获取，在本地调用的过程中需要注意跨域的问题，可以配置 devServer 的 proxy 代理来解决：

```
/* vue.config.js */  
  
module.exports = {  
  devServer: {  
    proxy: {  
      '/juheapi': {  
        target: 'http://api.juheapi.com/',  
        changeOrigin: true,  
        pathRewrite: {'^/juheapi': ''}  
      }  
    }  
  }  
}
```

具体实例代码可以参考：[ui-framework-project](#)

这样我们便完成了使用第三方接口获取数据实现页面渲染的功能，为自己的应用填充了数据和功能。

结语

第三方接口的使用能够帮助我们快速的获取数据并实现应用的动态交互，同时也有助于解决接口开发的人力及服务器资源消耗，为部分 Vue 项目开发提供数据保障。

至此，本小册的开发部分章节也将告一段落。在这部分的内容中我们一起学习了 Vue 项目开发的实用技巧、方法和工具，并在构建出的项目基础上了解并实践了 Vue 开发的相关技术要点。希望这部分内容的介绍能够巩固大家对于 Vue 开发的基础知识，弥补 Vue 开发的技能空白。



- 自己通过调用外部数据完成一个页面动态的数据渲染
- devServer 的 proxy 代理是如何实现接口转换和重定向的？各配置项的作用是什么？

留言

评论将在后台进行审核，审核通过后对所有人可见

霍霖笙箫 前端工程师

很实用的小册。一开始犹豫了下但这10块挂得真心值，作者提供了很好的思路和疑点梳理，感谢分享。

▲ 3 评论 2月前

拖出去斩了

老师，问一下，react 中建议将ajax放在componentDidMount，而不是componentWillMount阶段，vue的ajax一般放在created 阶段有问题吗？

▲ 0 收起评论 2月前

劳卜 前端工程师 @ TC

componentWillMount 阶段组件挂载成功，和 vue 中的 mounted 类似，在这个阶段获取数据最好

2月前

评论审核通过后显示

评论

Si

webpack 的跨域 proxy 配置在 build 之后为什么无效？

▲ 0 收起评论 3月前



node 发布云的服务端没有跨域问题,你 build 之后有足够文件了,所以而又你部署的服务器帮你做代理

3月前

评论审核通过后显示

评论

Robocop Mao web前端工程师

感觉小册写的很好, vue的多页和webpack配置等很有用

▲ 2 评论 3月前

我是豪 web 前端

终于熬到凌晨把这本小册子通读了一遍,我觉得这本小册涵盖了好多知识点和开发技巧,给了自己醍醐灌顶的感觉,大部分代码通俗易懂,代码注释也很多,还有源代码可以参考,我觉得很值得收藏,温故知新,明天我肯定会分享给我们前端同事来支持的!谢谢作者,我关注你了!

▲ 0 收起评论 4月前

劳卜 前端工程师 @ TC

非常感谢你的支持,一起加油~

4月前

iwasmaster Front-End Developer @ Hikvision

这么夸张?

3月前

评论审核通过后显示

评论