

Google Map

项目背景

Tajer 是一个以 Instagram 卖家为目标用户的快销型电商项目。为降低买家在填写地址表单的复杂度，引入了谷歌地图。项目基于 Nuxt 框架开发。

如何引入谷歌地图

1. 采用第三方封装的 vue 组件
优点：可以快速实现功能
缺点：
 - 缺少扩展性，可能无法应对未来的需求变更。
 - 如果第三方的实现有缺陷，会面临无法填坑的危险。
2. 自己封装
优点：自由，可以实现任何官方支持的功能。
缺点：难度大，实现功能慢。

封装地图组件的难点

Google Map 官网示例代码

```
<script>
  var map;
  function initMap() {
    map = new google.maps.Map(document.getElementById('map'), {
      center: {lat: 34.397, lng: 150.644},
      zoom: 8
    });
  }
</script>
<script src="https://maps.googleapis.com/maps/api/js?key=【YOUR_API_KEY】
&callback=initMap"></script>
```

那么是不是可以愉快地 ctrl C + ctrl V 呢？

1. 代码库引入的特殊性

常规封装思路

```
import xxx from 'xxx'
```

目前可行方法

```
<!-- 在 Html 模版中引入 -->
<!-- 无法实现懒加载，每个路由都会加载这个文件。 -->
<script src="https://maps.googleapis.com/maps/api/js?key=【YOUR_API_KEY】"></script>
```

```
// 可以实现懒加载
const script = document.createElement('script')
script.src = 'https://maps.googleapis.com/maps/api/js?key=[YOUR_API_KEY]'
document.head.appendChild(script)
```

2. 框架层面的限制

Nuxt 是一个服务端渲染的框架，写的代码如果是在服务端运行该怎么办？

服务端没有 document 对象，会报错！

那么该在哪里进行地图的初始化呢？

在 Nuxt 框架中，beforeCreate & created 这两个函数，既会运行在服务端，也会运行在浏览器端。当浏览器通过 URL 首次请求服务器时，这两个函数是运行在服务端的。用户后续点击页面中的链接，进行页面跳转时，这两个钩子函数是运行在浏览器端的。

而 beforeMount 方法中，Dom 节点尚未生成，但 Google Map 初始化需要一个 Dom 节点，所以也是不合适的。

解决方法：在 `mounted` 函数中加载地图库。

```
// 地图初始化函数
let initialized = false
let resolveInitPromise;
let rejectInitPromise;
const initPromise = new Promise((resolve, reject) => {
  resolveInitPromise = resolve;
  rejectInitPromise = reject;
});

export default function init(API_KEY) {
  if (initialized) return initPromise;
  initialized = true;
  const script = document.createElement('script');
  script.src = `https://maps.googleapis.com/maps/api/js?key=${API_KEY}&libraries=places`;
  script.onerror = rejectInitPromise;
  script.onload = () => {
    resolveInitPromise(window.google)
  };
  document.querySelector('head').appendChild(script);
  return initPromise;
}
```

```
import gmapsLoad from './gmap.js'
{
  mounted {
    const google = await gmapsLoad(Config.googleMapKey)
    const map = new google.maps.Map(document.querySelector('.google-map'), {
      // some options here
    });
  }
}
```

事件监听

```
const map = new google.maps.Map(document.querySelector('.google-map'), {
  // some options here
});
map.addListener('eventName', event => {
  // event handler
})
```

常用的事件有：

- click
- touch_start、touch_move、touch_end
- center_change

获取定位权限

具体实现

获取定位有两个 API 可以使用：

- navigator.geolocation.getCurrentPosition()
特点：速度快，精度稍低，回调函数仅触发一次。
- navigator.geolocation.watchPosition()
特点：速度慢，精度较高，在用户改变地理位置时，回调函数将被多次触发。必要时需要 **解除监听**。
使用 navigator.geolocation.clearWatch() 解除监听。解除监听需要一个 **监听 id**，类似于 setTimeout & clearTimeout 的关系。

具体的调用方式：

```
if (navigator.geolocation) {
  const onSuccess = position => {
    const { latitude, longitude } = position.coords
  }
  const onError = errorMessage => {
    alert('获取定位权限失败')
  }
  navigator.geolocation.getCurrentPosition(onSuccess, onError) // 此时将弹框向用户请求权限
  // navigator.geolocation.watchPosition(onSuccess, onError)
}
```

注意事项

- HTTPS环境！HTTPS环境！HTTPS环境！
在 HTTP 环境下，将 **不会** 向用户请求权限，直接调用失败回调函数。
本地开发时请使用 localhost 访问，或使用 Nginx 创建本地的 https 服务。 **不要** 使用 ip 访问，不然地图不会加载成功的。
- 注意地图组件加载的时间点
建议使用 v-if 懒加载地图组件，提升首屏体验。

因为组件初始化的顺序是 加载 js 库 => 请求定位权限 => 地图展现，所以如果进入页面直接加载地图，此时地图组件是一片空白，用户将在一脸懵逼的情况下面对蹦出来的权限请求，用户体验较差。

- 页面中的图片链接

安卓端只需要主域名 HTTPS 即可请求定位权限，站内图片可使用 http 链接。

而Safari 对全站 HTTPS 化要求极高，一旦夹杂 http 图片请求，将默认拒绝定位权限。

PS: 在 Safari 中，一旦用户首次拒绝了定位权限，只能进入设置中，清除 Safari 缓存，才能在下次进入站点时重新弹出定位权限请求的弹窗。

IP 定位

如果用户死活不同意定位怎么办？如何做降级处理？

[文档：IP定位](#)

```
const response = await axios.post(
  `https://www.googleapis.com/geolocation/v1/geolocate?key=${ Conf.googleMapKey }`,
  {
    considerIp: true
  }
)
if (response.status === 200) {
  const { location } = response.data
  this.mapOption.center = location // { lat: 'xxx', lng: 'yyy' }
  // 根据经纬度初始化地图
}
```