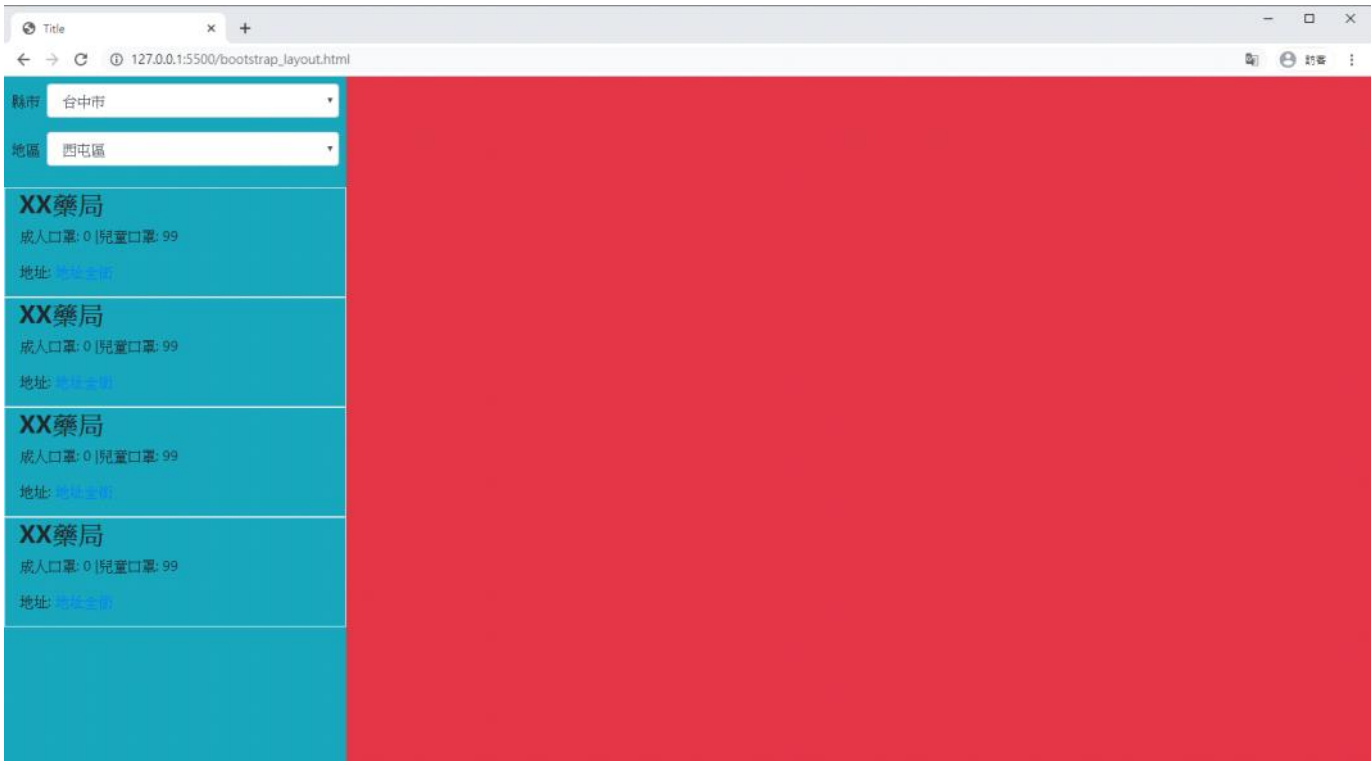
6. 口罩地圖 Mask

1. 設計畫面



定義出3:9 兩個色塊

```
<div class="container-fulid">
    <div class="row no-gutters" style="height: 100vh;">
        <div class="col-md-3 bg-info">…
        </div>
        <div class="col-md-9 bg-danger"></div>
    </div>
</div>
```

form-select

flex-fill 使得label字水平

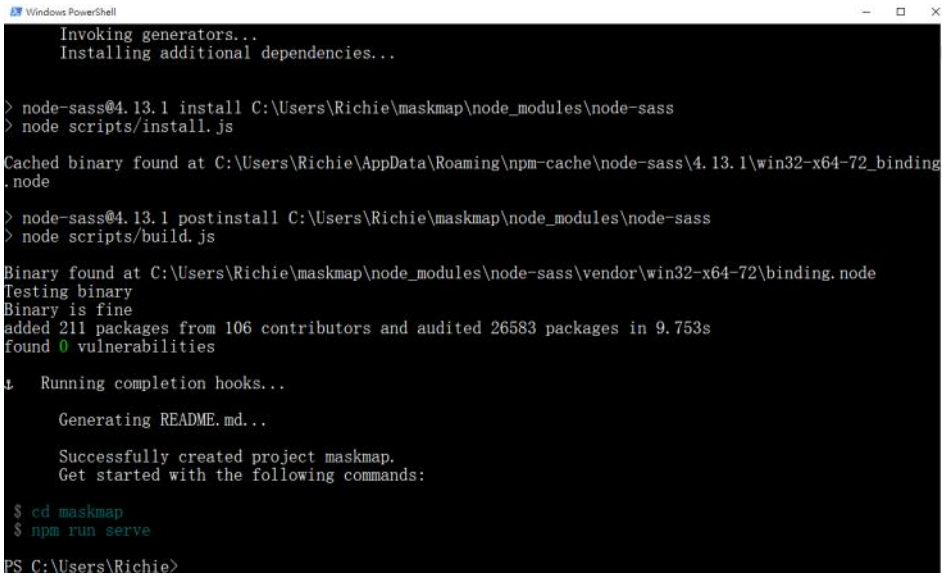
```
<div class="container-fulid">
    <div class="row no-gutters" style="height: 100vh;">
        <div class="col-md-3 bg-info">
            <div class="form-group d-flex p-2 mb-0">
                <label for="" class="mr-2 col-form-label text-right">縣市</label>
                <div class="flex-fill">
                    <select class="form-control">
                        <option>台中市</option>
                        <option>台北市</option>
                        <option>新竹市</option>
                        <option>桃園市</option>
                    </select>
                </div>
            </div>
            <div class="form-group d-flex p-2 mt-0">···
            </div>
            <div class="pl-3 border">···
            </div>
            <div class="pl-3 border">···
            </div>
            <div class="pl-3 border">···
            </div>
            <div class="pl-3 border">···
            </div>
        </div>
        <div class="col-md-9 bg-danger"></div>
    </div>
</div>
```

```
<div class="pl-3 border">
    <h3 class="font-weight-bold">XX藥局</h3>
    <p>成人口罩: 0 |兒童口罩: 99</p>
    <p>地址: <a href="" class="link">地址全銜</a></p>
</div>
```

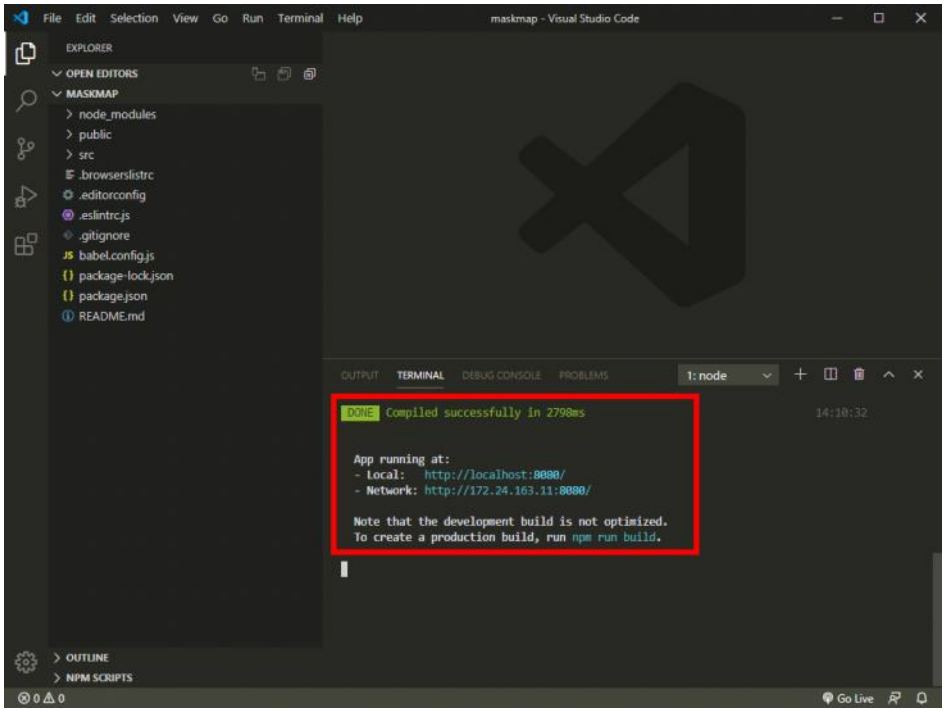
2. 安裝vue-cli

```
npm
Vue CLI v4.2.2

  New version available 4.2.2 → 4.2.3
  Run npm i -g @vue/cli to update!

? Please pick a preset: Manually select features
? Check the features needed for your project: Babel, CSS Pre-processors, Linter
? Pick a CSS pre-processor (PostCSS, Autoprefixer and CSS Modules are supported by default): Sass/SCSS
 (with node-sass)
? Pick a linter / formatter config: Airbnb
? Pick additional lint features: (Press <space> to select, <a> to toggle all, <i> to invert selection)
Lint on save
? Where do you prefer placing config for Babel, ESLint, etc.? In dedicated config files
? Save this as a preset for future projects? (y/N) n
```

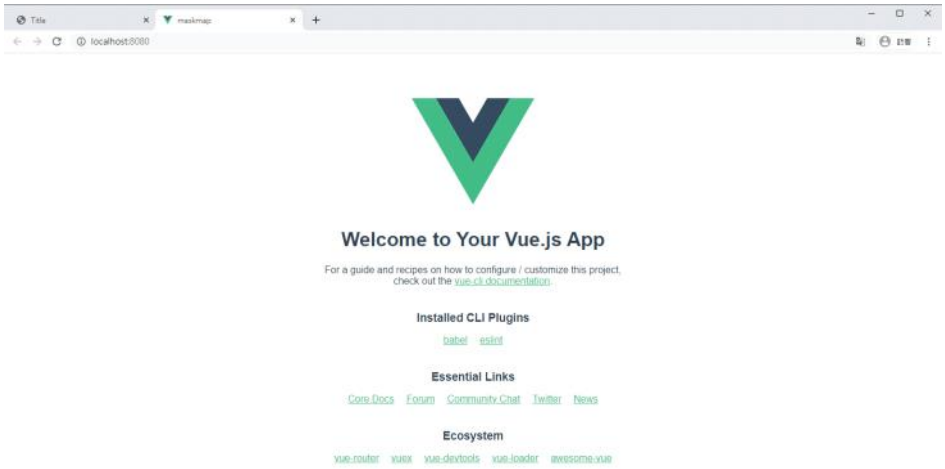
```
Vue CLI v4.2.2
✨  Creating project in C:\Users\Richie\maskmap.
⚙  Installing CLI plugins. This might take a while...

[.................] | fetchMetadata: sill pacote range manifest for strip-ansi@^3.0.1 fetched in 4ms
```



npm run serve

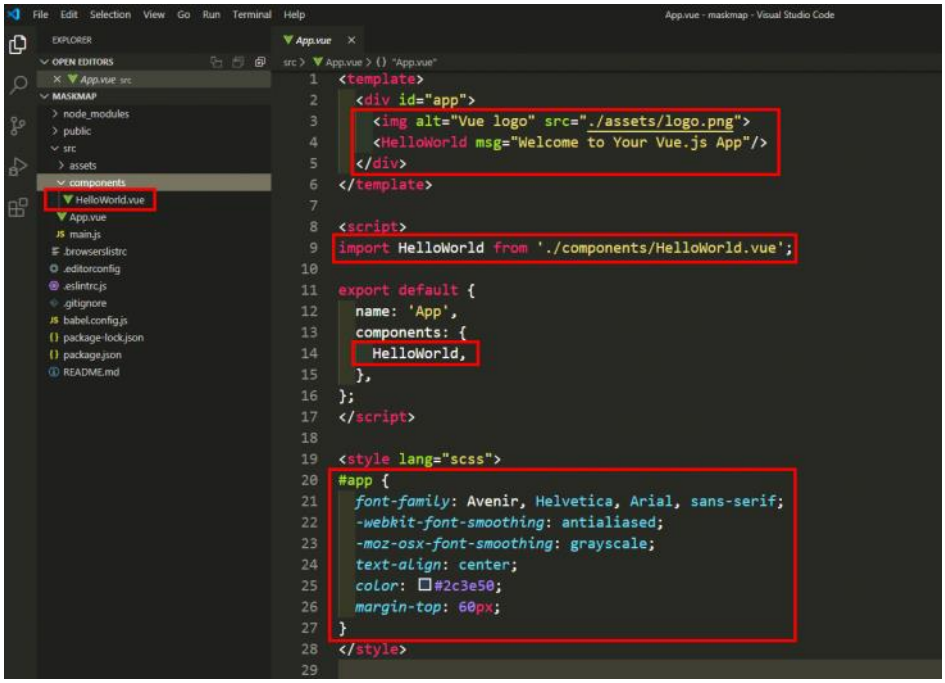


初始畫面

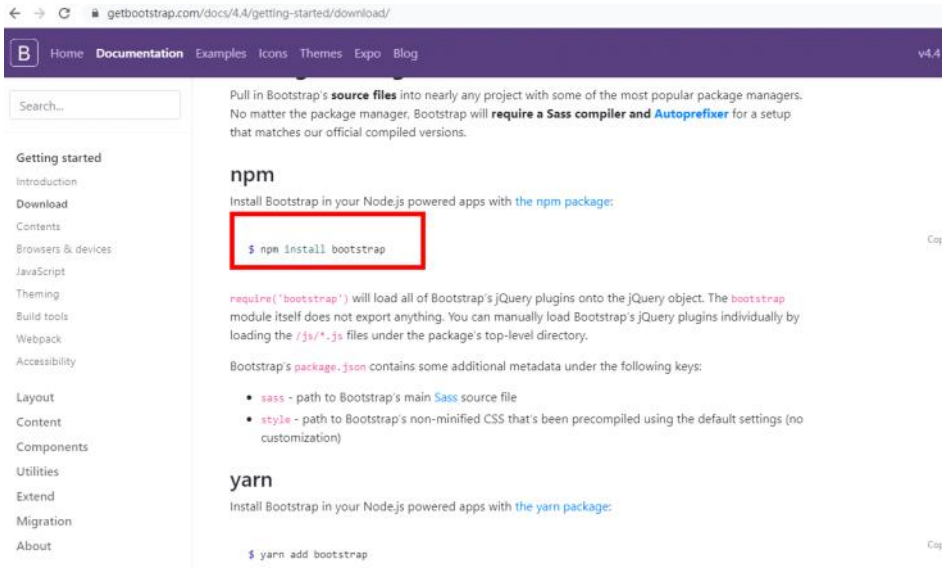


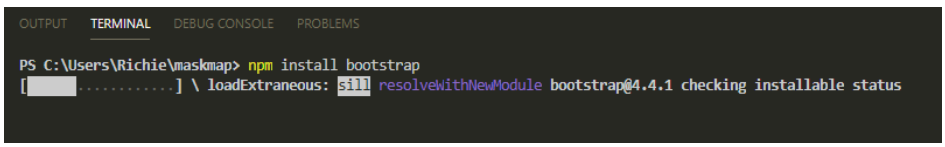
3. 精簡App.vue 內容

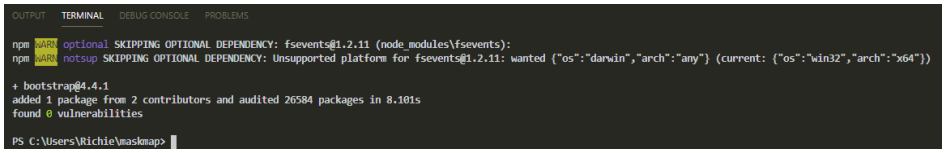
移除以下所圈選部分，只需將程式寫入App.vue，移除後vue的畫面應該為空白!



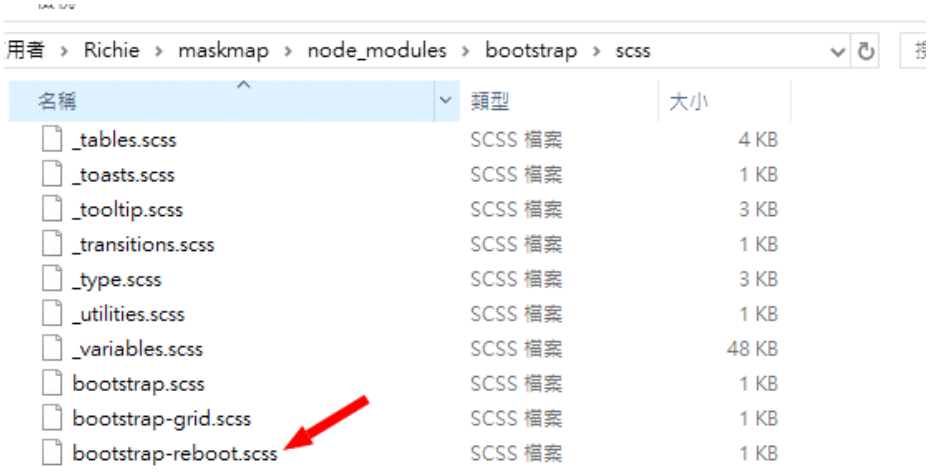
4. 安裝bootstrap



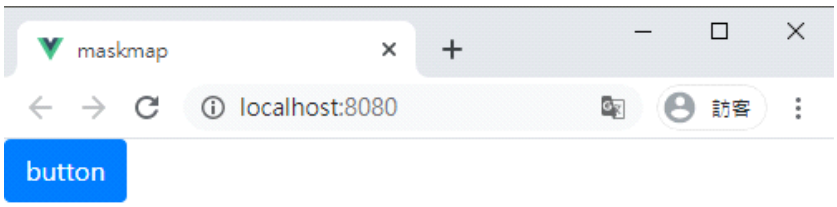




安裝完成後可以發現路徑為 node_modules\bootstrap\scss

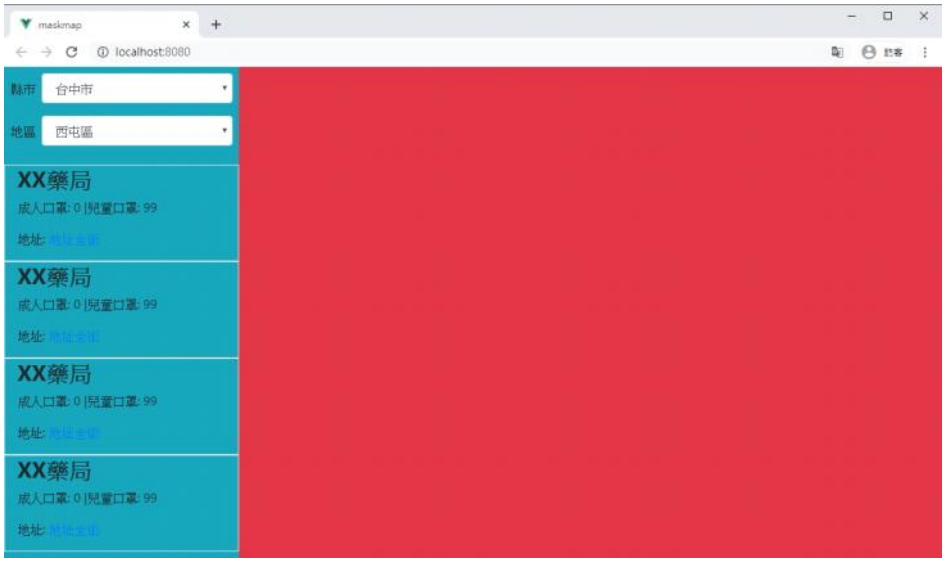


載入檔案 並試寫一段bootstrap 看是否有反應



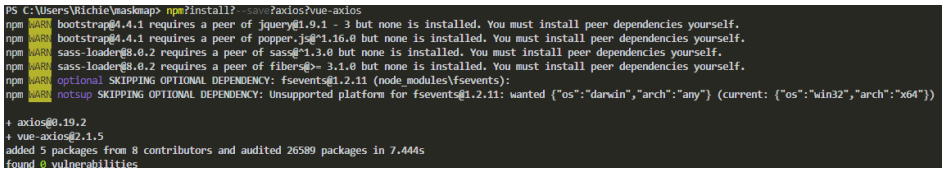
重畫面可發現載入bootstrap成功

5. 繪製"設計畫面"



6. 安裝 vue-axios
   https://www.npmjs.com/package/vue-axios

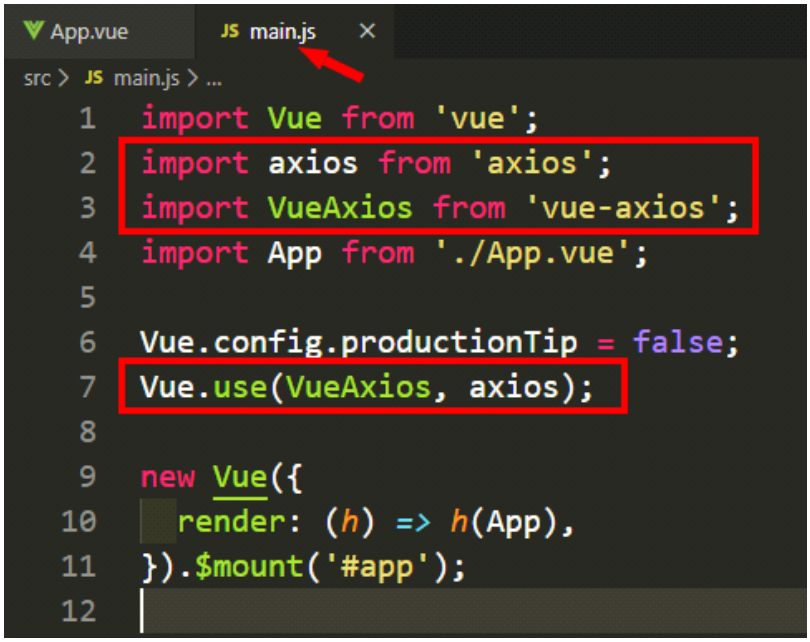
   安裝指令
   `npm install --save axios vue-axios`



   安裝完成後 載入axios
   參考官網說明

```
import Vue from 'vue'
import axios from 'axios'
import VueAxios from 'vue-axios'

Vue.use(VueAxios, axios)
```

   import的順序為外部資料放一起 內部資料放一起!
   而import資料必須要使用不然會產生錯誤!



7. 取得api資料
   https://kiang.github.io/pharmacies/json/points.json

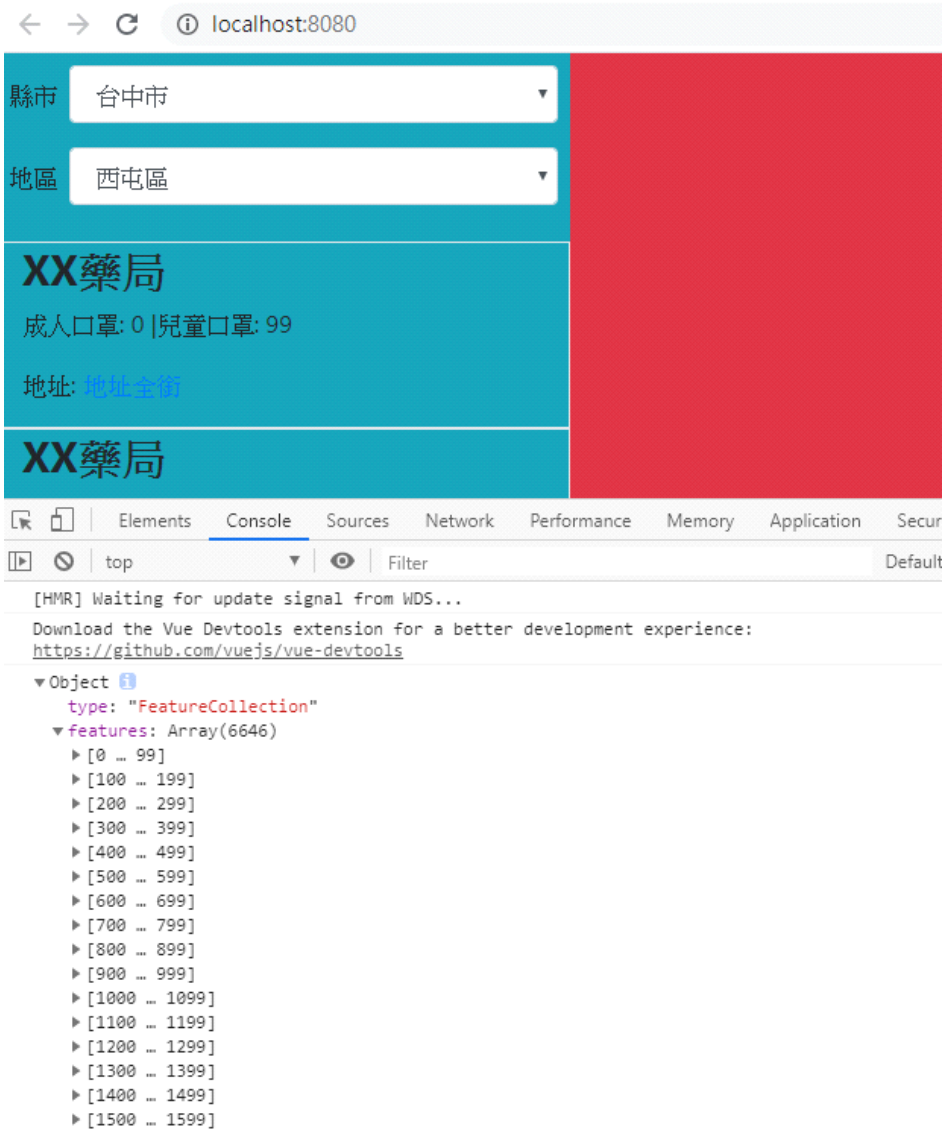
   參考官網語法

```
Vue.axios.get(api).then((response) => {
  console.log(response.data)
})

this.axios.get(api).then((response) => {
  console.log(response.data)
})

this.$http.get(api).then((response) => {
  console.log(response.data)
})
```

   由於載入api為一開始就必須要完成，因此寫在mounted裡面。

```
46                </div>
47              </div>
48              <div class="col-md-9 bg-danger"></div>
49            </div>
50          </div>
51        </div>
52      </template>
53
54      <script>
55
56      export default {
57        name: 'App',
58        components: {
59        },
60        mounted() {
61          const url = 'https://kiang.github.io/pharmacies/json/points.json';
62          this.$http.get(url).then((response) => {
63            console.log(response.data);
64          });
65        },
66      };
67      </script>
68
69      <style lang="scss">
70      @import 'bootstrap/scss/bootstrap';
71      </style>
72
```

可以觀察成功接收資料!



8. 安裝leafletjs

安裝leafletjs 參考 https://leafletjs.com/download.html

Using a Downloaded Version of Leaflet

Inside the archives downloaded from the above links, you will see four things:

- `leaflet.js` - This is the minified Leaflet JavaScript code.
- `leaflet-src.js` - This is the readable, unminified Leaflet JavaScript, which is sometimes helpful for debugging. (The integrity hash for this file is `sha512-6axRrTaCntT2gUQQnqcwJCDOQck41TwHtkTriihNct1L7Ri2J1q0XFYgKJY1do0BTkijrR5X6r4114OKCCLu/A==`)
- `leaflet.css` - This is the stylesheet for Leaflet.
- `images` - This is a folder that contains images referenced by `leaflet.css`. It must be in the same directory as `leaflet.css`.

Unzip the downloaded archive to your website's directory and add this to the `head` of your HTML code:

```
<link rel="stylesheet" href="/path/to/leaflet.css" />
<script src="/path/to/leaflet.js"></script>
```

Using a JavaScript package manager

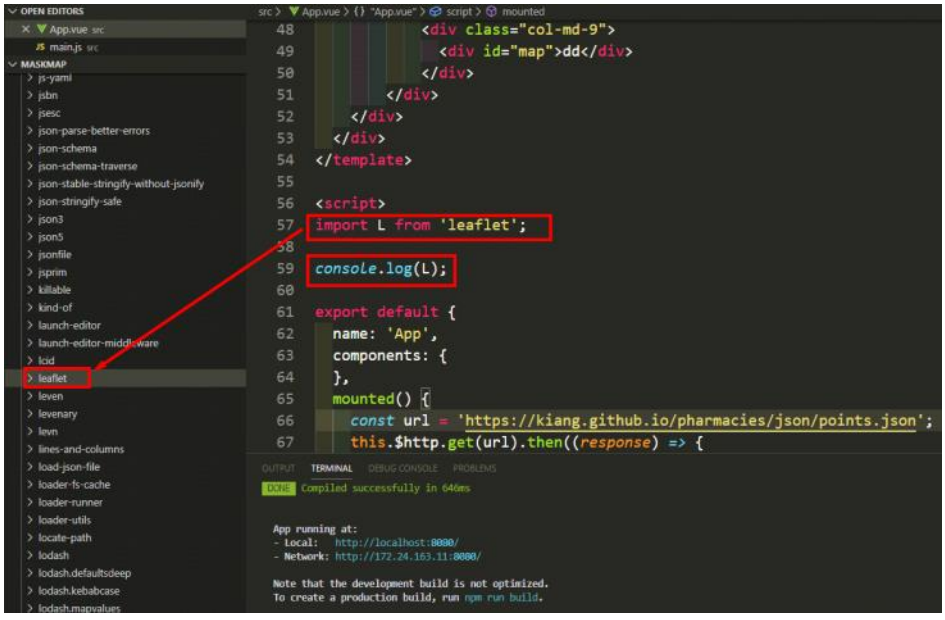
If you use the npm package manager, you can fetch a local copy of Leaflet by running:

```
npm install leaflet
```

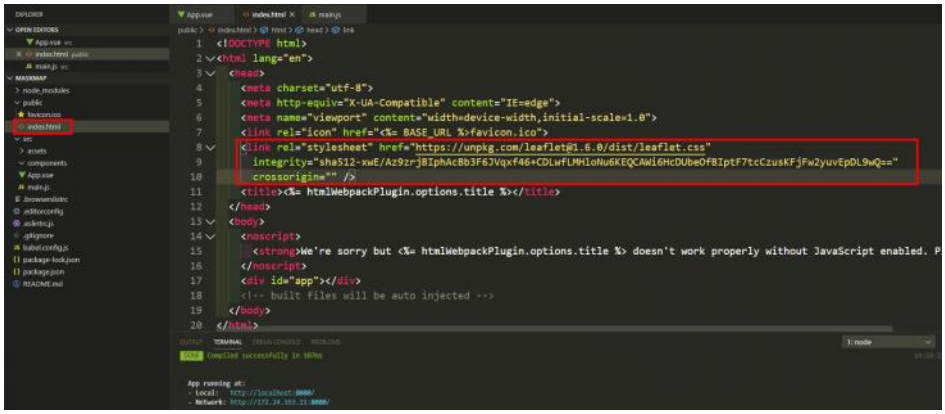
安裝後還是必須手動載入css

```
PS C:\Users\Richie\maskmap> npm install leaflet
npm WARN bootstrap@4.4.1 requires a peer of jquery@1.9.1 - 3 but none is installed. You must install peer dependencies yourself.
npm WARN bootstrap@4.4.1 requires a peer of popper.js@^1.16.0 but none is installed. You must install peer dependencies yourself.
npm WARN sass-loader@8.0.2 requires a peer of sass@^1.3.0 but none is installed. You must install peer dependencies yourself.
npm WARN sass-loader@8.0.2 requires a peer of fibers@>= 3.1.0 but none is installed. You must install peer dependencies yourself.
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.11 (node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.11: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})

+ leaflet@1.6.0
added 1 package and audited 26590 packages in 7.002s
found 0 vulnerabilities
```

import的資料必須要使用否則會產生錯誤 因此先以console.log() 處理!



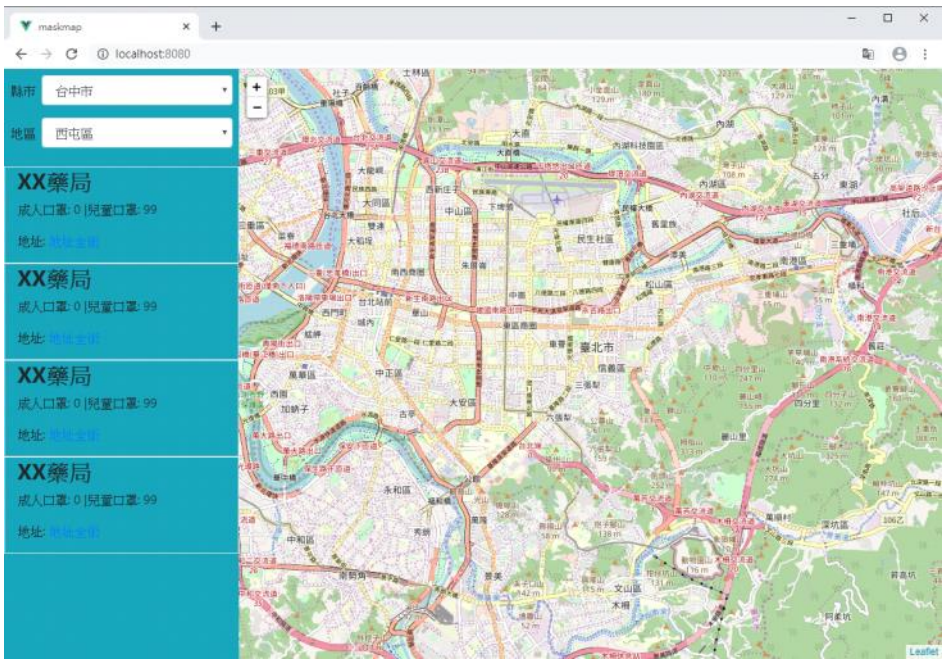
手動載入css




```
<script>
import L from 'leaflet';

console.log(L);
let osmMap = {};

export default {
  name: 'App',
  components: {
  },
  mounted() {
    const url = 'https://kiang.github.io/pharmacies/json/points.json';
    this.$http.get(url).then((response) => {
      console.log(response.data);
    });
    osmMap = L.map('map', {
      center: [25.033976, 121.5623502],
      zoom: 13,
    });
    L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png?{foo}',
      { foo: 'bar' }).addTo(osmMap);
  },
};
</script>
```

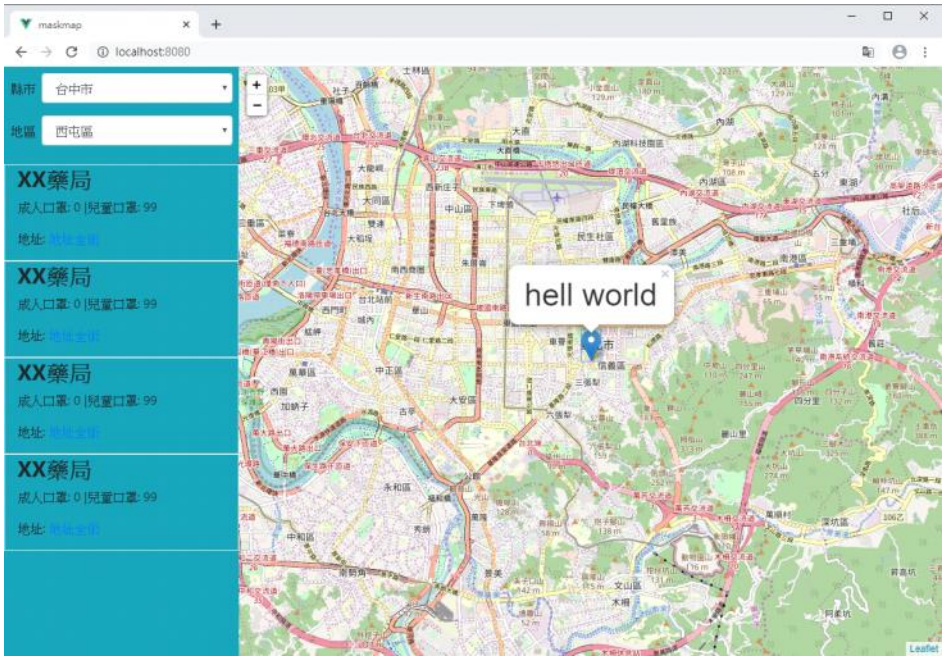
9. 繪製地圖(Open Stree Map)使用leafletjs
   參考 https://leafletjs.com/reference-1.6.0.html

```
<script>
import L from 'leaflet';

console.log(L);
let osmMap = {};

export default {
  name: 'App',
  components: {
  },
  mounted() {
    const url = 'https://kiang.github.io/pharmacies/json/points.json';
    this.$http.get(url).then((response) => {
      console.log(response.data);
    });
    osmMap = L.map('map', {
      center: [25.033976, 121.5623502],
      zoom: 13,
    });
    L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png?{foo}',
      { foo: 'bar' }).addTo(osmMap);
  },
};
</script>
```



加入marker 與 popup

```
export default {
  name: 'App',
  components: {
  },
  mounted() {
    const url = 'https://kiang.github.io/pharmacies/json/points.json';
    this.$http.get(url).then((response) => {
      console.log(response.data);
    });
    osmMap = L.map('map', {
      center: [25.033976, 121.5623502],
      zoom: 13,
    });
    L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png?{foo}',
      { foo: 'bar' }).addTo(osmMap);
    L.marker([25.033976, 121.5623502]).addTo(osmMap).bindPopup('<h1>hell world</h1>');
  },
};
```

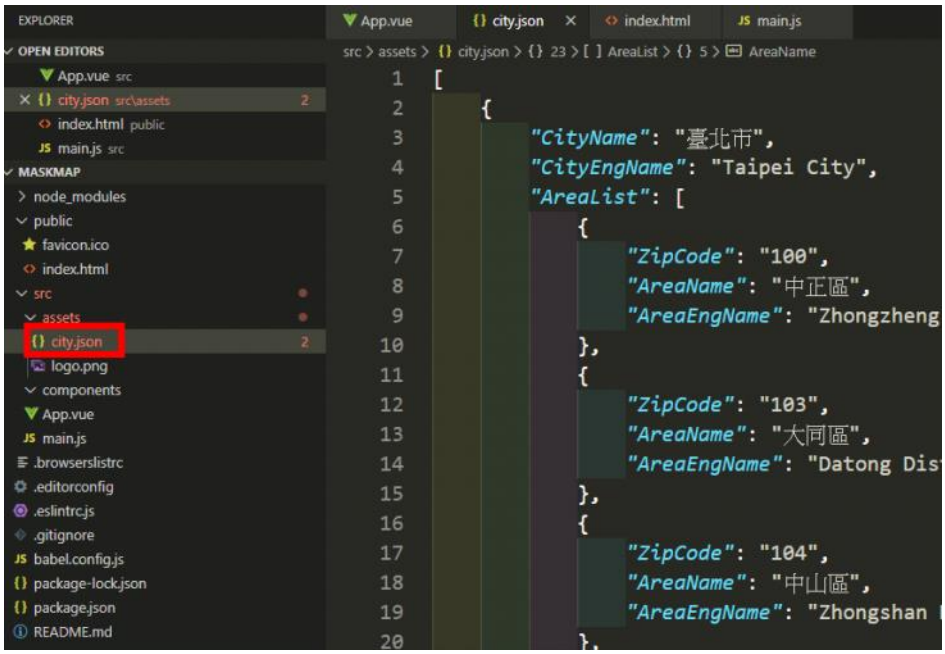


10. 載入台灣各區行政區資料

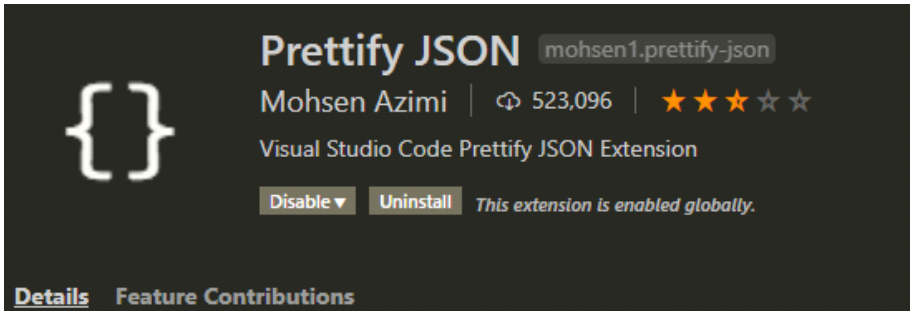
    由於行政區資料變動不太建議將資料放在本地端以減少網路流量!

    https://github.com/donma/TaiwanAddressCityAreaRoadChineseEnglishJSON

    放在src資料夾



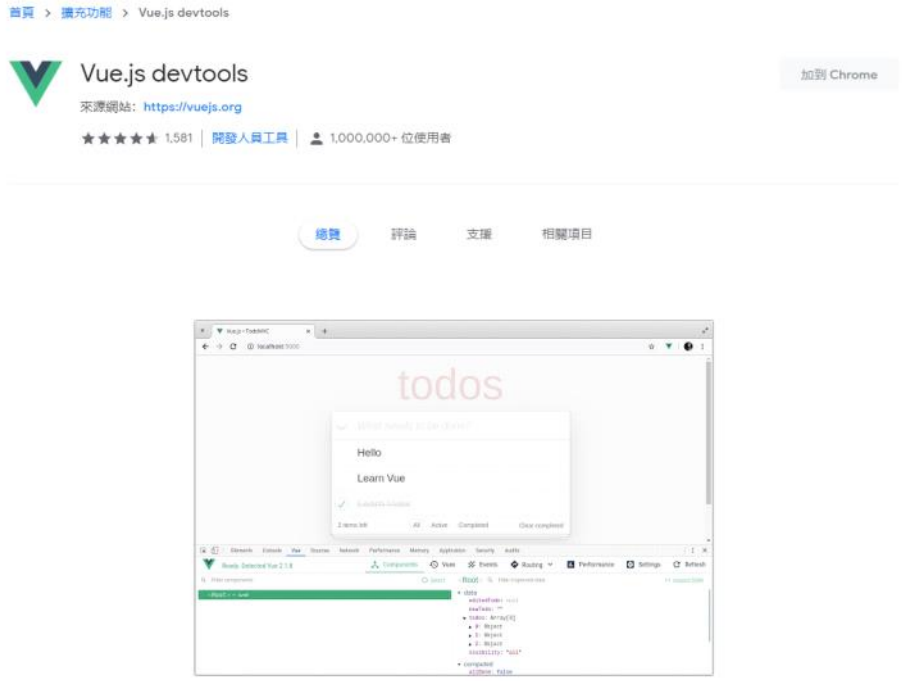
安裝 prettify json 方便閱讀json檔!



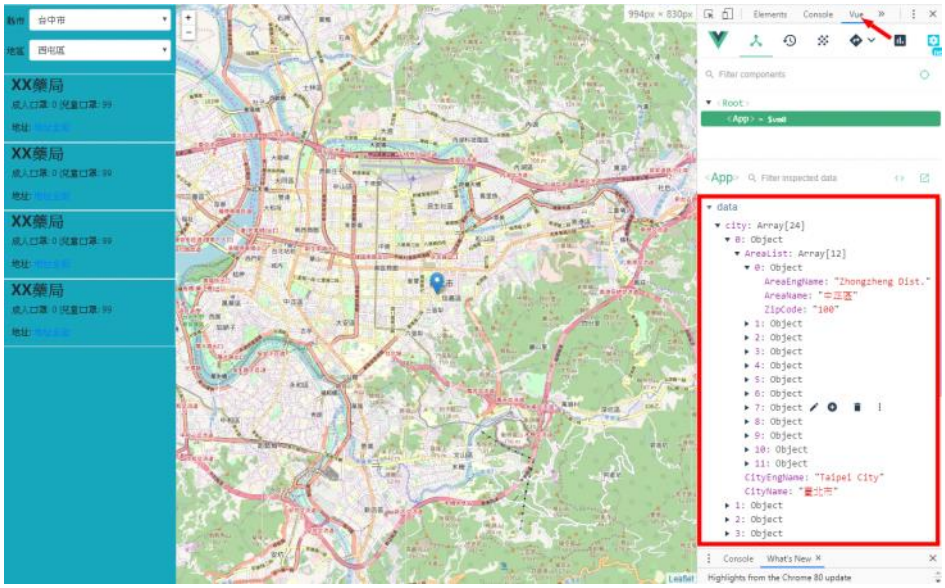
import city.json 並將資料放到vue 的data中

```
<script>
import L from 'leaflet';
import city from './assets/city.json';

console.log(L);
let osmMap = {};

export default {
  name: 'App',
  data: () => ({
    city,
  }),
  components: {
  },
  mounted() {
    const url = 'https://kiang.github.io/pharmacies/json/points.json';
    this.$http.get(url).then((response) => {
      console.log(response.data);
    });
    osmMap = L.map('map', {
```

安裝 Vue tool 開發工具 可以觀察vue裏面的變數



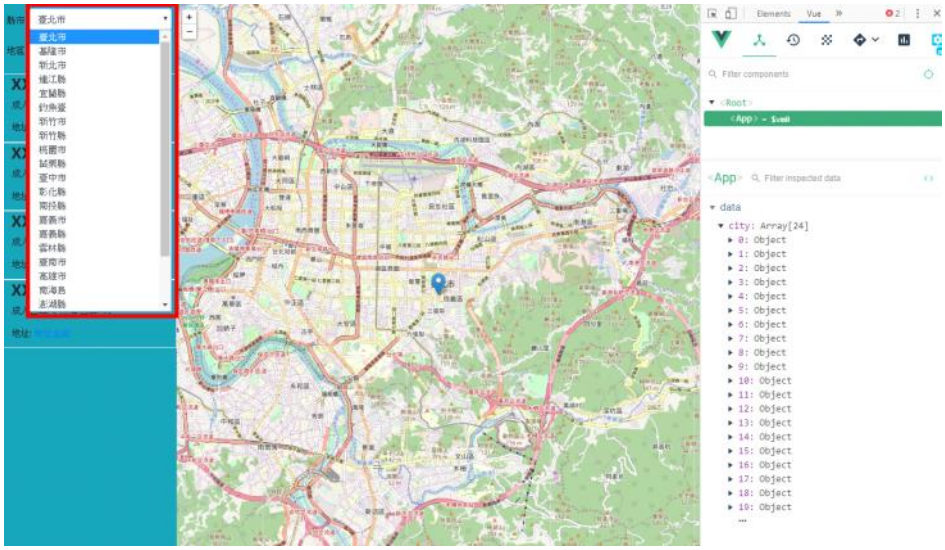
可以發現縣市資料目前已都放入vue裡面



11. 以 v-for 顯示資料顯示並綁定 select 的值

Vue-cli 3 以後使用 v-for 都必須加上 key

```
<div class="form-group d-flex p-2 mb-0">
    <label for="" class="mr-2 col-form-label text-right">縣市</label>
    <div class="flex-fill">
        <select class="form-control">
            <option v-for="item in city"
              :key="item.CityName">{{ item.CityName }}</option>
        </select>
    </div>
</div>
```



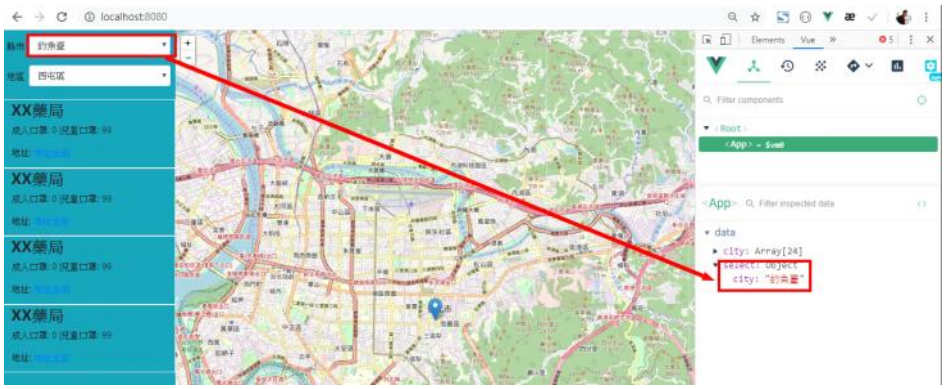
綁定 select 的值 使用 v-model

新增select.city並給於初始值 "臺北市"

```
name: 'App',
data: () => ({
  city,
  select: {
    city: '臺北市',
  },
}),
components: {
},
```

使用 v-model 綁定 select 與 select.city

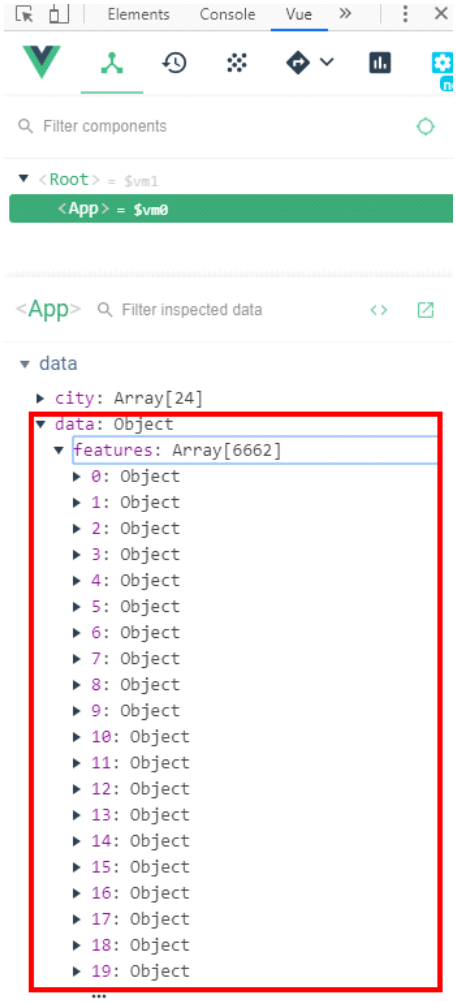
```
<div class="form-group d-flex p-2 mb-0">
    <label for="" class="mr-2 col-form-label text-right">縣市</label>
    <div class="flex-fill">
        <select class="form-control" v-model="select.city">
            <option v-for="item in city"
              :key="item.CityName">{{ item.CityName }}</option>
        </select>
    </div>
</div>
```



12. 將所有藥局資料接收放入vue
    新增變數data 用來接受藥局資料

```
export default {
  name: 'App',
  data: () => ({
    data: [],
    city,
    select: {
      city: '臺北市',
    },
  }),
  components: {
  },
  mounted() {
    const url = 'https://kiang.github.io/pharmacies/json/points.json';
    this.$http.get(url).then((response) => {
      console.log(response.data);
      this.data = response.data;
    });
    osmMap = L.map('map', {
      center: [25.033976, 121.5623502],
      zoom: 13,
    });
    L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png?{foo}',
      { foo: 'bar' }).addTo(osmMap);
    L.marker([25.033976, 121.5623502]).addTo(osmMap).bindPopup('<h1>hell world</h1>');
  },
};
</script>
```

查看是否正確放入vue data



13. 篩選資料並將該區域marker繪製出

    設定setMarker() 並以filter的功能現行測試可否讀取每一筆資料!

```
export default {
  name: 'App',
  data: () => ({
    data: [],
    city,
    select: {
      city: '臺北市',
    },
  }),
  methods: {
    setMarker() {
      const pharmacies = this.data.filter((pharmacy) => (
        console.log(pharmacy)
      ));
      console.log(pharmacies);
    },
  },
  mounted() {
    const url = 'https://kiang.github.io/pharmacies/json/points.json';
    this.$http.get(url).then((response) => {
      // console.log(response.data);
      this.data = response.data.features;
      this.setMarker();
    });
    osmMap = L.map('map', {
      center: [25.033976, 121.5623502],
      zoom: 13,
    });
```

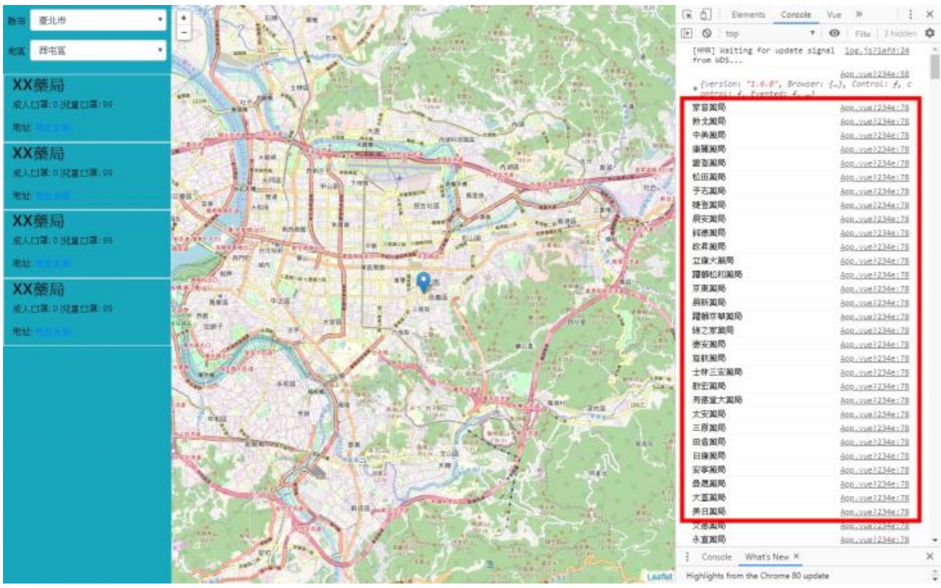
觀察filter是否有將每筆資料抓取

將所有要藥局的地區名稱等於select.city的取出

```
setMarker() {
  const pharmacies = this.data.filter((pharmacy) => (
    pharmacy.properties.county === this.select.city
  ));
  console.log(pharmacies);
},
```

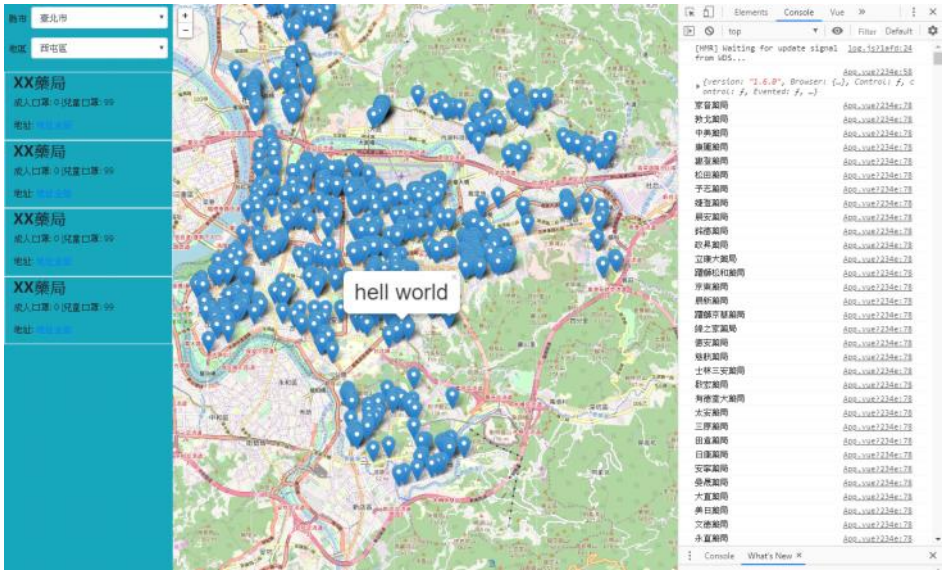
```
[HMR] Waiting for update signal      log.js?1afd:24
from WDS...
                                      App.vue?234e:58
▸ {version: "1.6.0", Browser: {…}, Control: ƒ, cont
  rol: ƒ, Evented: ƒ, …}
▸ []                                  App.vue?234e:75
 vue-devtools                         backend.js:2237
 Detected Vue v2.6.11
                                      App.vue?234e:58
▸ {version: "1.6.0", Browser: {…}, Control: ƒ, cont
  rol: ƒ, Evented: ƒ, …}
                                      App.vue?234e:75
  (708) [{…}, {…}, {…}, {…}, {…}, {…}, {…}, {…},
  {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…},
  {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…},
  {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…},
  {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…},
▾ {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…},
  {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…},
  {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…},
  {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…},
  {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…},
  {…}, {…}, …]
  ▸ [0 … 99]
  ▸ [100 … 199]
  ▸ [200 … 299]
  ▸ [300 … 399]
  ▸ [400 … 499]
  ▸ [500 … 599]
  ▸ [600 … 699]
  ▸ [700 … 707]
    length: 708
  ▸ __proto__: Array(0)
```

使用forEach將pharmacies所有的藥局名稱印出 以測試 forEach的功能!



沒問題的話就可以用同樣的方法將經緯度抓取來繪製marker

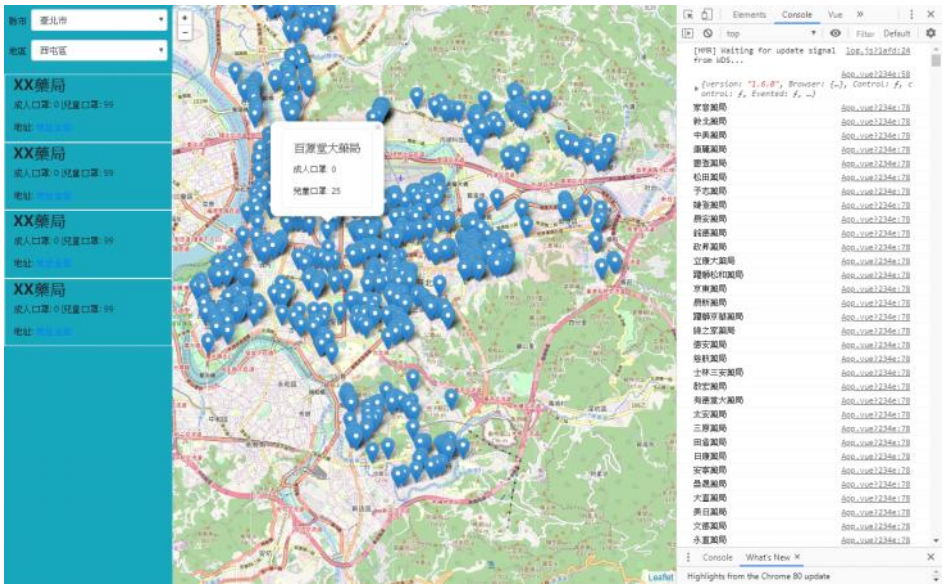
```
pharmacies.forEach((pharmacy) => {
  console.log(pharmacy.properties.name);
  L.marker([pharmacy.geometry.coordinates[1],
    pharmacy.geometry.coordinates[0]]).addTo(osmMap).bindPopup('<h1>hell world</h1>');
});
```



加入bindPopup 動態資料

`${pharmacy.properties.name}`

```
pharmacies.forEach((pharmacy) => {
  console.log(pharmacy.properties.name);
  L.marker([pharmacy.geometry.coordinates[1], pharmacy.geometry.coordinates[0]]).addTo(osmMap)
    .bindPopup(`<div class="card"><div class="card-body">
    <h5 class="card-title">${pharmacy.properties.name}</h5><p class="card-text">
    成人口罩: ${pharmacy.properties.mask_adult}</p><p class="card-text">
    兒童口罩: ${pharmacy.properties.mask_child}</p></div></div><strong>`);
});
```



14. 移除marker 並 配合select顯示不同區域的藥局

```
methods: {
  removeMarker() {
    osmMap.eachLayer((layer) => {
      if (layer instanceof L.Marker) {
        osmMap.removeLayer(layer);
      }
    });
  },
  setMarker() {
    const pharmacies = this.data.filter((pharmacy) => (
      pharmacy.properties.county === this.select.city
    ));
    // console.log(pharmacies);

    pharmacies.forEach((pharmacy) => {
      console.log(pharmacy.properties.name);
      L.marker([pharmacy.geometry.coordinates[1], pharmacy.geo
```

select資料變動時觸發 removeMarker() 與 setMarker()

```
<div class="flex-fill">
    <select class="form-control" v-model="select.city"
      @change="removeMarker(), setMarker()">
        <option v-for="item in city"
          :key="item.CityName">{{ item.CityName }}</option>
    </select>
</div>
```

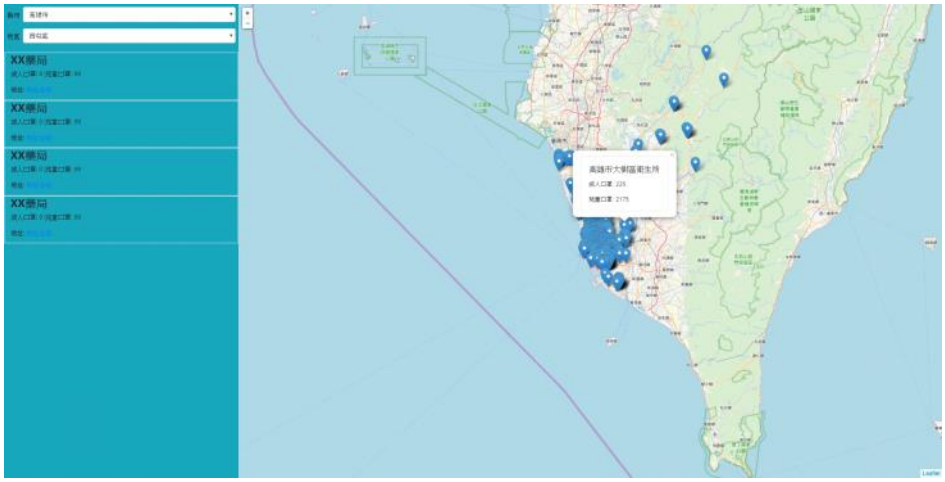
15. 轉移顯示地圖區域

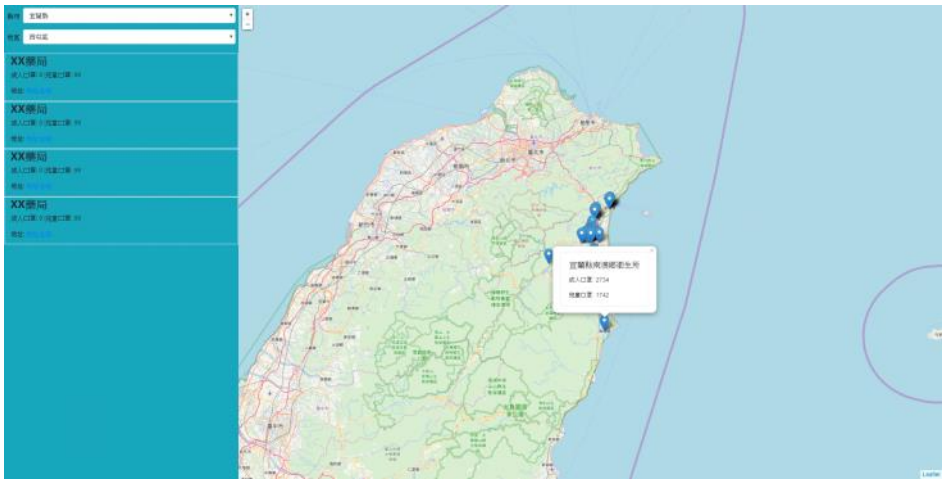
以該區域第一筆資料為地圖的轉心點 呼叫penTo()轉移地圖中心位置

```
penTo(item) {
  osmMap.panTo([item.geometry.coordinates[1], item.geometry.coordinates[0]]);
},
```

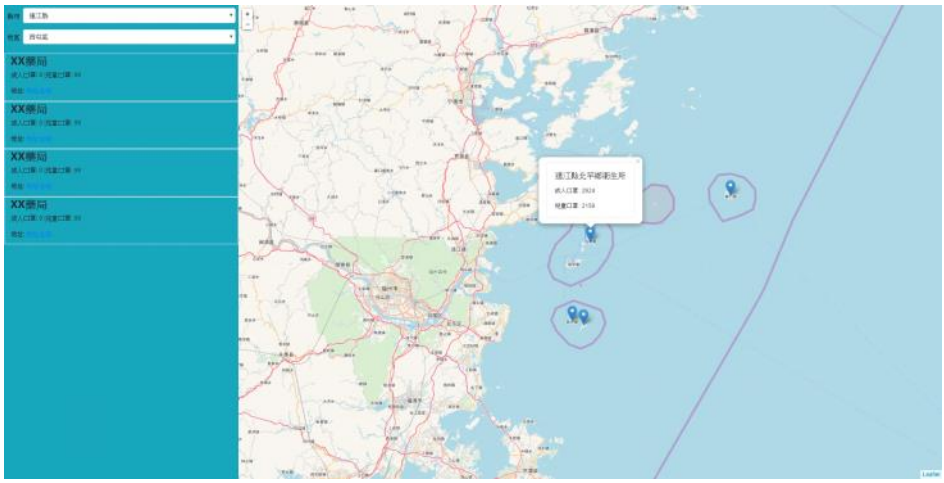
```
setMarker() {
  const pharmacies = this.data.filter((pharmacy) => (
    pharmacy.properties.county === this.select.city
  ));
  // console.log(pharmacies);

  pharmacies.forEach((pharmacy) => {
    console.log(pharmacy.properties.name);
    L.marker([pharmacy.geometry.coordinates[1], pharmacy.geometry.coordinates[0]]).addTo(osmMap)
      .bindPopup(`<div class="card"><div class="card-body">
      <h5 class="card-title">${pharmacy.properties.name}</h5><p class="card-text">
      成人口罩: ${pharmacy.properties.mask_adult}</p><p class="card-text">
      兒童口罩: ${pharmacy.properties.mask_child}</p></div></div><strong>`);
  });
  this.penTo(pharmacies[0]);
},
penTo(item) {
  osmMap.panTo([item.geometry.coordinates[1], item.geometry.coordinates[0]]);
},
```

測試是否會轉移位置!





+



16. 改寫setMarker() 以更新鄉鎮區選單來縮小顯示區域

    updateCity()

    將所挑選的縣市資料過濾後並將該單一縣市的資料存在region，並用來更新鄉鎮區選單。

    updateRegion()

    以所選取的鄉鎮區名稱將所有的藥局過濾，挑選出符合的縣市與鄉鎮區名稱的藥局，儲存在 selectedRegion

    setMarker()

    以selectedRegion將所有的藥局資料顯示出

```
updateCity() {
  const region = this.city.filter((tcity) => (
    tcity.CityName === this.select.city
  ));
  this.region = region[0].AreaList;
},
updateRegion() {
  const pharmacies = this.data.filter((pharmacy) => (
    (pharmacy.properties.county === this.select.city)
      && (pharmacy.properties.town === this.select.region)
  ));
  // console.log(pharmacies);
  this.selectedRegion = pharmacies;
},
setMarker() {
  this.selectedRegion.forEach((pharmacy) => {
    // console.log(pharmacy.properties.name);
    L.marker([pharmacy.geometry.coordinates[1], pharmacy.geometry.coordinates[0]])
      .addTo(osmMap)
      .bindPopup(`<div class="card"><div class="card-body">
      <h5 class="card-title">${pharmacy.properties.name}</h5><p class="card-text">
      成人口罩: ${pharmacy.properties.mask_adult}</p><p class="card-text">
      兒童口罩: ${pharmacy.properties.mask_child}</p></div></div><strong>`);
  });
  this.penTo(this.selectedRegion[0]);
},
```

縣市選單使用 updateCity()

鄉鎮區使用 removeMarker(), updateRegion(), setMarker()

```
<div class="form-group d-flex p-2 mb-0">
    <label for="" class="mr-2 col-form-label text-right">縣市</label>
    <div class="flex-fill">
        <select class="form-control" v-model="select.city"
          @change="updateCity()">
            <option v-for="item in city"
              :key="item.CityName">{{ item.CityName }}</option>
        </select>
    </div>
</div>
<div class="form-group d-flex p-2 mt-0">
    <label for="" class="mr-2 col-form-label text-right">地區</label>
    <div class="flex-fill">
        <select class="form-control" v-model="select.region"
          @change="removeMarker(), updateRegion(), setMarker()">
          <option :value="item.AreaName" v-for="item in region"
            :key="item.AreaName">
            {{ item.AreaName }}
          </option>
        </select>
    </div>
</div>
```

17. 以list-group列出該鄉鎮區的藥局

```
<ul class="list-group">
  <li class="list-group-item" v-for="item in selectedRegion"
    :key="item.properties.name">
    <h3 class="font-weight-bold" @mouseover="popupMaker(item)">
      {{ item.properties.name }}
    </h3>
    <p>成人口罩: {{ item.properties.mask_adult }}
      |兒童口罩: {{ item.properties.mask_child }}</p>
    <p>地址: <a href="" class="link">{{ item.properties.address }}</a></p>
  </li>
</ul>
```

滑鼠滑過時自動popup，並移動到該店家。

```
popupMaker(item) {
  // this.removeMarker();
  L.marker([item.geometry.coordinates[1], item.geometry.coordinates[0]])
    .addTo(osmMap)
    .bindPopup(`<div class="card"><div class="card-body">
    <h5 class="card-title">${item.properties.name}</h5><p class="card-text">
    成人口罩: ${item.properties.mask_adult}</p><p class="card-text">
    兒童口罩: ${item.properties.mask_child}</p></div></div><strong>`)
    .openPopup();
  this.penTo(item);
},
```