

```

1 [HOL]Vue.js Network Programming
2 -----
3 1. Project 준비
4   1)Openweathermap
5     -Visit https://openweathermap.org/
6     -회원 가입
7     -Sign In
8     -화면에서 [API Kyes] click
9     -Key 이름(myapi)입력 후 [Generate] click
10
11   2)Kakao
12     -Kakao 지도 Javascript API 는 키 발급을 받아야 사용할 수 있다.
13     -그리고 키를 발급받기 위해서는 카카오 계정이 필요하다.
14     -키 발급에는 아래 과정이 필요하다.
15       --카카오 개발자사이트 (https://developers.kakao.com) 접속
16       --개발자 등록 및 앱 생성 - Vue-Map
17       --웹 플랫폼 추가: 앱 선택 - [플랫폼] - [Web 플랫폼 등록] - 사이트 도메인 등록
18       --사이트 도메인 등록: [웹] 플랫폼을 선택하고, [사이트 도메인] 을 등록한다. (예: http://localhost:8080)
19       --앱 키 중에서 [JavaScript 키]를 지도 API의 appkey로 사용한다. (예
20         :0ee1d6e75f5488ecdd17e1f60a67bfbf)
21     -상단의 메뉴 중 [문서] > 지도/로컬 API 가이드 > 지도 > JavaScript > 시작하기
22
23     <script type="text/javascript" src="//dapi.kakao.com/v2/maps/sdk.js?appkey=발급받은
24     APP KEY를 넣으시면 됩니다."></script>
25
26 2. Project 생성
27   $ vue create vue-weather --default
28   $ cd vue-weather
29   $ npm run serve
30   $ npm install axios --save
31
32 3. src/App.vue
33
34   <template>
35     <div id="app">
36       <div id="information">
37         <div id="current-weather">
38           <!--현재 날씨 영역 -->
39         </div>
40         <div id="forecast">
41           <!-- 일기 예보 영역 -->
42         </div>
43       </div>
44       <div id="map">
45         <!-- Kakao Map 영역 -->
46       </div>
47     </div>
48   </template>
49
50   <script>
51
52   export default {
53     name: 'App'
54   }
55   </script>
56

```

```
57 <style>
58 #app {
59   position: absolute;
60   top:50%;
61   left:50%;
62   width:1000px;
63   height:600px;
64   margin-top: -300px;
65   margin-left: -500px;
66   border-radius: 3px;
67   background-color: #444;
68   box-shadow: 0px 2px 2px 2px rgba(0,0,0,0.1);
69 }
70 #app:after{
71   content:"";
72   display:block;
73   clear:both;
74 }
75 #app > #information{
76   float:left;
77   width:600px;
78 }
79 #app > #information > #current-weather{
80   margin-top: 100px;
81   background-color: blue;
82   height:100px;
83 }
84 #app > #information > #forecast{
85   padding-top: 10px;
86   background-color: red;
87   height:309px;
88 }
89 #app > #map{
90   float:right;
91   width:400px;
92   height:600px;
93   background-color: green;
94 }
95 hr {
96   width:80%;
97   margin:40px auto;
98   border:none;
99   border-top:1px solid white;
100 }
101 </style>
102
103
104 4. Component 구성하기
105 1)현재 날씨 컴포넌트
106   -src/components/CurrentWeather.vue
107
108   <template>
109     <div id="current-weather">
110       <div class="location">Seoul</div> <!--현재 지역표시 -->
111       <div class="weather">Sunny</div> <!--현재 날씨 표시 -->
112       <div class="temperature">30°C</div> <!-- 현재 온도 표시 -->
113     </div>
114   </template>
```

```
115
116     <script>
117     export default {
118         name : 'current-weather'
119     }
120     </script>
121
122     <style scoped>
123     .location{
124         text-align: center;
125         font-size:40pt;
126         color:white;
127     }
128     .weather{
129         text-align: center;
130         font-size:20pt;
131         font-weight: 100;
132         color:white;
133     }
134     .temperature{
135         text-align: center;
136         font-size:50pt;
137         font-weight: 100;
138         color:white;
139     }
140     </style>
141
142     -App.vue에 등록
143
144     <template>
145     <div id="app">
146         <div id="information">
147             <div id="current-weather">
148                 <current-weather /> <!--3. -->
149             </div>
150             <div id="forecast">
151                 <!-- 일기 예보 영역 -->
152             </div>
153         </div>
154         <div id="map">
155             <!-- Kakao Map 영역 -->
156         </div>
157     </div>
158     </template>
159
160     <script>
161     import CurrentWeather from './components/CurrentWeather.vue' //1.
162     export default {
163         name: 'App',
164         components:{
165             CurrentWeather //2.
166         }
167     }
168     </script>
169
170     <style>
171     body{
172         background-color: #fafafa; <---추가
```

```

173     }
174     #app {
175         position: absolute;
176         top:50%;
177         left:50%;
178         width:1000px;
179         height:600px;
180         margin-top: -300px;
181         margin-left: -500px;
182         border-radius: 3px;
183         background-color: #444;
184         box-shadow: 0px 2px 2px 2px rgba(0,0,0,0.1);
185     }
186     #app:after{
187         content:"";
188         display:block;
189         clear:both;
190     }
191     #app > #information{
192         float:left;
193         width:600px;
194     }
195     #app > #information > #current-weather{
196         margin-top: 100px;
197         /* 이곳에 있던 background-color와 height 속성을 삭제한다. */ <-- 수정
198     }
199     #app > #information > #forecast{
200         padding-top: 10px;
201         background-color: red;
202         height:233px;
203     }
204     #app > #map{
205         float:right;
206         width:400px;
207         height:600px;
208         background-color: green;
209     }
210     hr {
211         width:80%;
212         margin:40px auto;
213         border:none;
214         border-top:1px solid white;
215     }
216     </style>
217
218 2)Kakao Map Component
219   -Kakao developers > 제품 > 지도/로컬 > 문서보기
220
221   -public/index.html
222
223   <!DOCTYPE html>
224   <html lang="en">
225   <head>
226       <meta charset="utf-8">
227       <meta http-equiv="X-UA-Compatible" content="IE=edge">
228       <meta name="viewport" content="width=device-width,initial-scale=1.0">
229       <link rel="icon" href="<%= BASE_URL %>favicon.ico">
230       <title><%= htmlWebpackPlugin.options.title %></title>

```

```

231     <!--다음과 같이 Kakao Map JavaScript를 삽입한다.-->
232     <script type="text/javascript"
      src="//dapi.kakao.com/v2/maps/sdk.js?appkey=0ee1d6e75f5488ecdd17e1f60a67bfbf"
      ></script>
233 </head>
234 <body>
235   <noscript>
236     <strong>We're sorry but <%= htmlWebpackPlugin.options.title %> doesn't work
      properly without JavaScript enabled. Please enable it to continue.</strong>
237   </noscript>
238   <div id="app"></div>
239   <!-- built files will be auto injected -->
240 </body>
241 </html>
242
243 -src/components/KakaoMap.vue
244 --<a href="https://apis.map.kakao.com/web/guide/#more">https://apis.map.kakao.com/web/guide/#more
245 --더 살펴보기 > [Sample 페이지] link click
246 --<a href="https://apis.map.kakao.com/web/sample/">https://apis.map.kakao.com/web/sample/ > 오버레이 > 마커 생성하기 click
247 --아래 코드 복사
248 -----
249   var mapContainer = document.getElementById('map'), // 지도를 표시할 div
250       mapOption = {
251         center: new kakao.maps.LatLng(33.450701, 126.570667), // 지도의 중심좌표
252         level: 3 // 지도의 확대 레벨
253       };
254
255   var map = new kakao.maps.Map(mapContainer, mapOption); // 지도를 생성합니다
256
257   // 마커가 표시될 위치입니다
258   var markerPosition = new kakao.maps.LatLng(33.450701, 126.570667);
259
260   // 마커를 생성합니다
261   var marker = new kakao.maps.Marker({
262     position: markerPosition
263   });
264
265   // 마커가 지도 위에 표시되도록 설정합니다
266   marker.setMap(map);
267
268   // 아래 코드는 지도 위의 마커를 제거하는 코드입니다
269   // marker.setMap(null);
270 -----
271
272 <template>
273   <div>
274     <div id="map-canvas"></div>
275     <!--Kakao Map이 그려질 곳 -->
276   </div>
277 </template>
278
279 <script>
280   export default {
281     name: 'map',
282     mounted() {
283       //index.html에서 정의된 kakao object를 사용하기 위해서는 다음과 같이 주석을 달아야 한다.
284       /* global kakao */
285

```

```
286      //서울 위치
287      var mapContainer = document.getElementById('map-canvas'), // 지도를 표시할 div
288      mapOption = {
289        center: new kakao
290          .maps
291          .LatLng(37.566535, 126.977969), // 지도의 중심좌표
292        level: 3 // 지도의 확대 레벨
293      };
294
295      var map = new kakao
296        .maps
297        .Map(mapContainer, mapOption); // 지도를 생성합니다
298
299      // 마커가 표시될 위치입니다
300      var markerPosition = new kakao
301        .maps
302        .LatLng(37.566535, 126.977969);
303
304      // 마커를 생성합니다
305      var marker = new kakao
306        .maps
307        .Marker({position: markerPosition});
308
309      // 마커가 지도 위에 표시되도록 설정합니다
310      marker.setMap(map);
311
312    }
313  }
314 </script>
315 <style scoped>
316 #map-canvas{
317   width: 400px;
318   height: 600px;
319 }
320 </style>
321
322 -App.vue
323
324 <template>
325   <div id="app">
326     <div id="information">
327       <div id="current-weather">
328         <current-weather />
329       </div>
330       <div id="forecast">
331         <!-- 일기 예보 영역 -->
332       </div>
333     </div>
334     <div id="map">
335       <kakao-map /> <!-- 3. -->
336     </div>
337   </div>
338 </template>
339
340 <script>
341 import CurrentWeather from './components/CurrentWeather.vue'
342 import KakaoMap from './components/KakaoMap.vue' //1.
343 export default {
```

```
344     name: 'App',
345     components:{
346       CurrentWeather,
347       KakaoMap          //2.
348     }
349   }
350 </script>
```

3)일기예보 컴포넌트

-src/components/Forecast.vue

```
355   <template>
356     <div>
357       <ul class="forecasts">
358         <li class="item" v-for="_ in 7" v-bind:key="_"> <!-- 총 7일 -->
359           <div class="weekday">Sun</div> <!--요일 표시 -->
360           <div class="icon">
361             
362             <!-- 일기예보 아이콘 이미지를 불러온다. -->
363           </div>
364           <div class="temperature">30°C</div> <!--온도 표시 -->
365         </li>
366       </ul>
367     </div>
368   </template>
369
370   <script>
371   export default {
372     name : 'forecast'
373   }
374   </script>
375
376   <style scoped>
377   .forecasts{
378     display: flex;
379     width: 80%;
380     padding: 0;
381     margin: 0 auto;
382     align-items: center;
383     justify-content: center;
384   }
385   .forecasts > li.item{
386     position: relative;
387     flex: 1;
388     color: white;
389     text-align: center;
390     list-style-type: none;
391   }
392   .forecasts > li.item:first-child:before{
393     content: "";
394     position: absolute;
395     top: -10px;
396     left: 50%;
397     display: block;
398     width: 6px;
399     height: 6px;
400     margin-left: -3px;
```

```

402     border-radius: 50%;
403     background-color: #fa4444;
404 }
405 .forecasts > li.item > .temperature{
406     margin-top: -8px;
407 }
408 </style>
409
410 -App.vue
411
412 <template>
413   <div id="app">
414     <div id="information">
415       <div id="current-weather">
416         <current-weather />
417       </div>
418       <div id="forecast">
419         <forecast /> <!-- 3. -->
420       </div>
421     </div>
422     <div id="map">
423       <kakao-map /> <!-- Kakao Map 영역 -->
424     </div>
425   </div>
426 </template>
427
428 <script>
429 import CurrentWeather from './components/CurrentWeather.vue'
430 import KakaoMap from './components/KakaoMap.vue'
431 import Forecast from './components/Forecast.vue' //1.
432
433 export default {
434   name: 'App',
435   components:{
436     CurrentWeather,
437     KakaoMap,
438     Forecast //2.
439   }
440 }
441 </script>
442
443
444 5. Network Logic 적용하기
445 1)이제 실제로 데이터를 받아와야 한다.
446 2)현재 날씨 컴포넌트
447 -CurrentWeather.vue
448
449 <template>
450   <div id="current-weather">
451     <div class="location">{{ location }}</div> <!--현재 지역표시 -->
452     <div class="weather">{{ weather }}</div> <!--현재 날씨 표시 -->
453     <div class="temperature">{{ temperature }}</div> <!-- 현재 온도 표시 -->
454   </div>
455 </template>
456
457 <script>
458 import axios from 'axios'
459 export default {

```



```

460     name : 'current-weather',
461     data(){
462         return {
463             location:"",
464             weather: "",
465             temperature : 0,
466             lat : 37.566535, // 서울 위치
467             lng : 126.977969,
468             api_key : '30ebeda71bf08a762f2e1d8a4cb48731'
469         }
470     },
471     mounted(){
472         //axios를 통해 request를 보낸다.
473         axios.get(`http://api.openweathermap.org/data/2.5/weather?lat=${this.lat}&lon=
474             ${this.lng}&appid=${this.api_key}`)
475             .then((response) => {
476                 let data = response.data;
477                 this.location = data.name;
478                 this.weather = data.weather[0].main;
479                 this.temperature = (data.main.temp - 273.15).toFixed(0)
480                 //현재 온도, 절대 온도를 섭씨로 수정한다.
481             })
482             .catch((error) => {
483                 console.log(error)
484             })
485         }
486     }
487 </script>

```

3)Web Browser에서 확인

4)일기예보 컴포넌트

```

491 <template>
492 <div>
493 <ul class="forecasts">
494 <li class="item" v-for="forecast in forecasts" v-bind:key="forecast.dt">
495 <div class="weekday">
496 <!-- timestamp를 Date로 수정 후 필터를 통해 요일로 변환한다. -->
497 {{ convertToDate(forecast.dt) | weekday }}
498 </div>
499 <div class="icon">
500 
503 </div>
504 <div class="temperature">
505 <!-- 절대온도를 섭씨로 수정 -->
506 {{ (forecast.main.temp - 273.15).toFixed(0) }}°C
507 </div> <!--온도 표시 -->
508 </li>
509 </ul>
510 </div>
511 </template>
512
513 <script>
514 import axios from 'axios'

```

```

515 export default {
516   name : 'forecast',
517   data(){
518     return {
519       forecasts : [],
520       lat : 37.566535,
521       lng : 126.977969,
522       api_key : '30ebeda71bf08a762f2e1d8a4cb48731'
523     }
524   },
525   filters : {
526     weekday(date) { //요일 변환 필터
527       let weekdayName = ['Sun', 'Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat']
528       return weekdayName[date.getDay()]
529     }
530   },
531   mounted(){
532     //axios를 통해 일기예보 API를 불러온다.
533     //
534     axios.get(`http://api.openweathermap.org/data/2.5/forecast/daily?lat=${this.lat}&lon=${this.lng}&cnt=7&appid=${this.api_key}`)
535     axios.get(`http://api.openweathermap.org/data/2.5/forecast?lat=${this.lat}&lon=${this.lng}&appid=${this.api_key}`)
536     .then((response) => {
537       let data = response.data;
538       this.forecasts = data.list;
539       //불러온 데이터를 forecasts 데이터에 넣는다.
540     })
541     .catch((error) => {
542       console.log(error)
543     })
544   },
545   methods : {
546     convertToDate(timestamp){
547       //timestamp를 Date로 변환한다.
548       return new Date(timestamp * 1000)
549     }
550   }
551 }
552 </script>

```

6. Event Bus 사용하기

1)main.js에서 빈 Vue instance 생성

```

557 import Vue from 'vue'
558 import App from './App.vue'
559
560 Vue.config.productionTip = false
561
562 Vue.prototype.$bus = new Vue() //Event Bus용 빈 Vue instance 생성
563
564 new Vue({
565   render: h => h(App),
566 }).$mount('#app')
567
568 2)KakaoMap.vue
569

```

```
570 <template>
571   <div>
572     <div id="map-canvas"></div>
573     <!--Kakao Map이 그려질 곳 -->
574   </div>
575 </template>
576
577 <script>
578   export default {
579     name: 'map',
580     mounted() {
581       //index.html에서 정의된 kakao object를 사용하기 위해서는 다음과 같이 주석을 달아야 한다.
582       /* global kakao */
583
584       //서울 위치
585       var mapContainer = document.getElementById('map-canvas'), // 지도를 표시할 div
586         mapOption = {
587           center: new kakao
588             .maps
589             .LatLng(37.566535, 126.977969), // 지도의 중심좌표
590           level: 3 // 지도의 확대 레벨
591         };
592
593       var map = new kakao
594         .maps
595         .Map(mapContainer, mapOption); // 지도를 생성합니다
596
597       //처음 마운트 될때의 Kakao Map 좌표를 전달한다.
598       this.$bus.$emit('sendCoordnate', map.getCenter())
599       // 마커가 표시될 위치입니다
600       var markerPosition = new kakao
601         .maps
602         .LatLng(37.566535, 126.977969);
603
604       // 마커를 생성합니다
605       var marker = new kakao
606         .maps
607         .Marker({position: markerPosition});
608
609       // 마커가 지도 위에 표시되도록 설정합니다
610       marker.setMap(map);
611
612       map.addListener('center_changed', () => {
613         marker.setPosition(map.getCenter())
614       })
615
616       map.addListener('dragend', () => { //Kakao Map Drag가 끝날 경우
617         //EventBus에 kakao Map 좌표 전달
618         this.$bus.$emit('sendCoordnate', map.getCenter())
619         // console.log(map.getCenter())
620       })
621     }
622   }
623 </script>
624
625 3)CurrentWeather.vue
626
627 <template>
```

```

628     <div id="current-weather">
629         <div class="location">{{ location }}</div> <!--현재 지역표시 -->
630         <div class="weather">{{ weather }}</div> <!--현재 날씨 표시 -->
631         <div class="temperature">{{ temperature }}</div> <!-- 현재 온도 표시 -->
632     </div>
633 </template>
634
635 <script>
636 import axios from 'axios'
637 export default {
638     name : 'current-weather',
639     data(){
640         return {
641             location:"",
642             weather: "",
643             temperature : 0,
644             lat : 37.566535, // 서울 위치
645             lng : 126.977969,
646             api_key : '30ebeda71bf08a762f2e1d8a4cb48731'
647         }
648     },
649     created(){
650         //Component가 생성될 때 EventBus와 Event를 $on 한다.
651         this.$bus.$on('sendCoordinate', this.setCoordinate)
652     },
653     methods:{
654         setCoordinate(coordinate){
655             //Event Bus의 Event가 발생할 때 실행되는 메소드.
656             this.lat = coordinate.Ga
657             this.lng = coordinate.Ha
658             axios.get(` http://api.openweathermap.org/data/2.5/weather?lat=\${this.lat}&lon=\${this.lng}&appid=\${this.api\_key}`)
659                 .then((response) => {
660                     let data = response.data;
661                     this.location = data.name
662                     this.weather = data.weather[0].main;
663                     this.temperature = (data.main.temp - 273.15).toFixed(0);
664                 })
665                 .catch((error) => {
666                     console.log(error);
667                 })
668         }
669     },
670     mounted(){
671         //axios를 통해 request를 보낸다.
672         axios.get(` http://api.openweathermap.org/data/2.5/weather?lat=\${this.lat}&lon=\${this.lng}&appid=\${this.api\_key}`)
673             .then((response) => {
674                 let data = response.data;
675                 this.location = data.name;
676                 this.weather = data.weather[0].main;
677                 this.temperature = (data.main.temp - 273.15).toFixed(0)
678                 //현재 온도, 절대 온도를 섭씨로 수정한다.
679             })
680             .catch((error) => {
681                 console.log(error)
682             })
683     }

```

```
684     }  
685   </script>
```