

```
1 [HOL] Vuex
2 -----
3 Lab. Vuex 설치하기
4 1. Project 생성하기
5   1)$vue create basicvuex --default
6   2)VSCoDe에서 Terminal을 연 후, basicvuex 폴더로 이동한다.
7   3)다음 명령을 입력해서 vuex 패키지를 설치한다.
8   4)$ npm install vuex --save
9   5)설치가 완료되면 package.json 파일을 열어 설치를 확인한다.
10      "dependencies": {
11        "core-js": "^3.4.3",
12        "vue": "^2.6.10",
13        "vuex": "^3.1.2"
14      },
15
16 2. store folder 추가
17   1)어느 Component에서나 접근이 가능하도록 하기 위해 Store를 만들려면 src folder 아래에 store 폴더
   를 추가한다.
18   2)src/store folder 생성 후 store.js 파일을 생성한다.
19
20 3. src/store/store.js
21   import Vue from 'vue'
22   import Vuex from 'vuex'
23
24   Vue.use(Vuex)
25
26   export default new Vuex.Store({
27     strict : true, //Vuex 상태가 mutation 핸들러 밖에서 변할 때의 에러, 이를 통해 모든 상태 변화가 명
   확하게 트래킹 되도록 한다.
28     state : {
29       items : [
30         {country:'France', city : 'paris', attraction : '에펠탑', entrance_fee:10},
31         {country:'Italy', city : 'Venezia', attraction : '산마르코 대성당', entrance_fee:0},
32         {country:'Austria', city : 'Salzburg', attraction : '호엔잘쯔부르크성',
   entrance_fee:15.20},
33         {country:'Germany', city : 'Frankfurt', attraction : '뢰머광장', entrance_fee:0},
34         {country:'Nederland', city : 'Amsterdam', attraction : '국립미술관',
   entrance_fee:17.50}
35       ]
36     }
37     //getters:{},
38     //mutations : {},
39     //actions:{}}
40   })
41
42 4. main.js
43   1)Vuex를 사용할 수 있도록 수정한다.
44     import Vue from 'vue'
45     import App from './App.vue'
46     import store from './store/store'
47
48     Vue.use('Vuex');
49     Vue.config.productionTip = false
50
51     new Vue({
```

```
52     render: h => h(App),
53     store
54   }).$mount('#app')
```

## 56 5. App.vue 수정

57 1)Component props를 이용한 결과 화면과 동일한 결과가 나오도록 vuex로 구현하기 위해 수정한다.

```
58     <template>
59       <div id="app">
60         <travel-list></travel-list>
61       </div>
62     </template>
63
64     <script>
65     import TravelList from './components/TravelList.vue'
66
67     export default {
68       name: 'app',
69       components: {
70         TravelList
71       }
72     }
73   </script>
```

```
75 6. src/components/TravelList.vue
```

```
76 <template>  
77   <div id="travel-list">  
78     <p>유럽여행</p>  
79     <ul>  
80       <ul>  
81         <li v-for="item in items" :key="item.attraction"  
82           @click="gocountry(item.country)">  
83             <span>국가 : {{ item.country }}</span>&nbsp;   <br>  
84             <span>도시 : {{ item.city }}</span>&nbsp;   <br>  
85             <span>명소 : {{ item.attraction }}</span>&nbsp;   <br>  
86             <span>요금 : {{ item.entrance_fee | currency }}</span><br>  
87           </li>  
88         </ul>  
89       </ul>  
90     <travellist-details></travellist-details>  
91   </div>  
92 </template>
```

## 94 7. store.js 수정

95 1)TravelList.vue에서 li 에 해당 국가 정보가 있어야 TravelListDetails 컴포넌트에서 해당 국가에 해당하는 정보를 나타낼 수 있다.

96 2)그래서 `store.js`에서 아래의 코드를 추가한다.

```

97     state : {
98         items : [
99             {country:'France', city : 'paris', attraction : '에펠탑', entrance_fee:10},
100            {country:'Italy', city : 'Venezia', attraction : '산마르코 대성당',
entrance_fee:0},
101            {country:'Austria', city : 'Salzburg', attraction : '호엔잘쯔부르크성',
entrance_fee:15.20},
102            {country:'Germany', city : 'Frankfurt', attraction : '뢰머광장',
entrance_fee:0},

```

```

103         {country:'Nederland', city : 'Amsterdam', attraction : '국립미술관',
104           entrance_fee:17.50}
105     ],
106     selectedCountry : "    <--- 새로 추가

```

## 108 8. TravelList.vue 코드 추가

```

109 <script>
110 import TravelListDetails from './TravelListDetails'
111 export default {
112   components : {
113     'travellist-details' : TravelListDetails
114   },
115   methods : {
116     gocountry(inValue){
117       this.$store.state.selectedCountry = inValue
118     }
119   },
120   filters : {
121     currency(value){
122       return new Intl.NumberFormat('de-DE', {style:'currency',
123         currency:'EUR'}).format(value)
124     }
125   },
126   computed : {
127     items(){
128       return this.$store.state.items
129     }
130   }
131 }
132 </script>
133 <style scoped>
134 #travel-list{
135   color:white;
136   background:blue;
137   padding:10px 20px;
138 }
139 #travel-list ul {
140   padding:0;
141   list-style-type:none;
142 }
143 #travel-list li{
144   margin:10px;
145   padding:20px;
146   background:#1565c0;
147 }
148 </style>

```

```
149 9. src/components/TravelListDetails.vue
```

```
150 <template>  
151   <div id="travellist-details">  
152     <p>{{ this.$store.state.selectedCountry }} 여행</p>  
153     <ul>  
154       <li v-for="item in filteredItems" :key="item.attraction">  
155         <span>도시 : {{ item.city }}</span>&nbsp;&nbsp;
```

```
156         <span>명소 : {{item.attraction }} </span>&nbsp;   </span>
157         <span>요금 : {{item.entrance_fee | currency }} </span>
158     </li>
159 </ul>
160 </div>
161 </template>
162 <script>
163 export default {
164   filters : {
165     currency(value){
166       return new Intl.NumberFormat('de-DE', {style:'currency',
167         currency:'EUR'}).format(value);
168     },
169     computed:{
170       filteredItems(){
171         return this.$store.state.items.filter(item => {
172           return item.country == this.$store.state.selectedCountry
173         })
174       }
175     }
176   }
177 </script>
178 <style scoped>
179 #travel-details{
180   color:white;
181   background:teal;
182   padding:10px 20px;
183 }
184 #travel-details ul {
185   padding:0;
186   list-style-type:none;
187 }
188 #travel-details li{
189   margin:10px;
190   padding:20px;
191   background:#00695c;
192 }
193 </style>
```