

src > frontend > src > views > Home.vue > {} "Home.vue" > template > div.home > div#header > router-link.nav-link

```

15 <router-link :to="{ name: 'Test' }">
16   class="nav-link"
17   active-class="active">
18     테스트
19 </router-link>
20 <router-link :to="{ name: 'BoardListPage' }">
21   class="nav-link"
22   active-class="active">
23     게시판
24 </router-link>
25 <router-link :to="{ name: 'VuetifyBoardListPage' }">
26   class="nav-link"
27   active-class="active">
28     뷰티파이
29 </router-link>
30 <router-link :to="{ name: 'VuetifyBoardListPaginationPage' }">
31   class="nav-link"
32   active-class="active">
33     페이지네이션
34 </router-link>

```

↳ Views의 파일 이름.

src > frontend > src > views > Test.vue > {} "Test.vue" > template > div > p

```

1 <template>
2 <div>
3 <h1>뷰 테스트 페이지 입니다.</h1>
4 <button v-on:click="reverseMsg">Reverse Message</button>
5 <p>{{ message }}</p>
6 <b>count: {{ this.$store.state.count }}</b><br>
7 <b>count^2: {{ this.$store.getters.countSquare }}</b><br>
8 <b>weight: {{ this.$store.getters.getWeight }}</b><br>
9 <b>count^weight: {{ this.$store.getters.countWeightSquare }}</b><br>
10 <b>random: {{ this.$store.getters.getRandom }}</b><br>
11 <input type="button" @click="increment()" value="increment"/><br>
12 <input type="button" @click="decrement()" value="decrement"/><br>
13 <input type="button" @click="randomNumber()" value="random"/><br>
14 <!-- Component를 만든다는 것은 결국 나만의 커스텀 HTML 태그를 만든다는 것을 의미한다
15 LocalComponent가 쓰는 num은 현재 Component(Test)에서 사용하는 value와 같다-->
16 <!-- : ???="" v-bind와 같은 의미-->
17 <local-component :num="value"></local-component>
18 <button v-on:click="plus">클릭</button><br>
19 <global-component :initial-counter="counter"></global-component>
20 </div>
21 </template>

```

메소드를 호출하는 리스너 추가

반응성이 생기게 된다~

Data 클래스의 리터라이팅 필요.

이벤트에 따른 처리값

최초의 선언된 이벤트만 실행

하위 컴포넌트와 연결
(components package tree)

동작할 수 있음.

src > frontend > src > views > Test.vue > {} "Test.vue" > template > div > p

```

23 <script>
24 import Vue from 'vue'
25 import cookies from 'vue-cookies'
26 import LocalComponent from "@/components/LocalComponent"
27 import GlobalComponent from "@/components/GlobalComponent";
28 Vue.component(GlobalComponent.name, GlobalComponent)
29 Vue.use(cookies)
30 export default {
31   name: "Test",
32   data () {
33     return {
34       message: "지구를 지켜줘 ~ ♥", 이모티콘 Reverse 하면 오름가 줄겠.
35       counter: 7000,
36       value: 3000
37     }
38   },
39   components: {
40     'local-component': LocalComponent
41   },
42   methods: {
43     reverseMsg: function () {
44       this.message = this.message.split('').reverse().join('')
45     },
46     increment: function () {
47       this.$store.commit('increment') → mutation으로 보냄. commit('함수명', '전달인자')
48       this.$cookies.set('value',
49       this.$store.state.count, '1m')
50     },
51     decrement: function () {
52       this.$store.commit('decrement')
53       this.$cookies.set('value',
54       this.$store.state.count, '1m')
55     },
56     randomNumber () {
57       this.$store.dispatch('generateRandomNumber')
58     }
59   }
60 }

```

* Cookie : 브라우저에 저장되고 서버에는 저장 X

Vue.use(Cookies) 사용 등록해줘야함.

this.\$cookies.set("키", "값", "만료일")

this.\$cookies.get("키") → 쿠키 불러오기

this.\$cookies.remove("키") → 삭제하기

isKey("키") → 특정 쿠키 확인하기

keys().join("\n") → 모든 쿠키 가져오기

keys().forEach(cookie => this.\$cookies.remove(cookie))

→ 모든 쿠키 다 지우기.

* mutation은 실행 / Components에서 then으로
* 비동기 처리. / 콜백함수 실행.

→ Action으로 실행 dispatch('함수명', '전달인자')



vue.config.js

index.js

VuetyBoardListPageForm.vue

Test.vue



51



src > frontend > src > views > Test.vue > {} "Test.vue" > template > div > p

```
59   plus () {  
60     this.value++  
61   }  
62 },  
63 created() {  
64   this.$store.state.count = this.$cookies.get('value')  
65 }
```

변화된 값
변경해서 렌더링

```
1 export default {  
2   // Vue Board  
3   boards: [],  
4   board: null,  
5  
6   // For Test  
7   count: 0,  
8   weight: 3,  
9   random: 0  
10 }
```

가져온 데이터값 초기화.

EXPLORER

OPEN EDITORS

DEMO

- BoardModifyForm.vue
- BoardRead.vue
- BoardRegisterForm.vue
- GlobalComponent.vue U
- HelloWorld.vue
- LocalComponent.vue U
- plugins
 - vuetify.js
- router
 - index.js M
- store
 - actions.js M
 - getters.js M
 - index.js
 - mutation-types.js
 - mutations.js M**
 - states.js M
- views
 - vuetifyPage
 - VuetifyBoardListPage.vue U
 - VuetifyBoardListPaginatio... U
 - About.vue
 - BoardListPage.vue M
 - BoardModifyPage.vue

OUTLINE

TIMELINE

SONARLINT RULES

JAVA PROJECTS

SPRING BOOT DASHBOARD

- demo

mutations.js X

src > frontend > src > store > mutations.js > default > successGenerateRandomNumber

```
1 import {
2   FETCH_BOARD_LIST,
3   FETCH_BOARD
4 } from './mutation-types'
5
6 export default {
7   // for Board
8   [FETCH_BOARD_LIST] (state, boards) {
9     state.boards = boards
10  },
11   [FETCH_BOARD] (state, board) {
12     state.board = board
13  },
14   // for Test
15   increment (state) {
16     state.count++
17  },
18   decrement (state){
19     state.count--
20  },
21   successGenerateRandomNumber (state, payload) {
22     state.random = payload
23  }
24 }
```

변화된 상태값

Actions의 res

변화된 상태값에 따라
값 변화.

src > frontend > src > store > getters.js > default

```
1 export default {  
2   countSquare (state) {  
3     return Math.pow(state.count, 2)  
4   },  
5   countWeightSquare(state ,getters){  
6     return Math.pow(state.count, getters.getWeight)  
7   },  
8   getWeight (state){  
9     return state.weight  
10  },  
11  getRandom (state){  
12    return state.random  
13  }  
14 }
```

getters.js
state
getters

```
1 import {
2   FETCH_BOARD_LIST,
3   FETCH_BOARD
4 } from './mutation-types'
5
6 import axios from 'axios'
7
8 //비동기처리 아무거나 진행이될
9 export default {
10   //커밋하는게 여러개 있을 수 있으므로 {} 로 묶는다
11   fetchBoardList ({ commit }) {
12     return axios.get('http://localhost:7777/boards')
13       .then(res => {
14         //컨트롤러 list의 모든 데이터가 res.data에 있음
15         // 커밋하면 무테이션으로 이동
16         commit(FETCH_BOARD_LIST, res.data)
17       })
18   },
19   fetchBoard ({ commit }, boardNo) {
20     console.log('fetchBoard ' + commit + ', boardNo = ' + boardNo)
21     return axios.get('http://localhost:7777/boards/${boardNo}')
22       .then(res => {
23         console.log('fetchBoard - res: ' + res.data)
24         commit(FETCH_BOARD, res.data)
25       })
26   },
27   generateRandomNumber({ commit }) {
28     axios.get('http://localhost:7777/random')
29       .then((res) => {
30         console.log('res= ' + parseInt(res.data.randomNumber))
31         commit('successGenerateRandomNumber',
32           parseInt(res.data.randomNumber))
33       })
34   }
35 }
```

→ axios 통해 spring과 연결

→ Controller의 Get mapping Value나 같게 해야 함.

→ 무테이션으로 전송.

src > frontend > src > components > LocalComponent.vue > {} "LocalComponent.vue" > style

```
1 <template>
2   <div> {{ num }}</div>
3 </template>
4
5 <script>
6   export default {
7     name: "LocalComponent",
8     props: {
9       num: Number
10    }
11  }
12 </script>
13
14 <style scoped>
15 </style>
```

15
16
17

```
2 <button @click="addCounter">{{ counter }}</button>
3 </template>
4
5 <script>
6 export default {
7   name: "GlobalComponent",
8   // GlobalComponent 연결시 initial-counter
9   props: ['initialCounter'],
10  data: function () {
11    return {
12      counter: this.initialCounter
13    }
14  },
15  methods: {
16    addCounter () {
17      this.counter++
18    }
19  }
20 }
21 </script>
22
23 <style scoped>
24 </style>
```

Local Component와 다르.

여기다 더 큰
예외는 작성가능하다.

src > main > java > com > example > demo > controller > random > RandomController.java > Language Support for Java(TM) by Red Hat > RandomController

```
1 package com.example.demo.controller.random;
2
3 import java.util.Random;
4
5 import com.example.demo.entity.random.RandomNumber;
6 import com.example.demo.service.random.RandomNumberService;
7
8 import org.springframework.beans.factory.annotation.Autowired;
9 import org.springframework.http.ResponseEntity;
10 import org.springframework.stereotype.Controller;
11 import org.springframework.web.bind.annotation.CrossOrigin;
12 import org.springframework.web.bind.annotation.GetMapping;
13 import org.springframework.web.bind.annotation.ResponseBody;
14
15 import lombok.extern.java.Log;
16
17 @Log
18 @Controller
19 @CrossOrigin(origins = "http://localhost:8080", allowedHeaders = "*") → Vue와 연결
20 public class RandomController {
21
22     @Autowired
23     private RandomNumberService randNumService;
24
25     @GetMapping("/random")
26     @ResponseBody
27     public ResponseEntity<RandomNumber> getRandom() {
28         log.info("getRandom()");
29
30         RandomNumber rand = randNumService.getRandom();
31
32         return ResponseEntity.ok(rand); → Response가 되면 통신(rand)
33     }
34 }
```



EXPLORER

OPEN EDITORS

DEMO

- RandomController.java 1, U
- request
- response
- thymeleaf
- vue
- entity
 - random
 - RandomNumber.java U
- vue
 - Board.java
 - Member.java
 - TestMember.java
- game
- nativeinterface
- repository
- service
 - random
 - RandomNumberService.java U
 - RandomNumberServiceImpl.java U
- vue
 - BoardService.java
 - BoardServiceImpl.java
 - MemberService.java

OUTLINE

TIMELINE

SONARLINT RULES

JAVA PROJECTS

SPRING BOOT DASHBOARD

- demo

RandomNumber.java x

src > main > java > com > example > demo > entity > random > RandomNumber.java > RandomNumber

```
1 package com.example.demo.entity.random;
2
3 import java.util.Random;
4
5 import lombok.Getter;
6
7
8 @Getter
9 public class RandomNumber {
10     private Integer randNumber = new Random().nextInt(32) + 1;
11 }
```

src > main > java > com > example > demo > service > random > RandomNumberService.java > RandomNumberService > getRandom()

```
1 package com.example.demo.service.random;
2
3 import com.example.demo.entity.random.RandomNumber;
4
5
6 public interface RandomNumberService {
7     public RandomNumber getRandom();
8 }
```

src > main > java > com > example > demo > service > random > RandomNumberServiceImpl.java > Language Support for Java(TM) by Red Hat > RandomNumt

```
1 package com.example.demo.service.random;
2
3 import com.example.demo.entity.random.RandomNumber;
4 import com.example.demo.repository.random.RandomNumberRepository;
5
6 import org.springframework.beans.factory.annotation.Autowired;
7 import org.springframework.stereotype.Service;
8
9 @Service
10 public class RandomNumberServiceImpl implements RandomNumberService {
11
12     @Autowired
13     private RandomNumberRepository randNumRepo;
14
15     @Override
16     public RandomNumber getRandom() {
17         return randNumRepo.getRandom();
18     }
19 }
```

src > main > java > com > example > demo > service > random > RandomNumberServiceImpl.java > Language Support for Java(TM) by Red Hat > RandomNumt

src > main > java > com > example > demo > repository > random > RandomNumberRepository.java > RandomNumberRepository

```
1 package com.example.demo.repository.random;
2
3 import com.example.demo.entity.random.RandomNumber;
4
5 import org.springframework.stereotype.Repository;
6
7 @Repository
8 public class RandomNumberRepository {
9
10     public RandomNumber getRandom() {
11         return new RandomNumber();
12     }
13 }
14
```

엔티티는 25번

RandomNumberRepository.java

src > frontend > src > views > vuetifyPage > VuetifyBoardListPage.vue > {} "VuetifyBoardListPage.vue" > script > default > data

```
1 <template>
2   <div align="center">
3     <h2>뷰티파이 기반 게시판</h2>
4     <!-- Vuetify Form 배치 -->
5     <vuetify-board-list-page-form :list-array="pageArray"/>
6   </div>
7 </template>
8
9 <script>
10  import VuetifyBoardListPageForm from "@components/vuetify/VuetifyBoardListPageForm";
11  import axios from "axios";
12  export default {
13    name: "VuetifyBoardListPage",
14    components: {
15      VuetifyBoardListPageForm
16    },
17    data () {
18      return {
19        pageArray: []
20      }
21    },
22    created () {
23      axios.get('http://sample.bmaster.kro.kr/contacts')
24        .then(res => {
25          this.pageArray = res.data.contacts
26        })
27        .catch(err => {
28          console.log(err)
29        })
30    }
31  }
32 </script>
33
34 <style scoped>
35 </style>
```

초기값전송
주요이름.

src > frontend > src > views > vuetifyPage > VuetifyBoardListPaginationPage.vue > {} "VuetifyBoardListPaginationPage.vue" > script > default > name

```
1 <template>
2   <div align="center">
3
4   </div>
5 </template>
6
7 <script>
8   export default {
9     name: "VuetifyBoardListPaginationPage"
10  }
11 </script>
12
13 <style scoped>
14 </style>
```


EXPLORER

> OPEN EDITORS

DEMO

- logo.png
- logo.svg
- components
 - vueify
 - VuetifyBoardListPageF... 4, U
 - BoardList.vue
 - BoardModifyForm.vue
 - BoardRead.vue
 - BoardRegisterForm.vue
 - GlobalComponent.vue U
 - HelloWorld.vue
 - LocalComponent.vue U
- plugins
 - vuetify.js
- router
 - index.js M
- store
 - actions.js M
 - getters.js M
 - index.js
 - mutation-types.js
 - mutations.js M
 - states.js M

> OUTLINE

> TIMELINE

> SONARLINT RULES

> JAVA PROJECTS

> SPRING BOOT DASHBOARD

VuetifyBoardListPageForm.vue X

src > frontend > src > components > vuetify > VuetifyBoardListPageForm.vue > {} "VuetifyBoardListPageForm.vue"

```
1 <template>
2   <div>
3     <table>
4       <tr>
5         <th>번호</th>
6         <th>이름</th>
7         <th>연락처</th>
8         <th>주소</th>
9       </tr>
10      <tr v-for="page in paginatedData" :key="page.no">
11        <td>{{ page.no }}</td>
12        <td>{{ page.name }}</td>
13        <td>{{ page.tel }}</td>
14        <td>{{ page.address }}</td>
15      </tr>
16    </table>
17    <div style="margin-top: 1.5rem; text-align: center;
18      width: 15rem; height: 2rem; letter-spacing: 2px;">
19      <button :disabled="pageNum === 0" @click="prevPage">
20        이전
21      </button>
22      <span style="padding: 0 10px;">{{ pageNum + 1 }} / {{ pageCount }} 페이지</span>
23      <button :disabled="pageNum >= pageCount - 1" @click="nextPage">
24        다음
25      </button>
26    </div>
27  </div>
28 </template>
```

0 이면 이전 클릭 안됨

페이지 수와가 count보다 크면 X.

src > frontend > src > components > vuetify > VuetifyBoardListPageForm.vue > {} "VuetifyBoardListPageForm.vue"

```
30 <script>
31 export default {
32   name: "VuetifyBoardListPageForm",
33   data () {
34     return {
35       pageNum: 0
36     }
37   },
38   props: {
39     listArray: {
40       type: Array,
41       required: true
42     },
43     pageSize: {
44       type: Number,
45       required: false,
46       default: 20
47     }
48   },
49   methods: {
50     nextPage () {
51       this.pageNum += 1
52     },
53     prevPage () {
54       this.pageNum -= 1
55     }
56   },
```

```
57 computed: {  
58   pageCount() {  
59     const listLen = this.listArray.length  
60     const listSize = this.pageSize  
61     let page = Math.floor(listLen / listSize)  
62     if (listLen % listSize > 0) {  
63       page += 1  
64     }  
65     return page  
66   },  
67   paginatedData () {  
68     const start = this.pageNum * this.pageSize  
69     const end = start + this.pageSize  
70     return this.listArray.slice(start, end)  
71   }  
72 }  
73 }  
74 </script>  
75  
76 <style scoped>  
77 table {  
78   width: 70%;  
79   border-collapse: collapse;  
80 }  
81 table tr {  
82   border-bottom: 1px solid #333333;  
83 }  
84 </style>  
85
```

전체사양

=20

) 나누어 떨어지지 않음으로 페이지 1개 추가.

0~5

20

20씩 slice 해

자르기

