

# Routing

# Routing

- html의 링크 처리
- index.html을 main template으로 사용한다고 했을 때
- 다양한 화면을 제공할 필요가 있음.
- 주소에 따라 동적인 부분을 불러오는 것을 Routing이라 함

# 새 프로젝트 & vue-router 설치

- `vue create my-cafe`
- `cd my-cafe`
- `npm install --save vue-router`

# component 추가

path	name	component
/	Home	@/components/Home.vue
/menus	Memus	@/components/Menus.vue
/events	Events	@/components/Events.vue
/reviews	Reviews	@/components/Reviews.vue

# component 추가

- 각 파일이 구분되게 제목만 들어가게 모든 컴포넌트 작성

```
1  <template>
2    <div id="app">
3      <h1>Home</h1>
4    </div>
5  </template>
6
7  <script>
8    export default{
9      name: 'Home'
10   };
11  </script>
```

# Router 설정 추가

- src 폴더에 router 폴더를 추가하고 그 안에 router.js 작성

```
1  import Vue from 'vue'
2  import VueRouter from 'vue-router'
3
4  import Home from '@/components/Home.vue'
5  import Menus from '@/components/Menus.vue'
6  import Events from '@/components/Events.vue'
7  import Reviews from '@/components/Reviews.vue'
8
9  Vue.use(VueRouter)
10
11 export default new VueRouter({
12   mode: 'history',
13   routes: [
14     { path: '/', component: Home },
15     { path: '/menus', component: Menus },
16     { path: '/events', component: Events },
17     { path: '/reviews', component: Reviews }
18   ]
19 });
```

# main.js

- 다음과 같이 수정

```
1  import Vue from 'vue'
2  import App from './App.vue'
3
4  → import router from './router/router'
5
6  Vue.config.productionTip = false
7
8  new Vue({
9  →  router,
10     render: h => h(App),
11   }).$mount('#app')
```

# App.vue

- 다음과 같이 수정 (style 부분은 수정 없음)

```
1  <template>
2    <div id="app">
3      <ul>
4        <li><router-link to="/">Home</router-link></li>
5        <li><router-link to="/menus">Menus</router-link></li>
6        <li><router-link to="/events">Events</router-link></li>
7        <li><router-link to="/reviews">Reviews</router-link></li>
8      </ul>
9      <router-view/>
10    </div>
11  </template>
12
13  <script>
14    export default {
15      name: 'App'
16    }
17  </script>
```



# Path parameters

# Path parameter

- 주소의 일부분을 리소스의 아이디로 사용하는 방법으로 RESTful server에서 많이 사용되는 방법이다.
- /users: 전체 사용자
- /users/abc: 사용자 중 아이디가 abc 인 사람

# 방법1: \$route.params

- router.js를 다음과 같이 수정한다.

```
11 export default new VueRouter({
12   mode: 'history',
13   routes: [
14     { path: '/', component: Home },
15     { path: '/menus', component: Menus },
16     { path: '/menus/:menu_id', component: Menus },
17     { path: '/events', component: Events },
18     { path: '/reviews', component: Reviews }
19   ]
20 });
```



# 방법1: \$route.params

- Menus.vue를 다음과 같이 수정한다.
- 주소 창에 다음 주소를 입력해본다.
  - /menus
  - /menus/1
  - /menus/2

```
1  <template>
2    <div id="app">
3      <h1>Menus</h1>
4      <h1>{{ menu_name }}</h1>
5    </div>
6  </template>
7
8  <script>
9    export default{
10     name: 'Menus',
11     computed:{
12       menu_name(){
13         const id = this.$route.params.menu_id
14         if(id=='1')
15           return '아이스 아메리카노'
16         else if(id=='2')
17           return '카페라떼'
18         else
19           return ''
20       }
21     }
22   };
```

# 방법2: props

- router.js를 다음과 같이 수정한다.

```
11 export default new VueRouter({
12   mode: 'history',
13   routes: [
14     { path: '/', component: Home },
15     { path: '/menus', component: Menus },
16     { path: '/menus/:menu_id', component: Menus },
17     { path: '/events', component: Events },
18     { path: '/events/:event_id', component: Events, props: true },
19     { path: '/reviews', component: Reviews }
20   ]
21 });
```

# 방법2: props

- Event.vue를 다음과 같이 수정한다.
- 다음 주소로 접근해본다.
  - /events
  - /events/1
  - /events/2

```
1  <template>
2    <div id="app">
3      <h1>Events</h1>
4      <h1>{{ event_title }}</h1>
5    </div>
6  </template>
7
8  <script>
9  export default{
10    name: 'Events',
11    props: ['event_id'],
12    computed: {
13      event_title(){
14        if(this.event_id=='1'){
15          return '1+1!'
16        } else if(this.event_id=='2'){
17          return 'Free cookies'
18        } else
19          return ''
20      }
21    }
22  };
```

# Query String

# Query String

- Form에서의 입력을 GET Method로 전송할 때 볼 수 있다.
- 주소 다음에 ? 로 시작하며 변수이름=변수값의 형태를 가진다.
- 변수가 2개 이상이면 &로 연결한다.



# Menus.vue

- Menus.vue를 다음과 같이 수정한다.
- 주소창에 다음과 같이 입력해본다.
  - /menus?menu\_id=1

```
1 <template>
2   <div id="app">
3     <h1>Menus</h1>
4     <h1>{{ menu_name }}</h1>
5     <h1>{{searched_menu}}</h1>
6   </div>
7 </template>
8
9 <script>
10 export default{
11   name: 'Menus',
12   computed:{
13     menu_name(){
14       const id = this.$route.params.menu_id
15       if(id=='1')
16         return '아이스 아메리카노'
17       else if(id=='2')
18         return '카페라떼'
19       else
20         return ''
21     },
22     searched_menu(){
23       const id = this.$route.query.menu_id
24       if(id=='1')
25         return '아이스 아메리카노'
26       else if(id=='2')
27         return '카페라떼'
28       else
29         return ''
30     }
31   }
32 };
```

# Query String을 props로 받기

- router.js를 다음과 같이 수정한다.
  - query string을 prop으로 만들어 전달

```
11 export default new VueRouter({
12   mode: 'history',
13   routes: [
14     { path: '/', component: Home },
15     { path: '/menus', component: Menus },
16     { path: '/menus/:menu_id', component: Menus },
17     { path: '/events', component: Events, props: (route) => ({ query_id: route.query.event_id }) },
18     { path: '/events/:event_id', component: Events, props: true },
19     { path: '/reviews', component: Reviews }
20   ]
21 });
```

# Events.vue

- Events.vue를 다음과 같이 수정한다
- 다음 주소로 접근해본다.
  - /events?event\_id=1

```
1  <template>
2    <div id="app">
3      <h1>Events</h1>
4      <h1>{{ event_title }}</h1>
5      <h1>{{ searched_event }}</h1>
6    </div>
7  </template>
8
9  <script>
10 export default{
11   name: 'Events',
12   props: ['event_id', 'query_id'],
13   computed: {
14     event_title(){
15       if(this.event_id=='1'){
16         return '1+1!'
17       } else if(this.event_id=='2'){
18         return 'Free cookies'
19       } else
20         return ''
21     },
22     searched_event(){
23       if(this.query_id){
24         return '2+1'
25       } else
26         return ''
27     }
28   }
29 };
30 </script>
```