

프론트엔드 개발자를 위한 Vue.js 시작하기

# Vue.js </> 시작하기

## 응용 실습 - 앱 기능 구현



한국기술교육대학교  
온라인평생교육원

## 학습내용

- 실습 가이드
- Vuetify

## 학습목표

- 응용 실습에 데이터 Add, Update, Delete 기능을 구현할 수 있다.
- Vuetify를 이용한 애플리케이션의 화면 구성 방법에 대해 설명할 수 있다.

# 실습 가이드

## 실습 가이드

### 1 Add 기능 구현

1 유저 입력 데이터 획득

2 서버 전송

#### User Add

Email address:

First Name:

Last Name:

## 실습 가이드

### 2 Update 기능 구현

1 유저 입력 데이터 획득

2 State에서 데이터 획득, 출력

3 Update 데이터 서버 전송

# 실습 가이드

## 실습 가이드

### 2 Update 기능 구현

#### User Update

Email address:

First Name:

Last Name:

## 실습 가이드

### 3 Delete 기능 구현

#### User List

Id	FirstName	LastName	Email	Action
1	gildong	hong	a@a.com	<input type="button" value="Delete"/> <input type="button" value="Edit"/>
2	hyunjin	ryu	b@b.com	<input type="button" value="Delete"/> <input type="button" value="Edit"/>

# Vuetify

## Vuetify

### 1 Vuetify 기본

#### 1 Vuetify



#### Vuetify 특징

- Material Component를 제공하기 위한 UI Library
- 수려한 화면을 구성하기 위한 Component 및 Directive를 제공해 줌

## Vuetify

### 1 Vuetify 기본

#### 1 Vuetify



#### Vuetify 특징

- Vue.js 3 버전에서 사용하려면 Vuetify 3 버전을 이용해야 함



# Vuetify

## Vuetify

### 1 Vuetify 기본

#### 2 Vuetify 설치

```
>vue add vuetify
```

## Vuetify

### 1 Vuetify 기본

#### 2 Vuetify 설치

- ... 3 버전의 Vue.js 프로젝트가 생성되어야 함
- ... Vue CLI 명령으로 프로젝트에 Vuetify를 설치함
- ... Vuetify 설치 시 선택사항을 제시하며 V3(Alpha)를 선택해야 함

```
? Choose a preset:
  Default (recommended)
  Preview (Vuetify 3 + Vite)
  Prototype (rapid development)
> V3 (alpha)
  Configure (advanced)
```

# Vuetify

## Vuetify

### 1 Vuetify 기본

#### ③ Vuetify 객체 생성

- createVuetify() 함수로 Vuetify 객체를 생성함
- createVuetify() 함수에 Vuetify에서 제공하는 Component, Directive를 등록함

## Vuetify

### 1 Vuetify 기본

#### ③ Vuetify 객체 생성

```
import '@mdi/font/css/materialdesignicons.css'
import 'vuetify/lib/styles/main.sass'
import { createVuetify } from 'vuetify'
import * as components from 'vuetify/lib/components'
import * as directives from 'vuetify/lib/directives'

const vuetify = createVuetify({
  components,
  directives,
})
```

# Vuetify

## Vuetify

### 1 Vuetify 기본

#### 4 Vuetify 객체 적용

- Vuetify는 Plugin으로 제공되는 라이브러리임
- Vue 애플리케이션에 use() 함수로 적용함

```
var app = createApp(Home)
app.use(vuetify)
app.mount('#app')
```

## Vuetify

### 1 Vuetify 기본

#### 5 Component에서 이용

- Vuetify에서 제공되는 Component는 v- 로 시작하는 태그명으로 이용됨

```
<template>
  <div>
    <h3>Vuetify Test</h3>
    <v-btn flat color="error"> Error </v-btn>
  </div>
</template>
```

ERROR



# Vuetify

## Vuetify

### 2 Vuetify Component

#### 1 Avatar

- v-avatar Component 이용
- v-img Component로 지정한 이미지를 아바타 모양으로 출력

## Vuetify

### 2 Vuetify Component

#### 1 Avatar

- 사각형 이미지를 지정해도 기본으로 둥근 모양으로 출력함

```
<v-avatar size="large">
  <v-img src="https://cdn.vuetifyjs.com/images/lists/2.jpg"></v-img>
</v-avatar>
```



# Vuetify

## Vuetify

### 2 Vuetify Component

#### 2 Button

→ v-button Component 이용

→ variant, color, rounded 속성을 이용해 다양한 버튼 이용

```
<v-btn variant="outlined" color="primary"> Outlined Button </v-btn>
<v-btn rounded="pill" color="primary"> Pill Button </v-btn>
```

OUTLINED BUTTON

PILL BUTTON

## Vuetify

### 2 Vuetify Component

#### 3 Icon Button

→ v-button Component에 icon 속성 이용

```
<v-btn icon="mdi-heart" color="primary" />
```



# Vuetify

## Vuetify

### 2 Vuetify Component

#### 4 Rating

→ v-rating Component 이용

```
<v-rating
  v-model="rating"
  bg-color="orange-lighten-1"
  color="blue"
></v-rating>
```



## Vuetify

### 2 Vuetify Component

#### 5 List

→ v-list로 목록 출력

→ v-list 하위에 v-list-item으로 각 항목 출력

# Vuetify

## Vuetify

### 2 Vuetify Component

#### 5 List

→ v-list-item 하위에 v-list-item-avatar, v-list-item-content, v-list-item-icon Component로 항목 구성

## Vuetify

### 2 Vuetify Component

#### 5 List

```

<v-list subheader>
  <v-subheader>Recent chat</v-subheader>
  <v-list-item v-for="chat in recent" :key="chat.title">
    <v-list-item-avatar>
      <v-img :alt="'${chat.title} avatar'" :src="chat.avatar"></v-img>
    </v-list-item-avatar>
    <v-list-item-content>
      <v-list-item-title v-text="chat.title"></v-list-item-title>
    </v-list-item-content>
    <v-list-item-icon>
      <v-icon :color="chat.active ? 'deep-purple accent-4' : 'grey'">
        mdi-message-outline
      </v-icon>
    </v-list-item-icon>
  </v-list-item>
</v-list>

```



Jason Oner



Mike Carlson



Cindy Baker



Ali Connors

# 실습 플러스

## 응용 실습 - 앱 기능 구현

### 실습 플러스

vue-project - Visual Studio Code

## 1 앱 기능 구현 실습

■ VSCode 1.56.1 버전 사용

실습단계
이전에 만들어 놓은 프로젝트 실습에 전체 목록과 관련된 것을 추가, 제거, 수정
Vuex에 store 부분 추가
store → listStore.js 선택
import 받은 이유는? route로 화면을 조정하기 위함
한 명의 유저를 등록하기 위한 addUserAction 함수 추가
입력되어 있는 데이터를 서버에 post 방식으로 전송하고 response의 status 값을 비교하여 이전 화면으로 화면 전환
update와 관련된 action 생성
update → put 방식 사용
URL 수정하고 뒤에 user의 아이디를 알려주어야 함
매개변수를 user가 아니라 id 값으로 받아서 서버에 전달
components 폴더에 EditComponent.vue 파일 생성
입력되어 있는 데이터를 저장하기 위한 변수 선언
EditComponent가 출력된 router의 패스 정보

# 실습 플러스

실습단계
함수 명 수정, updateUserAction이 실행되도록 준비
update 상황일 경우, 데이터가 출력될 수 있게 computed 정의
mounted 함수 이용, 파라미터 값에 id가 정의가 되어있으면 그 값을 먼저 얻어야 함
isAdd 값이 true인 경우 title 문자를 Add로 지정, false인 경우 Update로 지정
user 입력을 받기 위한 input 준비
두 번째 입력받는 값은 firstName으로 지정
마지막 입력받는 값은 lastName으로 지정
버튼이 클릭되면 addUser라는 함수 호출되게 지정
ListComponent.vue의 methods 부분에 함수 추가
목록 화면에서 화면 전환을 목적으로 호출, route로 준비
template에 add로 넘어갈 수 있는 button 추가
tr Tag에 비어있는 td 항목에 delete와 update button 제공
Component 하나가 더 추가됨 → main.js에 import 받음
path 조건이 add로 들어오면 EditComponents가 실행되게 함
routes 조건 등록에 따라 화면 조정
서버가 구동 되어있는 상태에서 Vue 프로젝트 실행
현재 에러로 인해 addUserAction 부분을 '_'로 처리
route 객체가 아니라 store 객체
methods에 deleteUser가 호출됐을 때 store action을 발생시키고자 하는 것이므로 \$ 추가
Vue 명령으로 실행
브라우저에 Component 출력