

프론트엔드 개발자를 위한 Vue.js 시작하기

Vue.js </> 시작하기

응용 실습 - 앱 구조 구축



한국기술교육대학교
온라인평생교육원

학습내용

- 실습 가이드
- 플러그인

학습목표

- **Ajax, Vuex, Vue-router**를 이용하여 애플리케이션의 구조를 구성할 수 있다.
- **플러그인**에 대해 설명할 수 있다.

실습 가이드

실습 가이드

1 응용 실습 Overview



응용 실습 Overview 특징

- 게시판 기능 구현
- Ajax를 이용한 서버 연동
- SPA 적용
- State Management 적용

실습 가이드

2 사용 기술

1

Vue.js

2

vue-router

3

Vuex

4

Axios

실습 가이드

실습 가이드

3 화면 구성

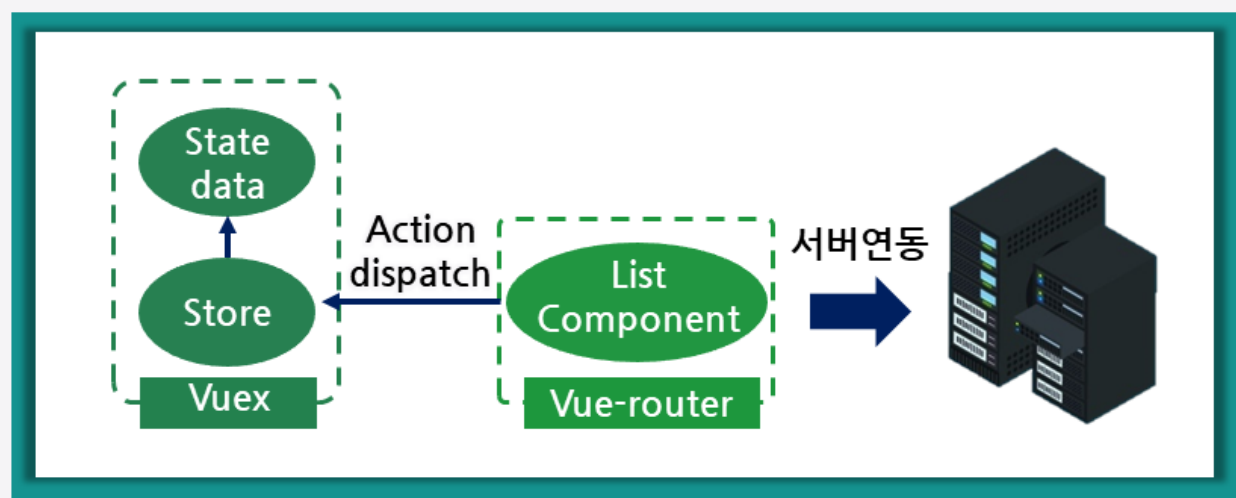
User List

Add User

Id	FirstName	LastName	Email	Action	
1	gildong	hong	a@a.com	Delete	Edit
2	hyunjin	ryu	b@b.com	Delete	Edit

실습 가이드

4 앱 구조



플러그인

플러그인

1 플러그인

① 플러그인

... 플러그인은 Vue의 구성요소를 묶어 다른 곳에서 쉽게 재사용하기 위한 기법

Vue의 구성요소

- Component
- Directive
- Function
- Data ...

플러그인

1 플러그인

① 플러그인

... 여러 애플리케이션에서 공통으로 사용하기 위한 구성요소

... vue-router, Vuex 등은 플러그인으로 만들어짐

플러그인

플러그인

1 플러그인

② 플러그인 선언

- ... 플러그인은 `install` 함수를 가지는 Object Literal로 선언됨
- ... 플러그인이 이용되면 **`install` 함수가 자동 호출됨**
- ... `install` 함수의 첫 번째 매개변수로 플러그인을 이용하는 Vue의 application 객체가 전달됨

플러그인

1 플러그인

② 플러그인 선언

- ... 애플리케이션에서 Global하게 이용하기 위한 구성요소를 등록하여 사용함

```
export default {
  install: (app) => {
    app.component('my-plugin-component', MyHeader)
  }
}
```

플러그인

플러그인

1 플러그인

③ 플러그인 이용

→ 플러그인을 애플리케이션에서 사용하기 위해서는 application 객체의 use() 함수에 플러그인을 등록해야 함

```
var app = createApp(Home)
app.use(MyPlugin)
```

플러그인

1 플러그인

④ Component에서 플러그인 이용

```
<template>
  <div>
    <h3>Plugin Test1</h3>
    <my-plugin-component></my-plugin-component>
  </div>
</template>
```

플러그인

플러그인

1 플러그인

⑤ 플러그인에 Directive 등록

→ Global하게 이용하고자 하는 Directive를 플러그인에 등록

```
export default {
  install: (app) => {
    app.directive("my-font-size", (el) => {
      el.style.fontSize = 20 + "px";
    })
  }
}
```

플러그인

1 플러그인

⑤ 플러그인에 Directive 등록

→ Component에서 플러그인 이용

```
<template>
  <div>
    <p>Hello</p>
    <p v-my-font-size>Hello</p>
  </div>
</template>
```


플러그인

플러그인

1 플러그인

⑥ 플러그인 Option

- Option은 플러그인을 이용하는 곳에서 플러그인에 전달하는 데이터
- install 함수의 두 번째 매개변수로 전달됨

```
app.use(MyPlugin, {
  size: 30
})
```

플러그인

1 플러그인

⑥ 플러그인 Option

- 플러그인 내에서 Option 이용

```
export default {
  install: (app, options) => {
    app.directive("my-font-size", (el) => {
      el.style.fontSize = options.size + "px";
    })
  }
}
```

플러그인

플러그인

1 플러그인

⑦ Mixin

- 여러 Component에서 공통으로 적용되는 구성요소를 플러그인으로 제공
- 애플리케이션의 mixin에 등록해야 함

플러그인

1 플러그인

⑦ Mixin

- 특정 Component에서 사용 설정을 하지 않아도 모든 플러그인이 적용된 애플리케이션의 Component에 적용됨

```
export default {
  install: (app, options) => {
    app.mixin({
      data() {
        return {
          msg: 'Hello Plugin'
        }
      },
      created() {
        console.log("Plugin created....")
      }
    })
  }
}
```

플러그인

플러그인

1 플러그인

⑧ Provide

- 여러 Component에서 공통으로 적용되는 구성요소를 플러그인으로 제공
- application의 provide() 함수로 등록해야 함

플러그인

1 플러그인

⑧ Provide

- 필요한 Component에서 Inject 해서 사용하게 됨

```
export default {
  install: (app, options) => {
    const myClick = () => {
      console.log('myClick....')
    }
    app.provide('myClick', myClick)
  }
}
```

플러그인

플러그인

1 플러그인

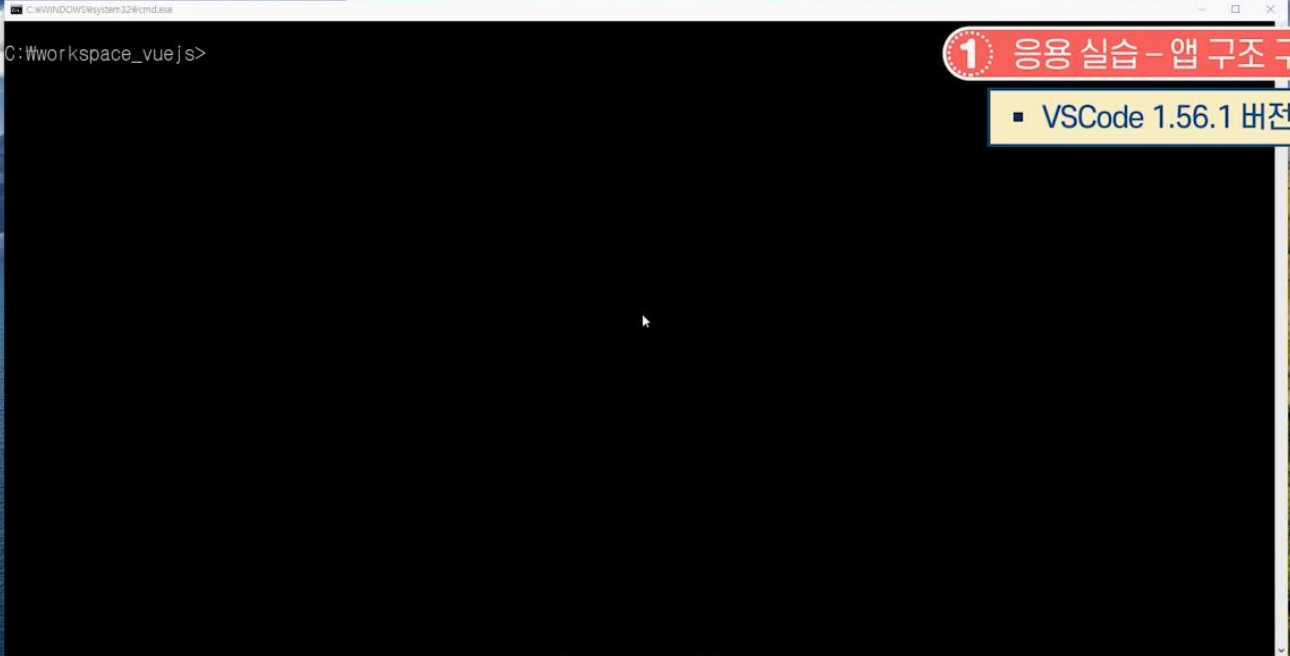
9 Provide 이용

```
import { inject } from 'vue'
myClick = inject('myClick')
```

실습 플러스

응용 실습 - 앱 구조 구축

실습 플러스



실습단계
vue create로 프로젝트 생성
방향키를 움직여서 vue3 선택
전체 프로젝트 Router로 화면 관리, 데이터는 Vuex 이용, 서버 연동 진행, 모듈 추가 설치
npm으로 vue-router 설치
npm으로 vuex 설치
HTTP를 통해서 서버 연동, 사용할 수 있는 모듈 설치
VS Code에서 지금 만들어진 프로젝트 열기
Vuex에서 데이터가 관리되기를 원함
src → store 폴더, listStore.js 파일 생성
axios import 받고 Object Literal 스타일로 store 선언
namespaced 선언, state 선언, 애플리케이션 전역 유지 데이터 users 데이터 유지
앱의 초기 데이터를 위한 서버 연동, 서버 데이터를 가지고 와서 애플리케이션 데이터로 구축할 때 사용하기 위한 getAllUsers 함수 등록
로컬에 있는 서버 이용

실습 플러스

실습단계
서버에 응답이 왔을 때 매개변수가 서버의 응답결과 값이 됨
action에서 commit 함수로 mutation에 있는 함수 명을 명시하여 호출되게 함, 두 번째 매개변수가 서버에서 받은 데이터
첫 번째 매개변수는 애플리케이션에서의 state 객체
Component에서 state 데이터를 쉽게 획득하기 위한 getter 선언
Component에서 id 값이 전달됨, 해당되는 유저를 전체 목록에서 찾아내서 return 시키기 위한 getter
Components 폴더 → ListComponent.vue 생성
Vuex를 이용하기 위한 import 먼저 선언하고 computed 함수 선언
listStore에서 namespace로 구분되는 store에 users 데이터 획득, template에 활용되면서 화면에 찍힘
Component의 LifeCycle 함수를 이용하여 액션 발생시킴
methods 옵션 추가, 비워둠
목록 화면을 구성하기 위한 테이블 구성
table 첫 줄에 thead Tag 추가
action, 비워둠
div Tag로 묶고 전체 화면이 Router에 의해 나올 수 있게 처리
script 부분 비움
style 제거함
main.js 수정
ListComponent를 등록
초기 값이 ListComponent가 나오도록 지정됨
Router 객체 생성
라우팅 정보가 들어가 있는 객체 선언
앱 생성
애플리케이션 객체에 router와 store 등록
중요! 제공된 실습 파일에서 다운받은 서버 이용
서버에 필요한 모듈 설치
Server.js 파일 : 서버 파일
node 명령으로 실행 → node server.js

실습 플러스

실습단계
vue-project 실행 → npm run serve 명령
브라우저에 결과 화면 출력