



Vuex

Vuex란?

- vue.js 애플리케이션에 대한 **상태 관리 패턴 + 라이브러리**
- 애플리케이션의 모든 컴포넌트에 대한 중앙 집중식 저장소 역할을 하며
- 예측 가능한 방식으로 상태를 변경할 수 있다

Vuex 도입의 장점

- 여러 개의 컴포넌트가 데이터를 공유
- 데이터 상태와 관련된 처리를 공통화
- 큰 상태 관리도 모듈을 사용해 간단하게 분리

상태 관리 패턴

- **상태** : 앱을 작동하는 원본 소스
- **뷰** : 상태의 선언적 매핑
- **액션** : 뷰에서 사용자 입력에 대해 반응적으로 상태를 바꾸는 방법



문제!

- 여러 뷰는 같은 상태에 의존한다
 - 지나치게 중첩된 컴포넌트를 통과하는 prop는 장황할 수 있으며
 - 형제 컴포넌트에서는 작동하지 않는다
- 서로 다른 뷰의 작업은 동일한 상태를 반영해야 할 수 있다

- 직접 부모/자식 인스턴스를 참조하거나 이벤트를 통해 상태의 여러 복사본을 변경 및 동기화하려는 등의 해결 방법을 사용해야 한다

→ 이러한 패턴은 모두 부서지기 쉽고 유지보수가 불가능한 코드로 빠르게 변경됨



컴포넌트에서 공유된 상태를 추출하고 이를 전역 싱글톤으로 관리해야 한다

- 모든 Vuex 애플리케이션의 중심에는 **store**가 있다
- "저장소"는 기본적으로 애플리케이션 **상태**를 보유하고 있는 컨테이너이다
- Vuex 저장소가 일반 전역 개체와 다른 점
 - Vuex store는 반응형이다
 - Vue 컴포넌트는 상태를 검색할 때 저장소의 상태가 변경되면 효율적으로 대응하고 업데이트한다
 - 저장소의 상태는 직접 변경할 수 없다
 - 저장소의 상태를 변경하는 유일한 방법은 명시적인 **커밋을 이용한 변이**이다
 - 이렇게 하면 모든 상태에 대한 추적이 가능한 기록이 남을 수 있으며
 - 툴을 사용하여 앱을 더 잘 이해할 수 있다

핵심 컨셉

상태 (state)

- 스토어에서 관리하고 있는 상태
- mutation 이외의 장소에서는 변경하면 안 됨

Getters

- state를 추출하기 위한 산출 데이터
- 산출 속성과 다르게 매개변수를 받을 수 있지만 setter 기능은 없다
- 메서드의 매개변수로 사용할 수 있는 속성 또는 메서드를 전달할 수 있다

변이 (mutation)

- 스테이트를 변경할 수 있는 방법
- 매개변수로 받는 정보
 - state 스테이트
 - payload 커밋에서 전달된 매개변수
- 변이는 무조건 동기적이어야 한다
- 커밋

액션

- 비동기 처리를 포함할 수 있는 메서드
- 데이터 가공 또는 비동기 처리를 실시한 후, 그 결과를 뮤테이션으로 커밋
- 여러 기능을 사용하기 위해 내용을 확인하거나 상태를 변경하는 등의 세부적인 처리는 모두 액션 내부에서 하는 것이 좋다
- 오류도 뮤테이션이 아니라 액션에서 throw하는 것이 좋다
- 디스패치
 - 등록되어 있는 액션을 호출하는 인스턴스 메서드
 - 다른 액션 내부에서도 사용할 수 있다
- 비동기 처리가 없는 경우 액션을 거치지 않고 직접 커밋할 수도 있다
- 하지만 여러 사람들이 함께 개발할 때 혼란을 주지 않도록 모두 dispatch로 통일하는 것을 추천한다

모듈

Vuex의 규칙

- 애플리케이션 레벨의 상태는 스토어로 관리하기
- 상태를 변경하는 것은 뮤테이션 내부에서만 수행하기
- 비동기 처리는 커밋하기 전에 완료해 두기