

# 8/3 ~ 8/7 5주차 -2

## Vuex-persistedstate에 대해서

Vue.js 화면을 새로고침 해도 안전한 방법이 있다.

Vue.js로 프로젝트를 개발 하는 경우 화면을 이동 할 때 일반 웹페이지처럼 페이지 전체를 새로고침하는 방식이 아니라 SPA와 같이 javascript로 필요한 부분만 교체하는 식으로 개발하게 된다.

이런 SPA방식의 개발은 모던한 웹사이트 개발에서 많은 잇점을 주고 있는 것이 사실이지만 몇가지 불편한 부분이 있다.

보통 모바일 웹이나 앱일 경우는 크게 착각하지 않는데 일반 웹브라우저에서 띄운 경우 습관적으로 뒤로가기 버튼을 클릭할 때 화면이 새로고침이 되어 버린다.

또는 기타 다른 이유로 새로고침하는 경우가 발생하게 되는데 이럴 때 지금까지 vuex의 store에 가지고 있거나 계산되어져 있던 모든 값들이 초기화 되어버리는 상황이 발생한다.

이런 불편함을 해소하기 위해 `localStorage`를 이용하여 값들을 다시 살려내주는 기능을 하는 것이 `vue-persistedstate` 플러그인 이다.

### vue-persistedstate의 원리

vuex를 사용하는 프로젝트는 프로젝트 전체에서 사용되는 변수를 store의 state에 저장하고 사용하게 된다.

그러면 vue-persistedstate는 이 state에 저장된 변수와 값을 그대로 웹브라우저의 localStorage에 업데이트를 해준다.

다시 말하면, state와 localStorage를 지속적으로 동기화를 해주는 역할을 한다.

localStorage에 등록된 내용은 쿠키나 세션처럼 화면을 새로고침 해도 없어지지 않기 때문에 새로 화면이 로딩을 하게되면 localStorage에 있는 내용을 state에 다시 동기화를 시켜주게 된다. 그러면 다시 값들이 살아나게 된다.

다만 고려해야할 점은 이 state의 변수를 너무 많이 사용하게 되면 프로그램의 퍼포먼스가 떨어진다.

## vue-persistedstate 설치하기

- npm install —save vue-persistedstate

## vue-persistedstate 적용하기

```
import createPersistedState from "vuex-persistedstate";

const store = new Vuex.Store({
  // ..
  plugins: [createPersistedState()],
});
```

## 선택한 모듈만 설정하는법

- Vue프로젝트가 커서 모든 state값을 localStorage에 저장할 경우 퍼포먼스의 문제가 발생할 수 있으므로 필요한 모듈만 선택해서 저장하는 것이 좋습니다.

아래의 예제는 auth와 cart 모듈의 state값만 저장되고 복구됩니다.

```
import Vue from "vue";
import Vuex from "vuex";
import createPersistedState from "vuex-persistedstate";

import { Auth } from "./auth";
import { Cart } from "./cart";
import { Products } from "./products";

Vue.use(Vuex);

export default new Vuex.Store({
  modules: {
    auth: Auth,
    cart: Cart,
    products: Products,
  },
  plugins: [
    createPersistedState({
      // 주목! : 여기에 쓴 모듈만 저장됩니다.
      paths: ["cart", "auth"],
    }),
  ],
});
```

copy javascript