

Persisting the Vuex Store

The Vuex store is not intended for long term storage of information (a database is much more suitable for that) but there are reasons you might want to persist your store.

One big one is that with Vuex, the store resets every time you refresh the page, so if a user reloads the page, they lose all their state info - which might cause issues in their use of the application.

Persisting the store solves this issue, as the store loads the persisted values rather than resetting on a page reload.

Setting up your store Vuex to persist is fairly simple.

The first thing to do is to add the **vuex-persistedstate** plugin to your Vue dependencies by issuing this command in the folder where your **package-lock.json** file lives:

```
npm install --save vuex-persistedstate
```

Once you have done this, open up your **index.js** file in the **store** directory.

At the top, where the other imports are, add this code

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

Then add this property as a separate section (outside of **state**, **mutations**, etc)

```
plugins: [createPersistedState()]
```

If you add it at the end, it would look something like the image below.

```
    localStorage.setItem('user', JSON.stringify(user));  
  },  
  LOGOUT(state) {  
    localStorage.removeItem('token');  
    localStorage.removeItem('user');  
    state.token = '';  
    state.user = {};  
    axios.defaults.headers.common = {};  
  }  
],  
  plugins: [createPersistedState()],  
});  
return store;  
}
```

That's all you should need to do for a basic configuration. And, just like that, your store will be persisted!