

Vuex

폴더 구조

Module별 폴더로 store를 나누었으며, state, getters, mutations, actions또한 개별로 관리하기 수월하게 분리 하였습니다.

또한 각 mutations와 actions의 type별 handler를 분리하여, 공동 작업자가 전체 애플리케이션에서 어떤 변이가 가능한지 한눈에 파악할 수 있습니다.

```
store
├── actionTypes.js
├── index.js
├── mutationsType.js
├── modules
│   ├── common
│   │   ├── actions.js
│   │   ├── getters.js
│   │   ├── index.js
│   │   ├── mutations.js
│   │   └── state.js
│   └── modules1
│       ├── actions.js
│       ├── getters.js
│       ├── index.js
│       ├── mutations.js
│       └── state.js
```

state.js

```
const state = {
  loadingComplete: false, // 로딩 완료 상태
};

export default state;
```

getters.js

```
const getters = {
  /**
   * @description 로딩 완료 상태 조회
   * @returns state.screen.loadingComplete 로딩 완료 상태
   */
  loadingComplete: (state) => state.loadingComplete,
};

export default getters;
```

mutations.js

```
import { commonMutationType } from '@store/mutationsType';

const mutations = {
  /**
   * @description 로딩 상태 설정
   * @param state
   * @param payload 전달 받은 값: Boolean
   */
  [commonMutationType.SET_LOADING_COMPLETE](state, payload) {
    try {
      if (typeof payload === 'boolean') {
        state.loadingComplete = payload;
      }
    } catch (e) {
      console.error(`[commonMutationType.SET_LOADING_COMPLETE] error: ${e}`);
    }
  },
};

export default mutations;
```

actions.js

```
import { commonActionType } from '@store/actionsType';
import { commonMutationType } from '@store/mutationsType';

const actions = {
  /**
   * @description action 설명
   * @param commit
   * @param payload 전달 받은 값
   */
  [commonActionType.SET_LOADING_COMPLETE]: (context, payload) => {
    context.commit(commonMutationType.SET_LOADING_COMPLETE, payload);
  },
};

export default actions;
```

```

    [commonActionType.ACTION_SAMPLE]({ commit }, payload) {
      commit(commonMutationType.SET_LOADING_COMPLETE, true);
    },
  };

export default actions;

```

mutationsType.js

```

/**
 * @description store의 각 module별 mutation type 관리
 */

const commonMutationType = {
  SET_LOADING_COMPLETE: 'SET_LOADING_COMPLETE', // 로딩 상태
};

const modules1MutationType = {
  SET_MODULE1: 'SET_MODULE1',
};

export { commonMutationType, modules1MutationType };

```

actionsType.js

```

/**
 * @description store의 각 action별 action type 관리
 */

const commonActionType = {
  ACTION_SAMPLE: 'ACTION_SAMPLE',
};

const modules1ActionType = {
  ACTION_SAMPLE: 'ACTION_SAMPLE',
};

export { commonActionType, modules1ActionType };

```

dispatch, commit 사용 예

```
import { commonMutationType } from '@store/mutationsType';
import { commonActionType } from '@store/actionsType';

export default {
  mounted() {
    this.$store.dispatch(commonActionType.ACTION_SAMPLE);
    this.$store.commit(commonMutationType.SET_LOADING_COMPLETE);
  },
};
```