

```
import { createSlice, createAsyncThunk } from "@reduxjs/toolkit";
import { cloneDeep } from "lodash";
import NhapKhoService from "services/sale/NhapKhoService";
import { initNhapKhoCt } from "views/app-views/sale/managements/QuanLyKho/NhapKho";
```

```
export const getNhapKhoCTByManx = createAsyncThunk(
  "nhapkho/getNhapKhoCTByManx",
  async (data, { rejectWithValue }) => {
    try {
      const response = await NhapKhoService.getNhapKhoCTByManx(data);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);
```

```
export const addNewNhapKhoCT = createAsyncThunk(
  "nhapkho/addNewNhapKhoCT",
  async (data, { rejectWithValue }) => {
    try {
      const { onSuccess } = data;
      const payload = cloneDeep(data);
      delete payload.onSuccess;
      const response = await NhapKhoService.addNhapKhoCT(data);
      if (onSuccess) onSuccess(response);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);
```

```
export const updateNhapKhoCT = createAsyncThunk(
  "nhapkho/updateNhapKhoCT",
  async (data, { rejectWithValue }) => {
    try {
      const { onSuccess } = data;
      const payload = cloneDeep(data);
      delete payload.onSuccess;
      const response = await NhapKhoService.updateNhapKhoCT(data);
      if (onSuccess) onSuccess(response);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);
```

```
export const deleteNhapKhoCT = createAsyncThunk(
  "nhapkho/deleteNhapKhoCT",
  async (data, { rejectWithValue }) => {
    try {
      const { onSuccess, id } = data;
      const response = await NhapKhoService.deleteNhapKhoCT(id);
      if (onSuccess) onSuccess(response);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);
```

```
const initialState = {
  loading: false,
  nhapKhoCTList: [],
};
```

```
export const nhapkhoSlice = createSlice({
  name: "nhapkho",
  initialState,
  reducers: {
    showLoading: (state) => {
```

```

    state.loading = true;
  },
  setHisInfo: (state, action) => {
    state.hisInfoList = action.payload;
  },
  setNhapKhoCTList: (state, action) => {
    state.nhapKhoCTList = action.payload;
  },
},
extraReducers: (builder) => {
  builder
    .addCase(getNhapKhoCTByManx.pending, (state) => {
      state.loading = true;
    })
    .addCase(getNhapKhoCTByManx.fulfilled, (state, action) => {
      state.loading = false;
      state.nhapKhoCTList = [initNhapKhoCt, ...action.payload];
    })
    .addCase(getNhapKhoCTByManx.rejected, (state, action) => {
      state.loading = false;
    })
    .addCase(addNewNhapKhoCT.pending, (state) => {
      state.loading = true;
    })
    .addCase(addNewNhapKhoCT.fulfilled, (state, action) => {
      state.loading = false;
    })
    .addCase(addNewNhapKhoCT.rejected, (state, action) => {
      state.loading = false;
    })
    .addCase(updateNhapKhoCT.pending, (state) => {
      state.loading = true;
    })
    .addCase(updateNhapKhoCT.fulfilled, (state, action) => {
      state.loading = false;
    })
    .addCase(updateNhapKhoCT.rejected, (state, action) => {
      state.loading = false;
    })
    .addCase(deleteNhapKhoCT.pending, (state) => {
      state.loading = true;
    })
    .addCase(deleteNhapKhoCT.fulfilled, (state, action) => {
      state.loading = false;
    })
    .addCase(deleteNhapKhoCT.rejected, (state, action) => {
      state.loading = false;
    });
},
});

export const { showLoading, setHisInfo, setNhapKhoCTList } =
  nhapkhoSlice.actions;

export default nhapkhoSlice.reducer;

```