

```

import { createSlice, createAsyncThunk } from "@reduxjs/toolkit";
import { cloneDeep } from "lodash";
import BhxhCategoryService from "services/BhxhCategoryService";
import axiosInstance from "auth/FetchInterceptor";
import { BHYT_IMPORT_DSNV } from "constants/ApiConstant";

export const searchGridAPI = createAsyncThunk(
  "bhytCategory/searchGridAPI",
  async (data, { rejectWithValue }) => {
    try {
      const response = await BhxhCategoryService.searchGrid(data);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

export const applyImportDsnvAPI = createAsyncThunk(
  "bhytCategory/applyImportDsnvAPI",
  async (data, { rejectWithValue }) => {
    try {
      const { onSuccess } = data;
      const payload = cloneDeep(data);
      delete payload.onSuccess;
      const response = await BhxhCategoryService.applyImportDsnv(data);
      if (onSuccess) onSuccess(response);
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

export const importDsnvAPI = createAsyncThunk(
  "bhytCategory/importDsnv",
  async (data, { rejectWithValue }) => {
    try {
      const { formData, onUploadCompleted } = data;
      const response = await axiosInstance.post(BHYT_IMPORT_DSNV, formData, {
        baseURL: process.env.REACT_APP_HRM_ENDPOINT_URL,
        ContentType: "multipart/form-data",
      });
      if (onUploadCompleted) onUploadCompleted(response.data);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

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

export const bhytCategorySlice = createSlice({
  name: "bhytCategory",
  initialState,
  reducers: {
    showLoading: (state) => {
      state.loading = true;
    },
  },
  extraReducers: (builder) => {
    builder
      .addCase(searchGridAPI.pending, (state) => {
        state.loading = true;
      })
      .addCase(searchGridAPI.fulfilled, (state, action) => {
        state.loading = false;
        state.bhytDsnvList = action.payload;
      })
  }
});

```

```

        .addCase(searchGridAPI.rejected, (state, action) => {
            state.loading = false;
        })
        .addCase(importDsnvAPI.fulfilled, (state, action) => {
            state.loading = false;
        })
        .addCase(importDsnvAPI.rejected, (state, action) => {
            state.loading = false;
        })
        .addCase(importDsnvAPI.pending, (state) => {
            state.loading = true;
        })
        .addCase(applyImportDsnvAPI.fulfilled, (state, action) => {
            state.loading = false;
        })
        .addCase(applyImportDsnvAPI.rejected, (state, action) => {
            state.loading = false;
        })
        .addCase(applyImportDsnvAPI.pending, (state) => {
            state.loading = true;
        });
    },
});

```

```

export const { showLoading } = bhytCategorySlice.actions;

```

```

export default bhytCategorySlice.reducer;

```