

```

import { createSlice, createAsyncThunk } from "@reduxjs/toolkit";
import { cloneDeep } from "lodash";
import DanhMucService from "services/sale/DanhMucService";
import { initDmLoaiDv } from "views/app-views/sale/category/DmLoaiDichVu";
import { initDmKk } from "views/app-views/sale/category/DmKk";
import { initDmdonvitinh } from "views/app-views/sale/category/Dmdonvitinh";
import { initDmhoatchat } from "views/app-views/sale/category/Dmhoatchat";
import { initDmnhomthuoc } from "views/app-views/sale/category/Dmnhomthuoc";
import { initDmthuocvattu } from "views/app-views/sale/category/Dmthuocvattu";
import { initHangSapHetTon } from "views/app-views/sale/category/Dmhanghetton";

// #region DmKk
export const getDmLoaiKk = createAsyncThunk(
  "danhmuc/getDmLoaiKk",
  async (_, { rejectWithValue }) => {
    try {
      const response = await DanhMucService.getDmLoaiKk();
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);
export const getDmKk = createAsyncThunk(
  "danhmuc/getDmKk",
  async (_, { rejectWithValue }) => {
    try {
      const response = await DanhMucService.getDmKk();
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);
export const getAllDmKk = createAsyncThunk(
  "danhmuc/getAllDmKk",
  async (data, { rejectWithValue }) => {
    try {
      const response = await DanhMucService.getAllDmKk(data);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);
export const createDmKk = createAsyncThunk(
  "danhmuc/addnewDmKk",
  async (data, { rejectWithValue }) => {
    try {
      const { onSuccess } = data;
      const payload = cloneDeep(data);
      delete payload.onSuccess;
      const response = await DanhMucService.addDmKk(payload);
      if (onSuccess) onSuccess(response);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);
export const updateDmKk = createAsyncThunk(
  "danhmuc/updateDmKk",
  async (data, { rejectWithValue }) => {
    try {
      const { onSuccess } = data;
      const payload = cloneDeep(data);
      delete payload.onSuccess;
      const response = await DanhMucService.updateDmKk(payload);

```

```

        if (onSuccess) onSuccess(response);
        return response.data;
    } catch (err) {
        return rejectWithValue(err.message || "Error");
    }
}
);

export const deleteDmkk = createAsyncThunk(
    "danhmuc/deleteDmkk",
    async (data, { rejectWithValue }) => {
        try {
            const { onSuccess, id } = data;
            const payload = cloneDeep(data);
            delete payload.onSuccess;
            const response = await DanhMucService.deleteDmkk(id);
            if (onSuccess) onSuccess(response);
            return response.data;
        } catch (err) {
            return rejectWithValue(err.message || "Error");
        }
    }
);
//endregion

//region DmLoaiDv
export const getDmLoaiDv = createAsyncThunk(
    "danhmuc/getDmLoaiDv",
    async (_, { rejectWithValue }) => {
        try {
            const response = await DanhMucService.getDmLoaiDv();
            return response.data;
        } catch (err) {
            return rejectWithValue(err.message || "Error");
        }
    }
);

export const getAllDmLoaiDv = createAsyncThunk(
    "danhmuc/getAllDmLoaiDv",
    async (_, { rejectWithValue }) => {
        try {
            const response = await DanhMucService.getAllDmLoaiDv();
            return response.data;
        } catch (err) {
            return rejectWithValue(err.message || "Error");
        }
    }
);

export const createDmLoaiDv = createAsyncThunk(
    "danhmuc/addnewDmLoaiDv",
    async (data, { rejectWithValue }) => {
        try {
            const { onSuccess } = data;
            const payload = cloneDeep(data);
            delete payload.onSuccess;
            const response = await DanhMucService.addDmLoaiDv(payload);
            if (onSuccess) onSuccess(response);
            return response.data;
        } catch (err) {
            return rejectWithValue(err.message || "Error");
        }
    }
);

export const updateDmLoaiDv = createAsyncThunk(
    "danhmuc/updateDmLoaiDv",
    async (data, { rejectWithValue }) => {
        try {
            const { onSuccess } = data;
            const payload = cloneDeep(data);
            delete payload.onSuccess;
            const response = await DanhMucService.updateDmLoaiDv(payload);

```

```

        if (onSuccess) onSuccess(response);
        return response.data;
    } catch (err) {
        return rejectWithValue(err.message || "Error");
    }
}
);

export const deleteDmLoaiDv = createAsyncThunk(
    "danhmuc/deleteDmLoaiDv",
    async (data, { rejectWithValue }) => {
        try {
            const { onSuccess, id } = data;
            const payload = cloneDeep(data);
            delete payload.onSuccess;
            const response = await DanhMucService.deleteDmLoaiDv(id);
            if (onSuccess) onSuccess(response);
            return response.data;
        } catch (err) {
            return rejectWithValue(err.message || "Error");
        }
    }
);
//#endregion

//#region Dmdoituong
export const getDmDoiTuong = createAsyncThunk(
    "danhmuc/getDmDoiTuong",
    async (_, { rejectWithValue }) => {
        try {
            const response = await DanhMucService.getDmDoiTuong();
            return response.data;
        } catch (err) {
            return rejectWithValue(err.message || "Error");
        }
    }
);
//#endregion

//#region Dmthuocvattu
export const getDmThuocVatTu = createAsyncThunk(
    "danhmuc/getDmThuocVatTu",
    async (_, { rejectWithValue }) => {
        try {
            const response = await DanhMucService.getDmThuocVatTu();
            return response.data;
        } catch (err) {
            return rejectWithValue(err.message || "Error");
        }
    }
);

export const getAllDmthuocvattu = createAsyncThunk(
    "danhmuc/getAllDmthuocvattu",
    async (_, { rejectWithValue }) => {
        try {
            const response = await DanhMucService.getAllDmThuocVatTu();
            return response.data;
        } catch (err) {
            return rejectWithValue(err.message || "Error");
        }
    }
);

export const searchDmThuocVatTu = createAsyncThunk(
    "danhmuc/searchDmThuocVatTu",
    async (data, { rejectWithValue }) => {
        try {
            const response = await DanhMucService.searchDmThuocVatTu(data);
            return response.data;
        } catch (err) {
            return rejectWithValue(err.message || "Error");
        }
    }
);

```

```

);

export const getHangSapHetTon = createAsyncThunk(
  "danhmuc/getHangSapHetTon",
  async (_, { rejectWithValue }) => {
    try {
      const response = await DanhMucService.getHangSapHetTon();
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

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

export const createDmthuocvattu = createAsyncThunk(
  "danhmuc/addnewDmThuocVatTu",
  async (data, { rejectWithValue }) => {
    try {
      const { onSuccess } = data;
      const payload = cloneDeep(data);
      delete payload.onSuccess;
      const response = await DanhMucService.addDmThuocVatTu(payload);
      if (onSuccess) onSuccess(response);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

export const updateDmthuocvattu = createAsyncThunk(
  "danhmuc/updateDmThuocVatTu",
  async (data, { rejectWithValue }) => {
    try {
      const { onSuccess } = data;
      const payload = cloneDeep(data);
      delete payload.onSuccess;
      const response = await DanhMucService.updateDmThuocVatTu(payload);
      if (onSuccess) onSuccess(response);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

export const deleteDmthuocvattu = createAsyncThunk(
  "danhmuc/deleteDmThuocVatTu",
  async (data, { rejectWithValue }) => {
    try {
      const { onSuccess, id } = data;
      const payload = cloneDeep(data);
      delete payload.onSuccess;
      const response = await DanhMucService.deleteDmThuocVatTu(id);
      if (onSuccess) onSuccess(response);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

```

```

    }
  }
);

//#endregion

//#region Dmphai
export const getDmPhai = createAsyncThunk(
  "danhmuc/getDmPhai",
  async (_, { rejectWithValue }) => {
    try {
      const response = await DanhMucService.getDmPhai();
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);
//#endregion

//#region DmloaiPhieu
export const getDmLoaiPhieuNhapXuat = createAsyncThunk(
  "danhmuc/getDmLoaiPhieuNhapXuat",
  async (_, { rejectWithValue }) => {
    try {
      const response = await DanhMucService.getDmLoaiPhieuNhapXuat();
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);
//#endregion

//#region Dmdonvitinh
export const getDmdonvitinh = createAsyncThunk(
  "danhmuc/getDmdonvitinh",
  async (_, { rejectWithValue }) => {
    try {
      const response = await DanhMucService.getDmdonvitinh();
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

export const getAllDmdonvitinh = createAsyncThunk(
  "danhmuc/getAllDmdonvitinh",
  async (data, { rejectWithValue }) => {
    try {
      const response = await DanhMucService.getAllDmdonvitinh(data);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

export const createDmdonvitinh = createAsyncThunk(
  "danhmuc/addnewDmdonvitinh",
  async (data, { rejectWithValue }) => {
    try {
      const { onSuccess } = data;
      const payload = cloneDeep(data);
      delete payload.onSuccess;
      const response = await DanhMucService.addDmdonvitinh(payload);
      if (onSuccess) onSuccess(response);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

```

```

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

export const deleteDmdonvitinh = createAsyncThunk(
  "danhmuc/deleteDmkk",
  async (data, { rejectWithValue }) => {
    try {
      const { onSuccess, id } = data;
      const payload = cloneDeep(data);
      delete payload.onSuccess;
      const response = await DanhMucService.deleteDmdonvitinh(id);
      if (onSuccess) onSuccess(response);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);
//endregion

//region Dmhoatchat
export const getDmhoatchat = createAsyncThunk(
  "danhmuc/getDmhoatchat",
  async (_, { rejectWithValue }) => {
    try {
      const response = await DanhMucService.getDmhoatchat();
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

export const getAllDmhoatchat = createAsyncThunk(
  "danhmuc/getAllDmdhoatchat",
  async (data, { rejectWithValue }) => {
    try {
      const response = await DanhMucService.getAllDmhoatchat(data);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

export const createDmhoatchat = createAsyncThunk(
  "danhmuc/addnewDmhoatchat",
  async (data, { rejectWithValue }) => {
    try {
      const { onSuccess } = data;
      const payload = cloneDeep(data);
      delete payload.onSuccess;
      const response = await DanhMucService.addDmhoatchat(payload);
      if (onSuccess) onSuccess(response);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

```

```

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

export const deleteDmhoatchat = createAsyncThunk(
  "danhmuc/deleteDmkk",
  async (data, { rejectWithValue }) => {
    try {
      const { onSuccess, id } = data;
      const payload = cloneDeep(data);
      delete payload.onSuccess;
      const response = await DanhMucService.deleteDmhoatchat(id);
      if (onSuccess) onSuccess(response);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);
//endregion

//region Dmnhomthuoc
export const getDmdmnhomthuoc = createAsyncThunk(
  "danhmuc/getDmnhomthuoc",
  async (_, { rejectWithValue }) => {
    try {
      const response = await DanhMucService.getDmnhomthuoc();
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

export const getAllDmnhomthuoc = createAsyncThunk(
  "danhmuc/getAllDmnhomthuoc",
  async (_, { rejectWithValue }) => {
    try {
      const response = await DanhMucService.getAllDmnhomthuoc();
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

export const createDmnhomthuoc = createAsyncThunk(
  "danhmuc/addnewDmnhomthuoc",
  async (data, { rejectWithValue }) => {
    try {
      const { onSuccess } = data;
      const payload = cloneDeep(data);
      delete payload.onSuccess;
      const response = await DanhMucService.addDmnhomthuoc(payload);
      if (onSuccess) onSuccess(response);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

```

```

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

```

```

export const deleteDmnhomthuoc = createAsyncThunk(
  "danhmuc/deleteDmnhomthuoc",
  async (data, { rejectWithValue }) => {
    try {
      const { onSuccess, id } = data;
      const payload = cloneDeep(data);
      delete payload.onSuccess;
      const response = await DanhMucService.deleteDmnhomthuoc(id);
      if (onSuccess) onSuccess(response);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);
//#endregion

```

```

export const getDmLyDoTc = createAsyncThunk(
  "danhmuc/getDmLyDoTc",
  async (data, { rejectWithValue }) => {
    try {
      const response = await DanhMucService.getDmLyDoTc(data);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

```

```

export const getDmHinhThucTT = createAsyncThunk(
  "danhmuc/getDmHinhThucTT",
  async (_, { rejectWithValue }) => {
    try {
      const response = await DanhMucService.getDmHinhThucTT();
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

```

```

export const getDmNhanVien = createAsyncThunk(
  "danhmuc/getDmNhanVien",
  async (data, { rejectWithValue }) => {
    try {
      const response = await DanhMucService.getDmNhanVien(data);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

```

```

export const getDmKhachHang = createAsyncThunk(
  "danhmuc/getDmKhachHang",
  async (data, { rejectWithValue }) => {
    try {
      const response = await DanhMucService.getDmKhachHang(data);
      return response.data;
    } catch (err) {

```



```

    return rejectWithValue(err.message || "Error");
  }
}
);

const initialState = {
  loading: false,
  dmLoaiKkList: [],
  dmKkList: [],
  dmLoaiPhieuNhapXuatList: [],
  dmDoiTuongList: [],
  dmThuocVatTuList: [],
  searchDmThuocVatTuList: [],
  allDmKkList: [],
  allDmLoaiDvList: [],
  allDmdonvitinhList: [],
  allDmhoatchatList: [],
  allDmnhomthuocList: [],
  allDmthuocvattuList: [],
  allHangSapHetTonList: [],
  dmPhaiList: [],
  dmHinhThucTTLList: [],
  dmLyDoTcList: [],
  dmNhanVienList: [],
  dmKhachHangList: [],
  dmkkId: null,
};

export const danhmucSlice = createSlice({
  name: "danhmuc",
  initialState,
  reducers: {
    showLoading: (state) => {
      state.loading = true;
    },
    setAllDmkkList: (state, action) => {
      state.allDmKkList = action.payload;
    },
    setAllDmLoaiDvList: (state, action) => {
      state.allDmLoaiDvList = action.payload;
    },
    setAllDmdonvitinhList: (state, action) => {
      state.allDmdonvitinhList = action.payload;
    },
    setAllDmhoatchatList: (state, action) => {
      state.allDmhoatchatList = action.payload;
    },
    setAllDmnhomthuocList: (state, action) => {
      state.allDmnhomthuocList = action.payload;
    },
    setAllDmthuocvattuList: (state, action) => {
      state.allDmthuocvattuList = action.payload;
    },
    setAllHangSapHetTonList: (state, action) => {
      state.allHangSapHetTonList = action.payload;
    },
  },
  extraReducers: (builder) => {
    builder
      .addCase(getDmKk.pending, (state) => {
        state.loading = true;
      })
      .addCase(getDmKk.fulfilled, (state, action) => {
        state.loading = false;
        state.dmKkList = action.payload;
        state.dmkkId =
          action.payload.filter((x) => x.maloai === 1).length > 0
            ? action.payload.filter((x) => x.maloai === 1)[0].value
            : null;
      })
      .addCase(getDmKk.rejected, (state, action) => {
        state.loading = false;
      }) //
      .addCase(searchDmThuocVatTu.pending, (state) => {

```

```

    state.loading = true;
  })
  .addCase(searchDmThuocVatTu.fulfilled, (state, action) => {
    state.loading = false;
    state.searchDmThuocVatTuList = [initDmkk, ...action.payload];
  })
  .addCase(searchDmThuocVatTu.rejected, (state, action) => {
    state.loading = false;
  }) //
  .addCase(getDmLoaiKk.pending, (state) => {
    state.loading = true;
  })
  .addCase(getDmLoaiKk.fulfilled, (state, action) => {
    state.loading = false;
    state.dmLoaiKkList = action.payload;
  })
  .addCase(getDmLoaiKk.rejected, (state, action) => {
    state.loading = false;
  }) //
  .addCase(getAllDmKk.pending, (state) => {
    state.loading = true;
  })
  .addCase(getAllDmKk.fulfilled, (state, action) => {
    state.loading = false;
    state.allDmKkList = [initDmkk, ...action.payload];
  })
  .addCase(getAllDmKk.rejected, (state, action) => {
    state.loading = false;
  }) //
  .addCase(getAllDmLoaiDv.pending, (state) => {
    state.loading = true;
  })
  .addCase(getAllDmLoaiDv.fulfilled, (state, action) => {
    state.loading = false;
    state.allDmLoaiDvList = [initDmLoaiDv, ...action.payload];
  })
  .addCase(getAllDmLoaiDv.rejected, (state, action) => {
    state.loading = false;
  })
  .addCase(getAllDmdonvitinh.pending, (state) => {
    state.loading = true;
  })
  .addCase(getAllDmdonvitinh.fulfilled, (state, action) => {
    state.loading = false;
    state.allDmdonvitinhList = [initDmdonvitinh, ...action.payload];
  })
  .addCase(getAllDmdonvitinh.rejected, (state, action) => {
    state.loading = false;
  })
  .addCase(getAllDmnhomthuoc.pending, (state) => {
    state.loading = true;
  })
  .addCase(getAllDmnhomthuoc.fulfilled, (state, action) => {
    state.loading = false;
    state.allDmnhomthuocList = [initDmnhomthuoc, ...action.payload];
  })
  .addCase(getAllDmnhomthuoc.rejected, (state, action) => {
    state.loading = false;
  })
  .addCase(getAllDmhoatchat.pending, (state) => {
    state.loading = true;
  })
  .addCase(getAllDmhoatchat.fulfilled, (state, action) => {
    state.loading = false;
    state.allDmhoatchatList = [initDmhoatchat, ...action.payload];
  })
  .addCase(getAllDmhoatchat.rejected, (state, action) => {
    state.loading = false;
  })
  .addCase(getDmLoaiPhieuNhapXuat.pending, (state) => {
    state.loading = true;
  })
  .addCase(getDmLoaiPhieuNhapXuat.fulfilled, (state, action) => {
    state.loading = false;
  })

```

```

    state.dmLoaiPhieuNhapXuatList = action.payload;
  })
  .addCase(getDmLoaiPhieuNhapXuat.rejected, (state, action) => {
    state.loading = false;
  })
  .addCase(getDmDoiTuong.pending, (state) => {
    state.loading = true;
  })
  .addCase(getDmDoiTuong.fulfilled, (state, action) => {
    state.loading = false;
    state.dmDoiTuongList = action.payload;
  })
  .addCase(getDmDoiTuong.rejected, (state, action) => {
    state.loading = false;
  })
  .addCase(getDmThuocVatTu.pending, (state) => {
    state.loading = true;
  })
  .addCase(getDmThuocVatTu.fulfilled, (state, action) => {
    state.loading = false;
    state.dmThuocVatTuList = action.payload;
  })
  .addCase(getDmThuocVatTu.rejected, (state, action) => {
    state.loading = false;
  })
  .addCase(getDmPhai.pending, (state) => {
    state.loading = true;
  })
  .addCase(getDmPhai.fulfilled, (state, action) => {
    state.loading = false;
    state.dmPhaiList = action.payload;
  })
  .addCase(getDmPhai.rejected, (state, action) => {
    state.loading = false;
  })
  .addCase(getAllDmthuocvattu.pending, (state) => {
    state.loading = true;
  })
  .addCase(getAllDmthuocvattu.fulfilled, (state, action) => {
    state.loading = false;
    state.allDmthuocvattuList = [initDmthuocvattu, ...action.payload];
  })
  .addCase(getAllDmthuocvattu.rejected, (state, action) => {
    state.loading = false;
  })
  .addCase(getHangSapHetTon.pending, (state) => {
    state.loading = true;
  })
  .addCase(getHangSapHetTon.fulfilled, (state, action) => {
    state.loading = false;
    state.allHangSapHetTonList = [initHangSapHetTon, ...action.payload];
  })
  .addCase(getHangSapHetTon.rejected, (state, action) => {
    state.loading = false;
  })
  .addCase(getDmLyDoTc.fulfilled, (state, action) => {
    state.loading = false;
    state.dmLyDoTcList = action.payload;
  })
  .addCase(getDmHinhThucTT.fulfilled, (state, action) => {
    state.loading = false;
    state.dmHinhThucTTList = action.payload;
  })
  .addCase(getDmNhanVien.fulfilled, (state, action) => {
    state.loading = false;
    state.dmNhanVienList = action.payload;
  })
  .addCase(getDmKhachHang.fulfilled, (state, action) => {
    state.loading = false;
    state.dmKhachHangList = action.payload;
  });
}

```

```

},

```

```

});

```

```
export const {
  showLoading,
  setAllDmkkList,
  setAllDmLoaiDvList,
  setAllDmdonvitinhList,
  setAllDmhoatchatList,
  setAllDmnhomthuocList,
  setAllDmthuocvattuList,
} = danhmucSlice.actions;

export default danhmucSlice.reducer;
```