```
import { createSlice, createAsyncThunk } from "@reduxjs/toolkit";
import KhachHangService from "services/KhachHangService";
import { cloneDeep } from "lodash";

///-> Bệnh nhân
export const getDSKhachHang = createAsyncThunk(
  "khachhang/getDSKhachHang",
  async (data, { rejectWithValue }) => {
    try {
      const response = await KhachHangService.getDSKhachHang(data);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);
export const getAllKhaiBaoNhom = createAsyncThunk(
  "khachhang/getAllKhaiBaoNhom",
  async (data, { rejectWithValue }) => {
    try {
      const response = await KhachHangService.getAllKhaiBaoNhom(data);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);
export const getDSNhom = createAsyncThunk(
  "khachhang/getDSNhom",
  async (data, { rejectWithValue }) => {
    try {
      const response = await KhachHangService.getDSNhom();
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);
export const getAllBenhICD10 = createAsyncThunk(
  "khachhang/getAllBenhICD10",
  async (data, { rejectWithValue }) => {
    try {
      const response = await KhachHangService.getAllBenhICD10();
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);
export const addNewKhaiBaoNhom = createAsyncThunk(
  "khachhang/addNewKhaiBaoNhom",
  async (data, { rejectWithValue }) => {
    try {
      const { onSuccess } = data;
      const payload = cloneDeep(data);
      delete payload.onSuccess;
      const response = await KhachHangService.addNewKhaiBaoNhom(payload);
      if (onSuccess) onSuccess(response);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);
export const editKhaiBaoNhom = createAsyncThunk(
  "khachhang/editKhaiBaoNhom",
  async (data, { rejectWithValue }) => {
    try {
      const { onSuccess } = data;
      const payload = cloneDeep(data);
      delete payload.onSuccess;
      const response = await KhachHangService.editKhaiBaoNhom(payload);
      if (onSuccess) onSuccess(response);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

export const dongBoKhachHangVaoNhom = createAsyncThunk(
  "khachhang/dongBoKhachHangVaoNhom",
  async (data, { rejectWithValue }) => {
    try {
      const { onSuccess } = data;
      const payload = cloneDeep(data);
      delete payload.onSuccess;
      const response = await KhachHangService.dongBoKhachHangVaoNhom(payload);
      if (onSuccess) onSuccess(response);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);
export const addBenhNhanNhom = createAsyncThunk(
  "khachhang/addBenhNhanNhom",
  async (data, { rejectWithValue }) => {
    try {
      const { onSuccess } = data;
      const payload = cloneDeep(data);
      delete payload.onSuccess;
      const response = await KhachHangService.addBenhNhanNhom(payload);
      if (onSuccess) onSuccess(response);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);
export const getLichSuKcbByIdBenhNhan = createAsyncThunk(
  "khachhang/getLichSuKcbByIdBenhNhan",
  async (id, { rejectWithValue }) => {
    try {
      const response = await KhachHangService.getLichSuKcbByIdBenhNhan(id);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);
export const getLichHenKcbByIdBenhNhan = createAsyncThunk(
  "khachhang/getLichHenKcbByIdBenhNhan",
  async (id, { rejectWithValue }) => {
    try {
      const response = await KhachHangService.getLichHenKcbByIdBenhNhan(id);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

///-> Khách hàng NCC
export const searchKhachHangNCC = createAsyncThunk(
  "khachhang/searchKhachHangNCC",
  async (data, { rejectWithValue }) => {
    try {
      const response = await KhachHangService.searchKhachHangNCC(data);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);
```

```
export const getAllKhachHangNCC = createAsyncThunk(
  "khachhang/getAllKhachHangNCC",
  async (data, { rejectWithValue }) => {
    try {
      const response = await KhachHangService.getAllKhachHangNCC();
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);
export const getKhachHangNCC = createAsyncThunk(
  "khachhang/getKhachHangNCC",
  async (data, { rejectWithValue }) => {
    try {
      const response = await KhachHangService.getKhachHangNCC(data);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);
export const upsertKhachHangNCC = createAsyncThunk(
  "khachhang/upsertKhachHangNCC",
  async (data, { rejectWithValue }) => {
    try {
      const { onSuccess } = data;
      const payload = cloneDeep(data);
      delete payload.onSuccess;
      const response = await KhachHangService.upsertKhachHangNCC(payload);
      if (onSuccess) onSuccess(response);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);
export const deleteKhachHangNCC = createAsyncThunk(
  "khachhang/deleteKhachHangNCC",
  async (data, { rejectWithValue }) => {
    try {
      const { onSuccess } = data;
      const payload = cloneDeep(data);
      delete payload.onSuccess;
      const response = await KhachHangService.deleteKhachHangNCC(payload);
      if (onSuccess) onSuccess(response);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

const initialState = {
  loading: false,
  khachHangList: [],
  khaiBaoNhomList: [],
  khaiBaoNhomAllList: [],
  allBenhICD10List: [],
  lichSuKCBKhachHangList: [],
  lichHenKCBKhachHangList: [],
  searchKhachHangNCCList: [],
};

export const khachhangSlice = createSlice({
  name: "khachhang",
  initialState,
  reducers: {
    showLoading: (state) => {
      state.loading = true;
    },
  },
  extraReducers: (builder) => {
    builder
      //searchKhachHangNCC
      .addCase(searchKhachHangNCC.pending, (state) => {
        state.loading = true;
      })
      .addCase(searchKhachHangNCC.fulfilled, (state, action) => {
        state.loading = false;
        state.searchKhachHangNCCList = action.payload;
      })
      .addCase(searchKhachHangNCC.rejected, (state, action) => {
        state.loading = false;
      })
      //
      .addCase(getDSKhachHang.pending, (state) => {
        state.loading = true;
      })
      .addCase(getDSKhachHang.fulfilled, (state, action) => {
        state.loading = false;
        state.khachHangList = action.payload;
      })
      .addCase(getDSKhachHang.rejected, (state, action) => {
        state.loading = false;
      })
      .addCase(getDSNhom.pending, (state) => {
        state.loading = true;
      })
      .addCase(getDSNhom.fulfilled, (state, action) => {
        state.loading = false;
        state.khaiBaoNhomList = action.payload;
      })
      .addCase(getDSNhom.rejected, (state, action) => {
        state.loading = false;
      })
      .addCase(getAllKhaiBaoNhom.pending, (state) => {
        state.loading = true;
      })
      .addCase(getAllKhaiBaoNhom.fulfilled, (state, action) => {
        state.loading = false;
        state.khaiBaoNhomAllList = action.payload;
      })
      .addCase(getAllKhaiBaoNhom.rejected, (state, action) => {
        state.loading = false;
      })
      .addCase(addNewKhaiBaoNhom.pending, (state) => {
        state.loading = true;
      })
      .addCase(addNewKhaiBaoNhom.fulfilled, (state, action) => {
        state.loading = false;
      })
      .addCase(addNewKhaiBaoNhom.rejected, (state, action) => {
        state.loading = false;
      })
      .addCase(editKhaiBaoNhom.pending, (state) => {
        state.loading = true;
      })
      .addCase(editKhaiBaoNhom.fulfilled, (state, action) => {
        state.loading = false;
      })
      .addCase(editKhaiBaoNhom.rejected, (state, action) => {
        state.loading = false;
      })
      .addCase(dongBoKhachHangVaoNhom.pending, (state) => {
        state.loading = true;
      })
      .addCase(dongBoKhachHangVaoNhom.fulfilled, (state, action) => {
        state.loading = false;
      })
      .addCase(dongBoKhachHangVaoNhom.rejected, (state, action) => {
        state.loading = false;
      })
      .addCase(getAllBenhICD10.pending, (state) => {
        state.loading = true;
      })
      .addCase(getAllBenhICD10.fulfilled, (state, action) => {
        state.loading = false;
```

```
        state.allBenhICD10List = action.payload;
      })
      .addCase(getAllBenhICD10.rejected, (state, action) => {
        state.loading = false;
      })
      .addCase(addBenhNhanNhom.pending, (state) => {
        state.loading = true;
      })
      .addCase(addBenhNhanNhom.fulfilled, (state, action) => {
        state.loading = false;
      })
      .addCase(addBenhNhanNhom.rejected, (state, action) => {
        state.loading = false;
      })
      .addCase(getLichSuKcbByIdBenhNhan.pending, (state) => {
        state.loading = true;
      })
      .addCase(getLichSuKcbByIdBenhNhan.fulfilled, (state, action) => {
        state.loading = false;
        state.lichSuKCBKhachHangList = action.payload;
      })
      .addCase(getLichSuKcbByIdBenhNhan.rejected, (state, action) => {
        state.loading = false;
      })
      .addCase(getLichHenKcbByIdBenhNhan.pending, (state) => {
        state.loading = true;
      })
      .addCase(getLichHenKcbByIdBenhNhan.fulfilled, (state, action) => {
        state.loading = false;
        state.lichHenKCBKhachHangList = action.payload;
      })
      .addCase(getLichHenKcbByIdBenhNhan.rejected, (state, action) => {
        state.loading = false;
      });
  },
});

export const { showLoading } = khachhangSlice.actions;

export default khachhangSlice.reducer;
```