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

///THIET LAP
export const GetAllCongTy = createAsyncThunk(
  "KhamSucKhoeDoan/GetAllCongTy",
  async (data, { rejectWithValue }) => {
    try {
      const response = await KhamSucKhoeDoanService.GetAllCongTy(data);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);
export const GetAllCongTyDto = createAsyncThunk(
  "KhamSucKhoeDoan/GetAllCongTyDto",
  async (data, { rejectWithValue }) => {
    try {
      const response = await KhamSucKhoeDoanService.GetAllCongTyDto();
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

export const GetCongTy = createAsyncThunk(
  "KhamSucKhoeDoan/GetCongTy",
  async (id, { rejectWithValue }) => {
    try {
      const response = await KhamSucKhoeDoanService.GetCongTy(id);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

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

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

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

export const GetDmDichVu = createAsyncThunk(
  "KhamSucKhoeDoan/GetDmDichVu",
  async (id, { rejectWithValue }) => {
    try {
      const response = await KhamSucKhoeDoanService.GetDmDichVu(id);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

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

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

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

export const GetPhanTramKinhDoanh = createAsyncThunk(
  "KhamSucKhoeDoan/GetPhanTramKinhDoanh",
  async (id, { rejectWithValue }) => {
```

```
    try {
      const response = await KhamSucKhoeDoanService.GetPhanTramKinhDoanh(id);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

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

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

export const GetAllHangMucKham = createAsyncThunk(
  "KhamSucKhoeDoan/GetAllHangMucKham",
  async (data, { rejectWithValue }) => {
    try {
      const response = await KhamSucKhoeDoanService.GetAllThietLapKskDoan(data);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);
export const GetAllMauKham = createAsyncThunk(
  "KhamSucKhoeDoan/GetAllMauKham",
  async (data, { rejectWithValue }) => {
    try {
      const response = await KhamSucKhoeDoanService.GetAllThietLapKskDoan(data);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

export const UpsertThietLapKskDoan = createAsyncThunk(
  "KhamSucKhoeDoan/UpsertThietLapKskDoan",
  async (data, { rejectWithValue }) => {
    try {
      const { onSuccess } = data;
```

```
      const payload = cloneDeep(data);
      delete payload.onSuccess;
      const response = await KhamSucKhoeDoanService.UpsertThietLapKskDoan(
        payload
      );
      if (onSuccess) onSuccess(response);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

export const DeleteThietLapKskDoan = createAsyncThunk(
  "KhamSucKhoeDoan/DeleteThietLapKskDoan",
  async (data, { rejectWithValue }) => {
    try {
      const { onSuccess } = data;
      const payload = cloneDeep(data);
      delete payload.onSuccess;
      const response = await KhamSucKhoeDoanService.DeleteThietLapKskDoan(
        payload.id
      );
      if (onSuccess) onSuccess(response);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);
////ĐOÀN KHÁM SỨC KHỎE
export const GetAllDoanKhamSucKhoe = createAsyncThunk(
  "KhamSucKhoeDoan/GetAllDoanKhamSucKhoe",
  async (data, { rejectWithValue }) => {
    try {
      const response = await KhamSucKhoeDoanService.GetAllDoanKhamSucKhoe(data);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);
export const GetDoanKhamSucKhoe = createAsyncThunk(
  "KhamSucKhoeDoan/GetDoanKhamSucKhoe",
  async (data, { rejectWithValue }) => {
    try {
      const response = await KhamSucKhoeDoanService.GetDoanKhamSucKhoe(data);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

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

export const DeleteDoanKhamSucKhoe = createAsyncThunk(
  "KhamSucKhoeDoan/DeleteDoanKhamSucKhoe",
```

```
  async (data, { rejectWithValue }) => {
    try {
      const { onSuccess } = data;
      const payload = cloneDeep(data);
      delete payload.onSuccess;
      const response = await KhamSucKhoeDoanService.DeleteDoanKhamSucKhoe(
        payload
      );
      if (onSuccess) onSuccess(response);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);
//Phuong an kinh doanh

export const GetAllPhuongAnKinhDoanhKsk = createAsyncThunk(
  "KhamSucKhoeDoan/GetAllPhuongAnKinhDoanhKsk",
  async (data, { rejectWithValue }) => {
    try {
      const response = await KhamSucKhoeDoanService.GetAllPhuongAnKinhDoanhKsk(
        data
      );
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);
export const GetPhuongAnKinhDoanh = createAsyncThunk(
  "KhamSucKhoeDoan/GetPhuongAnKinhDoanh",
  async (data, { rejectWithValue }) => {
    try {
      const response = await KhamSucKhoeDoanService.GetPhuongAnKinhDoanh(data);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);
export const GetPhuongAnKinhDoanhChiTiet = createAsyncThunk(
  "KhamSucKhoeDoan/GetPhuongAnKinhDoanhChiTiet",
  async (data, { rejectWithValue }) => {
    try {
      const response = await KhamSucKhoeDoanService.GetPhuongAnKinhDoanhChiTiet(
        data
      );
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);
export const UpsertPhuongAnKinhDoanh = createAsyncThunk(
  "KhamSucKhoeDoan/UpsertPhuongAnKinhDoanh",
  async (data, { rejectWithValue }) => {
    try {
      const { onSuccess } = data;
      const payload = cloneDeep(data);
      delete payload.onSuccess;
      const response = await KhamSucKhoeDoanService.UpsertPhuongAnKinhDoanh(
        payload
      );
      if (onSuccess) onSuccess(response);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);
export const UpsertPhuongAnKinhDoanhChiTiet = createAsyncThunk(
  "KhamSucKhoeDoan/UpsertPhuongAnKinhDoanhChiTiet",
  async (data, { rejectWithValue }) => {
```

```
    try {
      const { onSuccess } = data;
      const payload = cloneDeep(data);
      delete payload.onSuccess;
      const response =
        await KhamSucKhoeDoanService.UpsertPhuongAnKinhDoanhChiTiet(payload);
      if (onSuccess) onSuccess(response);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);
export const UpdatePhuongAnKinhDoanh = createAsyncThunk(
  "KhamSucKhoeDoan/UpdatePhuongAnKinhDoanh",
  async (data, { rejectWithValue }) => {
    try {
      const { onSuccess } = data;
      const payload = cloneDeep(data);
      delete payload.onSuccess;
      const response = await KhamSucKhoeDoanService.UpdatePhuongAnKinhDoanh(
        payload
      );
      if (onSuccess) onSuccess(response);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);
export const DeletePhuongAnKinhDoanh = createAsyncThunk(
  "KhamSucKhoeDoan/DeletePhuongAnKinhDoanh",
  async (data, { rejectWithValue }) => {
    try {
      const { onSuccess } = data;
      const payload = cloneDeep(data);
      delete payload.onSuccess;
      const response = await KhamSucKhoeDoanService.DeletePhuongAnKinhDoanh(
        payload
      );
      if (onSuccess) onSuccess(response);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);
export const DeletePhuongAnKinhDoanhChiTiet = createAsyncThunk(
  "KhamSucKhoeDoan/DeletePhuongAnKinhDoanhChiTiet",
  async (data, { rejectWithValue }) => {
    try {
      const { onSuccess } = data;
      const payload = cloneDeep(data);
      delete payload.onSuccess;
      const response =
        await KhamSucKhoeDoanService.DeletePhuongAnKinhDoanhChiTiet(payload);
      if (onSuccess) onSuccess(response);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);
//Đơn hàng
export const GetAllDonHangKhamSucKhoe = createAsyncThunk(
  "KhamSucKhoeDoan/GetAllDonHangKhamSucKhoe",
  async (data, { rejectWithValue }) => {
    try {
      const response = await KhamSucKhoeDoanService.GetAllDonHangKhamSucKhoe(
        data
      );
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
```

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

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

export const GetAllDonHangKhamSucKhoeNhanSu = createAsyncThunk(
  "KhamSucKhoeDoan/GetAllDonHangKhamSucKhoeKhac",
  async (data, { rejectWithValue }) => {
    try {
      const response =
        await KhamSucKhoeDoanService.GetAllDonHangKhamSucKhoeKhac(data);
      return response.data;
```

```
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);
export const GetAllDonHangKhamSucKhoeVatTu = createAsyncThunk(
  "KhamSucKhoeDoan/GetAllDonHangKhamSucKhoeVatTu",
  async (data, { rejectWithValue }) => {
    try {
      const response =
        await KhamSucKhoeDoanService.GetAllDonHangKhamSucKhoeKhac(data);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);
export const UpsertDonHangKhamSucKhoeKhac = createAsyncThunk(
  "KhamSucKhoeDoan/UpsertDonHangKhamSucKhoe",
  async (data, { rejectWithValue }) => {
    try {
      const { onSuccess } = data;
      const payload = cloneDeep(data);
      delete payload.onSuccess;
      const response =
        await KhamSucKhoeDoanService.UpsertDonHangKhamSucKhoeKhac(payload);
      if (onSuccess) onSuccess(response);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);
export const DeleteDonHangKhamSucKhoeKhac = createAsyncThunk(
  "KhamSucKhoeDoan/DeleteDonHangKhamSucKhoe",
  async (data, { rejectWithValue }) => {
    try {
      const { onSuccess } = data;
      const payload = cloneDeep(data);
      delete payload.onSuccess;
      const response =
        await KhamSucKhoeDoanService.DeleteDonHangKhamSucKhoeKhac(payload);
      if (onSuccess) onSuccess(response);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);
///THANH LÝ HỢP ĐỒNG

export const GetAllThanhLyHopDongKsk = createAsyncThunk(
  "KhamSucKhoeDoan/GetAllThanhLyHopDongKsk",
  async (data, { rejectWithValue }) => {
    try {
      const response = await KhamSucKhoeDoanService.GetAllThanhLyHopDongKsk(
        data
      );
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);
export const GetThanhLyHopDongKsk = createAsyncThunk(
  "KhamSucKhoeDoan/GetThanhLyHopDongKsk",
  async (data, { rejectWithValue }) => {
    try {
      const response = await KhamSucKhoeDoanService.GetThanhLyHopDongKsk(data);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
```

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

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

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

```
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);
export const UpsertThanhLyHopDongKskChiTietKhac = createAsyncThunk(
  "KhamSucKhoeDoan/UpsertThanhLyHopDongKskChiTietKhac",
  async (data, { rejectWithValue }) => {
    try {
      const { onSuccess } = data;
      const payload = cloneDeep(data);
      delete payload.onSuccess;
      const response =
        await KhamSucKhoeDoanService.UpsertThanhLyHopDongKskChiTietKhac(
          payload
        );
      if (onSuccess) onSuccess(response);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);
//Thu tiền
export const GetAllThuTien = createAsyncThunk(
  "KhamSucKhoeDoan/GetAllThuTien",
  async (data, { rejectWithValue }) => {
    try {
      const response = await KhamSucKhoeDoanService.GetAllThuTien(data);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);
export const GetThuTien = createAsyncThunk(
  "KhamSucKhoeDoan/GetThuTien",
  async (data, { rejectWithValue }) => {
    try {
      const response = await KhamSucKhoeDoanService.GetThuTien(data);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);
export const UpsertThuTien = createAsyncThunk(
  "KhamSucKhoeDoan/UpsertThuTien",
  async (data, { rejectWithValue }) => {
    try {
      const { onSuccess } = data;
      const payload = cloneDeep(data);
      delete payload.onSuccess;
      const response = await KhamSucKhoeDoanService.UpsertThuTien(payload);
      if (onSuccess) onSuccess(response);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);
export const DeleteThuTien = createAsyncThunk(
  "KhamSucKhoeDoan/DeleteThuTien",
  async (data, { rejectWithValue }) => {
    try {
      const { onSuccess } = data;
      const payload = cloneDeep(data);
      delete payload.onSuccess;
      const response = await KhamSucKhoeDoanService.DeleteThuTien(payload);
      if (onSuccess) onSuccess(response);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
```

```
    }
  }
);
//Báo cáo công nợ
export const GetBaoCaoCongNo = createAsyncThunk(
  "KhamSucKhoeDoan/GetBaoCaoCongNo",
  async (data, { rejectWithValue }) => {
    try {
      const response = await KhamSucKhoeDoanService.GetBaoCaoCongNo(data);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);
export const GetBaoCaoDoanhThu = createAsyncThunk(
  "KhamSucKhoeDoan/GetBaoCaoDoanhThu",
  async (data, { rejectWithValue }) => {
    try {
      const response = await KhamSucKhoeDoanService.GetBaoCaoDoanhThu(data);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);
export const ExportBaoCao = createAsyncThunk(
  "KhamSucKhoeDoan/ExportBaoCao",
  async (data, { rejectWithValue }) => {
    try {
      const { onSuccess } = data;
      const payload = cloneDeep(data);
      delete payload.onSuccess;
      const response = await KhamSucKhoeDoanService.ExportBaoCao(payload);
      if (onSuccess) onSuccess(response);
      return response.data;

      // const response = await KhamSucKhoeDoanService.ExportBaoCao(data);
      // return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

const initialState = {
  loading: false,
  CongTyList: [],
  CongTyDtoList: [],
  DmDichVuList: [],
  DoanKhamList: [],
  DoanKhamSucKhoeDetail: {},
  HangMucKhamList: [],
  MauKhamList: [],
  PhuongAnKinhDoanhList: [],
  PhuongAnKinhDoanhDetail: {},
  PhuongAnKinhDoanhChiTietList: [],
  DonHangKhamSucKhoeDetail: {},
  DonHangKhamSucKhoeList: [],
  ThanhLyHopDongKskDetail: {},
  ThanhLyHopDongKskList: [],
  ThanhLyHopDongKskChiTietList: [],
  ThuTienDetail: {},
  ThuTienList: [],
  BaoCaoCongNoList: [],
  BaoCaoDoanhThuList: [],
  ThanhLyHopDongChiTietKhacList: [],
  DonHangKhamSucKhoeNhanSuList: [],
  DonHangKhamSucKhoeVatTuList: [],
  PhanTramKinhDoanhList: [],
};

export const khamSucKhoeDoanSlice = createSlice({
  name: "khamSucKhoeDoan",
```

```
  initialState,
  reducers: {
    showLoading: (state) => {
      state.loading = true;
    },
  },
  extraReducers: (builder) => {
    builder
      //GetAllPhanTramKinhDoanh
      .addCase(GetAllPhanTramKinhDoanh.pending, (state) => {
        state.loading = true;
      })
      .addCase(GetAllPhanTramKinhDoanh.fulfilled, (state, action) => {
        state.loading = false;
        state.PhanTramKinhDoanhList = action.payload;
      })
      .addCase(GetAllPhanTramKinhDoanh.rejected, (state, action) => {
        state.loading = false;
      })
      //GetAllDonHangKhamSucKhoeVatTu
      .addCase(GetAllDonHangKhamSucKhoeVatTu.pending, (state) => {
        state.loading = true;
      })
      .addCase(GetAllDonHangKhamSucKhoeVatTu.fulfilled, (state, action) => {
        state.loading = false;
        state.DonHangKhamSucKhoeVatTuList = action.payload;
      })
      .addCase(GetAllDonHangKhamSucKhoeVatTu.rejected, (state, action) => {
        state.loading = false;
      })
      //GetAllDonHangKhamSucKhoeNhanSu
      .addCase(GetAllDonHangKhamSucKhoeNhanSu.pending, (state) => {
        state.loading = true;
      })
      .addCase(GetAllDonHangKhamSucKhoeNhanSu.fulfilled, (state, action) => {
        state.loading = false;
        state.DonHangKhamSucKhoeNhanSuList = action.payload;
      })
      .addCase(GetAllDonHangKhamSucKhoeNhanSu.rejected, (state, action) => {
        state.loading = false;
      })
      //GetThanhLyHopDongChiTietKhac
      .addCase(GetThanhLyHopDongChiTietKhac.pending, (state) => {
        state.loading = true;
      })
      .addCase(GetThanhLyHopDongChiTietKhac.fulfilled, (state, action) => {
        state.loading = false;
        state.ThanhLyHopDongChiTietKhacList = action.payload;
      })
      .addCase(GetThanhLyHopDongChiTietKhac.rejected, (state, action) => {
        state.loading = false;
      })
      //GetBaoCaoDoanhThu
      .addCase(GetBaoCaoDoanhThu.pending, (state) => {
        state.loading = true;
      })
      .addCase(GetBaoCaoDoanhThu.fulfilled, (state, action) => {
        state.loading = false;
        state.BaoCaoDoanhThuList = action.payload;
      })
      .addCase(GetBaoCaoDoanhThu.rejected, (state, action) => {
        state.loading = false;
      })
      //GetBaoCaoCongNo
      .addCase(GetBaoCaoCongNo.pending, (state) => {
        state.loading = true;
      })
      .addCase(GetBaoCaoCongNo.fulfilled, (state, action) => {
        state.loading = false;
        state.BaoCaoCongNoList = action.payload;
      })
      .addCase(GetBaoCaoCongNo.rejected, (state, action) => {
        state.loading = false;
      })
```

```
//GetThuTien
.addCase(GetThuTien.pending, (state) => {
  state.loading = true;
})
.addCase(GetThuTien.fulfilled, (state, action) => {
  state.loading = false;
  state.ThuTienDetail = action.payload;
})
.addCase(GetThuTien.rejected, (state, action) => {
  state.loading = false;
})
//GetAllThuTien
.addCase(GetAllThuTien.pending, (state) => {
  state.loading = true;
})
.addCase(GetAllThuTien.fulfilled, (state, action) => {
  state.loading = false;
  state.ThuTienList = action.payload;
})
.addCase(GetAllThuTien.rejected, (state, action) => {
  state.loading = false;
})
//GetThanhLyHopDongChiTiet
.addCase(GetThanhLyHopDongChiTiet.pending, (state) => {
  state.loading = true;
})
.addCase(GetThanhLyHopDongChiTiet.fulfilled, (state, action) => {
  state.loading = false;
  state.ThanhLyHopDongKskChiTietList = action.payload;
})
.addCase(GetThanhLyHopDongChiTiet.rejected, (state, action) => {
  state.loading = false;
})
//GetThanhLyHopDongKsk
.addCase(GetThanhLyHopDongKsk.pending, (state) => {
  state.loading = true;
})
.addCase(GetThanhLyHopDongKsk.fulfilled, (state, action) => {
  state.loading = false;
  state.ThanhLyHopDongKskDetail = action.payload;
})
.addCase(GetThanhLyHopDongKsk.rejected, (state, action) => {
  state.loading = false;
})
//GetAllThanhLyHopDongKsk
.addCase(GetAllThanhLyHopDongKsk.pending, (state) => {
  state.loading = true;
})
.addCase(GetAllThanhLyHopDongKsk.fulfilled, (state, action) => {
  state.loading = false;
  state.ThanhLyHopDongKskList = action.payload;
})
.addCase(GetAllThanhLyHopDongKsk.rejected, (state, action) => {
  state.loading = false;
})
//GetAllCongTy
.addCase(GetAllCongTy.pending, (state) => {
  state.loading = true;
})
.addCase(GetAllCongTy.fulfilled, (state, action) => {
  state.loading = false;
  state.CongTyList = action.payload;
})
.addCase(GetAllCongTy.rejected, (state, action) => {
  state.loading = false;
})
//GetAllCongTyDto
.addCase(GetAllCongTyDto.fulfilled, (state, action) => {
  state.loading = false;
  state.CongTyDtoList = action.payload;
})
//GetAllDmDichVu
.addCase(GetAllDmDichVu.pending, (state) => {
  state.loading = true;
```

```
})
.addCase(GetAllDmDichVu.fulfilled, (state, action) => {
  state.loading = false;
  state.DmDichVuList = action.payload;
})
.addCase(GetAllDmDichVu.rejected, (state, action) => {
  state.loading = false;
})
//GetAllDoanKhamSucKhoe
.addCase(GetAllDoanKhamSucKhoe.pending, (state) => {
  state.loading = true;
})
.addCase(GetAllDoanKhamSucKhoe.fulfilled, (state, action) => {
  state.loading = false;
  state.DoanKhamList = action.payload;
})
.addCase(GetAllDoanKhamSucKhoe.rejected, (state, action) => {
  state.loading = false;
})
//GetDoanKhamSucKhoe
.addCase(GetDoanKhamSucKhoe.pending, (state) => {
  state.loading = true;
})
.addCase(GetDoanKhamSucKhoe.fulfilled, (state, action) => {
  state.loading = false;
  state.DoanKhamSucKhoeDetail = action.payload;
})
.addCase(GetDoanKhamSucKhoe.rejected, (state, action) => {
  state.loading = false;
})
//GetAllHangMucKham
.addCase(GetAllHangMucKham.pending, (state) => {
  state.loading = true;
})
.addCase(GetAllHangMucKham.fulfilled, (state, action) => {
  state.loading = false;
  state.HangMucKhamList = action.payload;
})
.addCase(GetAllHangMucKham.rejected, (state, action) => {
  state.loading = false;
})
//GetAllMauKham
.addCase(GetAllMauKham.pending, (state) => {
  state.loading = true;
})
.addCase(GetAllMauKham.fulfilled, (state, action) => {
  state.loading = false;
  state.MauKhamList = action.payload;
})
.addCase(GetAllMauKham.rejected, (state, action) => {
  state.loading = false;
})
//GetAllPhuongAnKinhDoanhKsk
.addCase(GetAllPhuongAnKinhDoanhKsk.pending, (state) => {
  state.loading = true;
})
.addCase(GetAllPhuongAnKinhDoanhKsk.fulfilled, (state, action) => {
  state.loading = false;
  state.PhuongAnKinhDoanhList = action.payload;
})
.addCase(GetAllPhuongAnKinhDoanhKsk.rejected, (state, action) => {
  state.loading = false;
})
//GetPhuongAnKinhDoanh
.addCase(GetPhuongAnKinhDoanh.pending, (state) => {
  state.loading = true;
})
.addCase(GetPhuongAnKinhDoanh.fulfilled, (state, action) => {
  state.loading = false;
  state.PhuongAnKinhDoanhDetail = action.payload;
})
.addCase(GetPhuongAnKinhDoanh.rejected, (state, action) => {
  state.loading = false;
})
```

```
      //GetPhuongAnKinhDoanhChiTiet
      .addCase(GetPhuongAnKinhDoanhChiTiet.pending, (state) => {
        state.loading = true;
      })
      .addCase(GetPhuongAnKinhDoanhChiTiet.fulfilled, (state, action) => {
        state.loading = false;
        state.PhuongAnKinhDoanhChiTietList = action.payload;
      })
      .addCase(GetPhuongAnKinhDoanhChiTiet.rejected, (state, action) => {
        state.loading = false;
      })
      //GetAllDonHangKhamSucKhoe
      .addCase(GetAllDonHangKhamSucKhoe.pending, (state) => {
        state.loading = true;
      })
      .addCase(GetAllDonHangKhamSucKhoe.fulfilled, (state, action) => {
        state.loading = false;
        state.DonHangKhamSucKhoeList = action.payload;
      })
      .addCase(GetAllDonHangKhamSucKhoe.rejected, (state, action) => {
        state.loading = false;
      })
      //GetDonHangKhamSucKhoe
      .addCase(GetDonHangKhamSucKhoe.pending, (state) => {
        state.loading = true;
      })
      .addCase(GetDonHangKhamSucKhoe.fulfilled, (state, action) => {
        state.loading = false;
        state.DonHangKhamSucKhoeDetail = action.payload;
      })
      .addCase(GetDonHangKhamSucKhoe.rejected, (state, action) => {
        state.loading = false;
      });
  },
});

export const { showLoading } = khamSucKhoeDoanSlice.actions;

export default khamSucKhoeDoanSlice.reducer;
```