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

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

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

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

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

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

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

export const thutienSlice = createSlice({
  name: "thutien",
  initialState,
  reducers: {
    showLoading: (state) => {
      state.loading = true;
    },
    setThutienList: (state, action) => {
      state.thutienList = action.payload;
    },
  },
  extraReducers: (builder) => {
    builder
      .addCase(getDSThuChi.pending, (state) => {
        state.loading = true;
      })
      .addCase(getDSThuChi.fulfilled, (state, action) => {
        state.loading = false;
        state.thutienList = action.payload;
      })
      .addCase(getDSThuChi.rejected, (state, action) => {
        state.loading = false;
      })
      .addCase(getPhieuThuChi.fulfilled, (state, action) => {
        state.loading = false;
        state.phieuThuChi = action.payload;
      });
  },
});

export const { showLoading, setThutienList } = thutienSlice.actions;

export default thutienSlice.reducer;
```