

```

import React, { useEffect, useRef, useState } from "react";

import { useDispatch, useSelector } from "react-redux";

import { SystemButtonSource } from "constants";
//import moment from "moment";
import MeditechTablePage, { FilterComponent } from "components/table-layout";
import { notification, Select } from "antd";

import {
  DeleteDonHangKhamSucKhoe,
  GetAllDmDichVu,
  GetAllDoanKhamSucKhoe,
  GetAllDonHangKhamSucKhoe,
  GetAllHangMucKham,
  GetAllMauKham,
  GetAllPhuongAnKinhDoanhKsk,
} from "store/slices/khamSucKhoeDoanSlice";
import ModalDonHang from "../ModalDonHang";
import DanhSachDonHang from "../DanhSachDonHang";
import { getAllEmployee } from "store/slices/employeeSlice";

const initSearch = {
  startDate: null, //moment(moment().now).month(-6)
  endDate: null, //moment(moment().now)
  searchText: null,
};

const listTrangThai = [
  { value: 0, label: "Chuẩn bị triển khai" },
  { value: 1, label: "Đang khám" },
  { value: 2, label: "Đang hoàn thiện hồ sơ" },
  { value: 3, label: "Thanh lý hợp đồng" },
];

const DonKhamSucKhoe = ({ queryId, openModalDetail }) => {
  const [visibleModal, setVisibleModal] = useState(false);
  const [donHangId, setDonHangId] = useState(null);
  const { branchId } = useSelector((state) => state.auth);
  const dispatch = useDispatch();
  const { loading, DonHangKhamSucKhoeList } = useSelector((state) => state.ksk);

  // eslint-disable-next-line react-hooks/exhaustive-deps
  const reloadData = () => {
    searchFormRef.current = {
      ...searchFormRef.current,
      chiNhanhId: branchId,
    };
    dispatch(GetAllDonHangKhamSucKhoe(searchFormRef.current));
  };

  useEffect(() => {
    if (openModalDetail) {
      setVisibleModal(openModalDetail);
      setDonHangId(queryId);
    }
  }, [queryId, openModalDetail]);

  useEffect(() => {
    reloadData();
    // eslint-disable-next-line react-hooks/exhaustive-deps
  }, [dispatch, branchId]);

  useEffect(() => {
    dispatch(GetAllHangMucKham({ loai: 1 }));
    dispatch(GetAllMauKham({ loai: 2 }));
    dispatch(GetAllDoanKhamSucKhoe({ chiNhanhId: branchId }));
    dispatch(GetAllDmDichVu({ ...initSearch }));
    dispatch(
      GetAllPhuongAnKinhDoanhKsk({
        ...initSearch,
        chiNhanhId: branchId,
      })
    );
  });

```

```

    trangThai: [1],
  });
}, [dispatch, branchId]);

useEffect(() => {
  dispatch(getAllEmployee({}));
}, [dispatch]);
const searchFormRef = useRef({ ...initSearch });
const [searchFormData, setSearchFormData] = useState(searchFormRef.current);
const searchHandle = (dataSearch) => {
  searchFormRef.current = { ...searchFormRef.current, ...dataSearch };
  setSearchFormData({ ...searchFormRef.current });
  reloadData();
};
const onRenderSearchView = () => {
  return (
    <FilterComponent
      datasource={searchFormData}
      onSearchImmediate={searchHandle}
      // renderFilterMoreView={renderFilterMoreView}
      renderInputSearch
      renderDatePicker
    >
    <Select
      key="trangThai"
      data-field="trangThai"
      style={{ width: "100%", minWidth: 200 }}
      placeholder="Trạng thái"
      options={listTrangThai}
      allowClear
      mode="multiple"
    />
    </FilterComponent>
  );
};

const onDeleteHandler = (id) => {
  dispatch(
    DeleteDonHangKhamSucKhoe({
      id: id,
      onSuccess: ({ data }) => {
        if (data.isOk) {
          reloadData();
          notification.info({
            message: "Xóa đơn hàng",
            description: "Xóa phiếu thành công",
          });
        } else {
          notification.error({
            message: "Xóa đơn hàng",
            description: data.description,
          });
        }
      },
    })
  );
};

const onOpenDetailHandler = (id) => {
  setDonHangId(id);
  setVisibleModal(true);
};

const getRightButton = (isLocked) => {
  const arr = [];
  if (isLocked) {
    arr.push({
      keyName: SystemButtonSource.addBtn,
      title: "Thêm",
      showMore: false,
      // disable: !allowAddNew,
    });
  }
}

```

```

    return arr;
  };

  const onRightTopBtnHandler = (key) => {
    switch (key) {
      case SystemButtonSource.addBtn:
        setVisibleModal((prev) => !prev);
        break;
      default:
        break;
    }
  };

  return (
    <>
      <MeditechTablePage
        tableTitle="ĐƠN HÀNG KHÁM SỨC KHỎE"
        btnHeaderRightSource={getRightButton(true)}
        btnHeaderRightSourceHandle={onRightTopBtnHandler}
        onRenderSearchView={onRenderSearchView}
      >
        <DanhSachDonHang
          DonHangKhamSucKhoeList={DonHangKhamSucKhoeList}
          loading={loading}
          donHangId={donHangId}
          setDonHangId={setDonHangId}
          reloadData={reloadData}
          onOpenDetail={onOpenDetailHandler}
          onDelete={onDeleteHandler}
        />
      </MeditechTablePage>

      <ModalDonHang
        visibleModal={visibleModal}
        setVisibleModal={setVisibleModal}
        loading={loading}
        donHangId={donHangId}
        setDonHangId={setDonHangId}
        reloadData={reloadData}
      />
    </>
  );
};

export default DonKhamSucKhoe;

```