

```

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 ModalDoanKham from "../ModalDoanKham";
import {
  DeleteDoanKhamSucKhoe,
  GetAllCongTyDto,
  GetAllDoanKhamSucKhoe,
} from "store/slices/khamSucKhoeDoanSlice";
import DanhSachDoanKham from "../DanhSachDoanKham";

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

const listTrangThai = [
  { value: 0, label: "Chưa phương án" },
  { value: 1, label: "Đã lên phương án" },
];

const DoanKham = ({ queryId, openModalDetail }) => {
  const [visibleModal, setVisibleModal] = useState(false);
  const [doanKhamId, setDoanKhamId] = useState(null);
  const { branchId } = useSelector((state) => state.auth);
  const dispatch = useDispatch();
  const { loading, DoanKhamList } = useSelector((state) => state.ksk);

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

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

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

  useEffect(() => {
    dispatch(GetAllCongTyDto());
  }, [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(
    DeleteDoanKhamSucKhoe({
      id: id,
      onSuccess: ({ data }) => {
        if (data.isOk) {
          console.log("vào đây", data);
          reloadData();
          notification.info({
            message: "Xóa đoàn",
            description: "Xóa phiếu thành công",
          });
        } else {
          notification.error({
            message: "Xóa đoàn",
            description: data.description,
          });
        }
      },
    })
  );
};

const onOpenDetailHandler = (id) => {
  setDoanKhamId(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="ĐOÀN KHÁM SỨC KHỎE"
      btnHeaderRightSource={getRightButton(true)}
      btnHeaderRightSourceHandle={onRightTopBtnHandler}
      onRenderSearchView={onRenderSearchView}
    >

```

```
>
  <DanhSachDoanKham
    DoanKhamList={DoanKhamList}
    loading={loading}
    doanKhamId={doanKhamId}
    setDoanKhamId={setDoanKhamId}
    reloadData={reloadData}
    onOpenDetail={onOpenDetailHandler}
    onDelete={onDeleteHandler}
  />
</MeditechTablePage>

<ModalDoanKham
  visibleModal={visibleModal}
  setVisibleModal={setVisibleModal}
  loading={loading}
  doanKhamId={doanKhamId}
  setDoanKhamId={setDoanKhamId}
  reloadData={reloadData}
/>
</>
);
};
export default DoanKham;
```