

```

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 {
  DeletePhuongAnKinhDoanh,
  GetAllDmDichVu,
  GetAllDoanKhamSucKhoe,
  GetAllHangMucKham,
  GetAllMauKham,
  GetAllPhuongAnKinhDoanhKsk,
} from "store/slices/khamSucKhoeDoanSlice";
import ModalPhuongAn from "../ModalPhuongAn";
import DanhSachPhuongAn from "../DanhSachPhuongAn";

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

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

const PhuongAnKinhDoanh = ({ queryId, openModalDetail }) => {
  const [visibleModal, setVisibleModal] = useState(false);
  const [phuongAnId, setPhuongAnId] = useState(null);
  const { branchId } = useSelector((state) => state.auth);
  const dispatch = useDispatch();
  const { loading, PhuongAnKinhDoanhList } = useSelector((state) => state.ksk);

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

  useEffect(() => {
    if (openModalDetail) {
      setVisibleModal(openModalDetail);
      setPhuongAnId(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, branchId]);

  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(
    DeletePhuongAnKinhDoanh({
      id: id,
      onSuccess: ({ data }) => {
        if (data.isOk) {
          reloadData();
          notification.info({
            message: "Xóa phương án kinh doanh",
            description: "Xóa phiếu thành công",
          });
        } else {
          notification.error({
            message: "Xóa phương án kinh doanh",
            description: data.description,
          });
        }
      },
    })
  );
};

```

```

const onOpenDetailHandler = (id) => {
  setPhuongAnId(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="PHƯƠNG AN KINH DOANH"
  btnHeaderRightSource={getRightButton(true)}
  btnHeaderRightSourceHandle={onRightTopBtnHandler}
  onRenderSearchView={onRenderSearchView}
>
  <DanhSachPhuongAn
    PhuongAnKinhDoanhList={PhuongAnKinhDoanhList}
    loading={loading}
    phuongAnId={phuongAnId}
    setPhuongAnId={setPhuongAnId}
    reloadData={reloadData}
    onOpenDetail={onOpenDetailHandler}
    onDelete={onDeleteHandler}
  />
</MeditechTablePage>

<ModalPhuongAn
  visibleModal={visibleModal}
  setVisibleModal={setVisibleModal}
  loading={loading}
  phuongAnId={phuongAnId}
  setPhuongAnId={setPhuongAnId}
  reloadData={reloadData}
/>
</>
);
};
export default PhuongAnKinhDoanh;

```