

```
import { Button, Form, Input, Spin, Switch, Tag } from "antd";
import React, { useMemo, useState } from "react";
import Utils from "utils";
```

```
import { PrinterOutlined } from "@ant-design/icons";
import MediTable from "components/Custom";
import ActionButtons from "components/ActionButtons";
import DocumentViewer from "constants/DocumentViewer";
```

```
export const initHoSo = {
  action: "initial",
  isRequired: true,
};
```

```
const EditableCell = ({
  editing,
  dataIndex,
  title,
  record,
  index,
  children,
  ...restProps
}) => {
  return (
    <td {...restProps}>
      {editing ? (
        dataIndex === "status" ? (
          <Form.Item
            name={dataIndex}
            style={{
              margin: 0,
            }}
            valuePropName="checked"
          >
            <Switch />
          </Form.Item>
        ) : (
          <Form.Item
            name={dataIndex}
            style={{
              margin: 0,
            }}
            rules={[
              {
                required: true,
                message: `Bắt buộc!`,
              },
            ]}
          >
            <Input placeholder={title} />
          </Form.Item>
        )
      ) : (
        children
      )}
    </td>
  );
};
```

```
const DanhSachDonHang = ({
  onOpenDetail,
  onDelete,
  DonHangKhamSucKhoeList,
  loading,
}) => {
  const [isPrinter, setIsPrinter] = useState(false);
  const [printData, setPrintData] = useState({});
  const [donHangId, setDonHangId] = useState(null);
```

```
const tableColumns = useMemo(
  () => [
```

```

{
  title: "STT",
  align: "center",
  render: (_, __, index) => index + 1,
},
{
  title: "Mã",
  dataIndex: "maDonHang",
  editable: true,
},
{
  title: "Phương án",
  dataIndex: "tenPhuongAn",
  editable: true,
},
{
  title: "Tên đoàn",
  dataIndex: "tenDoanKham",
  editable: true,
},
{
  title: "Hạng mục khám",
  dataIndex: "hangMucKham",
  editable: true,
},
{
  title: "Mẫu khám",
  dataIndex: "mauKham",
  editable: true,
},
{
  title: "Người tạo",
  dataIndex: "createdByName",
  editable: true,
},
{
  title: "Ngày tạo",
  dataIndex: "createdAt",
  render: (value) => value && Utils.formatDate(value),
},
{
  title: "Trạng thái",
  align: "center",
  dataIndex: "trangThai",
  editable: true,
  render: (value, record) =>
    record.id && (
      <Tag
        color={
          value === 3
            ? "green"
            : value === 2
            ? "blue"
            : value === 1
            ? "yellow"
            : "red"
        }
      >
        {value === 3
          ? "Thanh lý hợp đồng"
          : value === 2
          ? "Đang hoàn thiện hồ sơ"
          : value === 1
          ? "Đang khám"
          : "Chuẩn bị triển khai"}
        </Tag>
      ),
},
{
  fixed: "right",
  align: "center",
  render: (_, record, index) => {
    return (
      <>

```

```

        <ActionButtons
          record={record}
          editable={false}
          edit={() => {
            if (onOpenDetail) onOpenDetail(record.id);
          }}
          handleDelete={() => {
            if (onDelete) onDelete(record.id);
          }}
        />

        <Button
          onClick={() => {
            setDonHangId(record.id);
            setIsPrinter(true);
          }}
          icon={<PrinterOutlined />}
          shape="circle"
        />
      />
    />
  ),
],
[onDelete, onOpenDetail]
);

const mergedColumns = tableColumns.map((col) => {
  return {
    ...col,
    onCell: (record) => ({
      record,
      dataIndex: col.dataIndex,
      title: col.title,
      //editing: isEditing(record),
    }),
  };
});
return (
  <>
    <Spin tip="Đang tải..." spinning={loading}>
      <MediTable
        components={{
          body: {
            cell: EditableCell,
          },
        }}
        tableColumns={mergedColumns}
        dataSource={DonHangKhamSucKhoeList}
        scroll={{ x: "max-content" }}
        rowKey={(item) => item?.id}
      />
    </Spin>

    <DocumentViewer
      openModel={isPrinter}
      setIsPrinter={setIsPrinter}
      soPhieu={printData.sophieue}
      idPhieu={donHangId}
      setPrintData={setPrintData}
      tenMauIn={"DonKSK.rptDonKSK"}
      loaiPhieu={"DonKsk"}
      tieuDe={`In đơn KSK ${printData.sophieue || ""}`}
      param={{ id: donHangId }}
    />
  </>
);
};

export default DanhSachDonHang;

```

