

```

import { Form, Input, Spin, Switch } from "antd";
import React, { useMemo } from "react";

import MediTable from "components/Custom";
import ActionButtons from "components/ActionButtons";
import Utils from "utils";

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 DSBaoCaoDoanhThu = ({ onOpenDetail, BaoCaoDoanhThuList, loading }) => {
  const tableColumns = useMemo(
    () => [
      {
        title: "STT",
        align: "center",
        width: "80px",
        render: (_, __, index) => index + 1,
      },
      {
        title: "Công ty",
        dataIndex: "tenCongTy",
        editable: true,
      },
    ],
  );

```

```

{
  title: "Tên đoàn",
  dataIndex: "tenDoan",
  editable: true,
},
{
  title: "Địa chỉ",
  dataIndex: "diaChi",
  editable: true,
  width: "250px",
},
{
  title: "Tổng tiền (1)",
  dataIndex: "tongTienThucTe",
  editable: true,
  width: "140px",
  align: "right",
  render: (e) => Utils.formatterNumber(e),
},
{
  title: "Chiết khấu ĐV(2)",
  dataIndex: "chietKhau",
  editable: true,
  width: "140px",
  align: "right",
  render: (e) => Utils.formatterNumber(e),
},
{
  title: "Chi phí khác (3)",
  dataIndex: "chiPhiKhac",
  editable: true,
  width: "140px",
  align: "right",
  render: (e) => Utils.formatterNumber(e),
},
{
  title: "Chiết khấu NVKD ",
  dataIndex: "chietKhauKd",
  editable: true,
  width: "140px",
  align: "right",
  render: (e) => Utils.formatterNumber(e),
},
{
  title: "Chiết khấu NVHT ",
  dataIndex: "chietKhauHt",
  editable: true,
  width: "140px",
  align: "right",
  render: (e) => Utils.formatterNumber(e),
},

{
  fixed: "right",
  align: "center",
  width: "100px",
  render: (_, record, index) => {
    return (
      <>
      <ActionButtonButtons
        record={record}
        editable={false}
        edit={() => {
          if (onOpenDetail) onOpenDetail(record.id);
        }}
      />
    </>
  );
},
],
[onOpenDetail]
);

```

```

const mergedColumns = tableColumns.map((col) => {
  return {
    ...col,
    onCell: (record) => ({
      record,
      dataIndex: col.dataIndex,
      title: col.title,
      //editing: isEditing(record),
    }),
  };
});

const Footer = useMemo(() => {
  let tongTienThucTe = 0;
  let chietKhau = 0;
  let chiPhiKhac = 0;
  for (let index = 0; index <= BaoCaoDoanhThuList.length - 1; index++) {
    tongTienThucTe += BaoCaoDoanhThuList[index].tongTienThucTe;
    chietKhau += BaoCaoDoanhThuList[index].chietKhau;
    chiPhiKhac += BaoCaoDoanhThuList[index].chiPhiKhac;
  }

  var strChietKhauKd = [];
  BaoCaoDoanhThuList.reduce(function (res, value) {
    if (!res[value.ptNvKd]) {
      res[value.ptNvKd] = { ptNvKd: value.ptNvKd, chietKhauKd: 0 };
      strChietKhauKd.push(res[value.ptNvKd]);
    }
    res[value.ptNvKd].chietKhauKd += value.chietKhauKd;
    return res;
  }, {});

  var strKd = strChietKhauKd.map(
    (m) =>
    `Chiết khấu NVKD (${m.ptNvKd}%): ${Utils.formatterNumber(
      m.chietKhauKd
    )}`
  );

  var strChietKhauHt = [];
  BaoCaoDoanhThuList.reduce(function (res, value) {
    if (!res[value.ptNvHt]) {
      res[value.ptNvHt] = { ptNvHt: value.ptNvHt, chietKhauHt: 0 };
      strChietKhauHt.push(res[value.ptNvHt]);
    }
    res[value.ptNvHt].chietKhauHt += value.chietKhauHt;
    return res;
  }, {});

  var strHt = strChietKhauHt.map(
    (m) =>
    `Chiết khấu NVHT (${m.ptNvHt}%): ${Utils.formatterNumber(
      m.chietKhauHt
    )}`
  );

  return (
    <>
    <h5>Tổng cộng: ({Utils.formatterNumber(tongTienThucTe)})</h5>
    <h5>Tổng chiết khấu: ({Utils.formatterNumber(chietKhau)})</h5>
    <h5>
      Tổng (1)-(2): ({Utils.formatterNumber(tongTienThucTe - chietKhau)})
    </h5>
    <h5>
      Tổng (1)-(2)-(3): (
        {Utils.formatterNumber(tongTienThucTe - chietKhau - chiPhiKhac)})
    </h5>

    <h5>{strKd}</h5>
    <h5>{strHt}</h5>
    </>
  );
}, [BaoCaoDoanhThuList]);

```

```

return (
  <>
    <Spin tip="Đang tải..." spinning={loading}>
      <MediTable
        components={{
          body: {
            cell: EditableCell,
          },
        }}
        tableColumns={mergedColumns}
        dataSource={BaoCaoDoanhThuList}
        scroll={{ x: "max-content" }}
        rowKey={{(item) => item?.id}}
        onRenderFooter={() => Footer}
      />
    </Spin>
  </>
);
};

export default DSBaoCaoDoanhThu;

```