

```

import React from "react";
import {
  Input,
  InputNumber,
  Checkbox,
  Form,
  Switch,
  DatePicker,
  Select,
} from "antd";
import { dateFormatList } from "constants";
import { dateTimeFormatList } from "constants";
import Utils from "utils";
export const parserNumber = (val) => {
  if (!val) return 0;
  return Number.parseFloat(
    // eslint-disable-next-line no-useless-escape
    val.replace(/\$\s?|(\.+)/g, "").replace(/(\,|{1})/g, ".")
  ).toFixed(2);
};
const EditableCell = ({
  editing,
  dataIndex,
  title,
  inputType,
  record,
  index,
  children,
  isRequired = false,
  ...restProps
}) => {
  const inputNode =
    inputType === "number" ? (
      <InputNumber
        placeholder={title}
        min={0}
        formatter={(value) => Utils.formatterNumber(value)}
        parser={(value) => parserNumber(value)}
      />
    ) : inputType === "checkbox" ? (
      <Checkbox />
    ) : inputType === "switch" ? (
      <Switch />
    ) : inputType === "select" ? (
      <Select
        value={record.groupId}
        options={
          restProps?.options.map((x) => ({ value: x.id, label: x.name })) || []
        }
        // onChange={(value) => {
        //   restProps.form.setFieldValue(dataIndex, value);
        // }}
        allowClear
        showSearch
        optionFilterProp="children"
        filterOption={(input, option) =>
          (option?.label ?? "").toLowerCase().includes(input.toLowerCase())
        }
        filterSort={(optionA, optionB) =>
          (optionA?.label ?? "")
            .toLowerCase()
            .localeCompare((optionB?.label ?? "").toLowerCase())
        }
        mode={restProps?.mode || ""}
      />
    ) : inputType === "date" ? (
      <DatePicker placeholder={title} format={dateFormatList} />
    ) : inputType === "datetime" ? (
      <DatePicker
        placeholder={title}
        format={dateTimeFormatList}

```

```

      showTime={{ format:"HH:mm" }}
    />
  ) : (
    <Input placeholder={title} />
  );
return (
  <td {...restProps}>
    {editing ? (
      <Form.Item
        name={dataIndex}
        style={{ margin: 0 }}
        valuePropName={
          ["checkbox", "switch"].includes(inputType) ? "checked" : "value"
        }
        rules={[
          {
            required: isRequired,
            message: `bắt buộc`,
          },
        ]}
      >
        {inputNode}
      </Form.Item>
    ) : (
      children
    )}
  </td>
);
};

export default EditableCell;

```