(https://github.com/AllenFang/react-bootstrap-table)

ReactBsTable (index.html)

React-bootstrap-table

Basic Table

Use `<BootstrapTable>` to define table and `<TableHeaderColumn>` to defined column
Use `data` to specify the data that you want to display on table

Product ID	**Product Name**	**Product Price**
1	Product1	101
2	Product2	102
3	Product3	103
4	Product4	104
5	Product5	105
6	Product6	106

```
<BootstrapTable data={products}>
  <TableHeaderColumn dataField="id" isKey={true}>Product ID</TableHeaderColumn>
  <TableHeaderColumn dataField="name">Product Name</TableHeaderColumn>
  <TableHeaderColumn dataField="price">Product Price</TableHeaderColumn>
</BootstrapTable>
```

Table Style

Like basic Bootstrap table classes, ex: .table-striped or .table-hover, reactBsTable support that all.
Use `striped` to make a striped table
Use `hover` to enable hover on table
Use `condensed` to make a condensed table

Product ID	**Product Name**	**Product Price**
1	Product1	101
2	Product2	102
3	Product3	103
4	Product4	104
5	Product5	105
6	Product6	106

```
<BootstrapTable data={products} striped={true} hover={true} condensed={true}>
  <TableHeaderColumn dataField="id" isKey={true}>Product ID</TableHeaderColumn>
  <TableHeaderColumn dataField="name">Product Name</TableHeaderColumn>
  <TableHeaderColumn dataField="price">Product Price</TableHeaderColumn>
</BootstrapTable>
```

Table Scroll

Use `height` to set the height on table, default is 100%
If table content overflow, reactBsTable will enable scrolling on table auto.

Product ID	Product Name	Product Price
1	Product1	101
2	Product2	102

```
<BootstrapTable data={products} height="120">
  <TableHeaderColumn dataField="id" isKey={true}>Product ID</TableHeaderColumn>
  <TableHeaderColumn dataField="name">Product Name</TableHeaderColumn>
  <TableHeaderColumn dataField="price">Product Price</TableHeaderColumn>
</BootstrapTable>
```

Column Width

Use `width` to set the column width. Default unit is Pixels.

Prod...	Product Name	Product Price

1	Product1	101
2	Product2	102
3	Product3	103
4	Product4	104
5	Product5	105
6	Product6	106

```
<BootstrapTable data={products}>
  <TableHeaderColumn dataField="id" isKey={true} width="70">Product ID</TableHeaderColumn>
  <TableHeaderColumn dataField="name" width="150">Product Name</TableHeaderColumn>
  <TableHeaderColumn dataField="price">Product Price</TableHeaderColumn>
</BootstrapTable>
```

Column Align

Use `dataAlign` to set the column align, default is left. Available value is left, center, right, start and end.

Product ID	Product Name	Product Price
1	Product1	101
2	Product2	102
3	Product3	103
4	Product4	104
5	Product5	105
6	Product6	106

```
<BootstrapTable data={products}>
  <TableHeaderColumn dataField="id" isKey={true} dataAlign="right">Product I
D</TableHeaderColumn>
  <TableHeaderColumn dataField="name" dataAlign="center">Product Name</Table
HeaderColumn>
  <TableHeaderColumn dataField="price">Product Price</TableHeaderColumn>
</BootstrapTable>
```

Column Sort

Use `dataSort` to enable sort on column. You can click the column header to sort column.
Use `sortFunc` **(optional)** to assign a custom sort function for column. See example (advance.html#customSort)

Product ID ▾▴	Product Name ▾▴	Product Price
1	Product1	101
2	Product2	102
3	Product3	103
4	Product4	104
5	Product5	105
6	Product6	106

```
<BootstrapTable data={products}>
  <TableHeaderColumn dataField="id" isKey={true} dataSort={true}>Product ID
</TableHeaderColumn>
  <TableHeaderColumn dataField="name" dataSort={true}>Product Name</TableHea
derColumn>
  <TableHeaderColumn dataField="price">Product Price</TableHeaderColumn>
</BootstrapTable>
```

Column Format

Use `dataFormat` to customize your column.

Product ID	Product Name	Product Price
1	Product1	$ 101
2	Product2	$ 102
3	Product3	$ 103
4	Product4	$ 104
5	Product5	$ 105
6	Product6	$ 106

```
function priceFormatter(cell, row){
  return '<i class="glyphicon glyphicon-usd"></i> ' + cell;
}
<BootstrapTable data={products}>
  <TableHeaderColumn dataField="id" isKey={true}>Product ID</TableHeaderColu
mn>
  <TableHeaderColumn dataField="name">Product Name</TableHeaderColumn>
  <TableHeaderColumn dataField="price" dataFormat={priceFormatter}>Product P
rice</TableHeaderColumn>
</BootstrapTable>
```

Table Pagination

Use `pagination` to enable the pagination on your table.

Product ID	Product Name	Product Price

0	Item name 0	100
1	Item name 1	101
2	Item name 2	102
3	Item name 3	103
4	Item name 4	104
5	Item name 5	105
6	Item name 6	106
7	Item name 7	107
8	Item name 8	108

10 ▾

1 2 3 4 5 > >>

```
</BootstrapTable>
```

Row selection(Single)

Use `selectRow` to specify the row selection detail. In addition, selectRow accept a object
This is object contain these properties:

`mode` : radio/checkbox **(required)**, to specify the single selection or multi selection in table.

`clickToSelect` : true/false **(optional)**, if true, click the row will trigger selection on that row, default is false

`bgColor` : css color **(optional)**, if assigned, this color will present on backgroud if row selection.

`onSelect` : callback function **(optional)**, if a row be selected, this function will be called.

	Product ID	Product Name	Product Price
○	1	Product1	101
○	2	Product2	102
○	3	Product3	103
○	4	Product4	104
○	5	Product5	105
○	6	Product6	106

```
function onRowSelect(row, isSelected){
  console.log(row);
  console.log("selected: " + isSelected)
}

var selectRowProp = {
  mode: "radio",
  clickToSelect: true,
  bgColor: "rgb(238, 193, 213)",
  onSelect: onRowSelect
};
<BootstrapTable data={products} selectRow={selectRowProp}>
  <TableHeaderColumn dataField="id" isKey={true}>Product ID</TableHeaderColumn>
  <TableHeaderColumn dataField="name">Product Name</TableHeaderColumn>
  <TableHeaderColumn dataField="price">Product Price</TableHeaderColumn>
</BootstrapTable>
```

Row selection(Multiple)

Use `selectRow` to specify the row selection detail. In addition, selectRow accept a object
This is object contain these properties:
`mode` : radio/checkbox **(required)**, to specify the single selection or multi selection in table.
`clickToSelect` : true/false **(optional)**, if true, click the row will trigger selection on that row, default is false
`bgColor` : css color **(optional)**, if assigned, this color will present on backgroud if row selection.
`onSelect` : callback function **(optional)**, if a row be selected, this function will be called.
`onSelectAll` : callback function **(optional)**, if select all in checkbox mode, this function will be called.

□	Product ID	Product Name	Product Price

☐	1	Product1	101
☐	2	Product2	102
☐	3	Product3	103
☐	4	Product4	104
☐	5	Product5	105
☐	6	Product6	106

```
function onRowSelect(row, isSelected){
  console.log(row);
  console.log("selected: " + isSelected)
}

function onSelectAll(isSelected){
  console.log("is select all: " + isSelected);
}
var selectRowProp = {
  mode: "checkbox",
  clickToSelect: true,
  bgColor: "rgb(238, 193, 213)",
  onSelect: onRowSelect,
  onSelectAll: onSelectAll
};
<BootstrapTable data={products} selectRow={selectRowProp}>
  <TableHeaderColumn dataField="id" isKey={true}>Product ID</TableHeaderColumn>
  <TableHeaderColumn dataField="name">Product Name</TableHeaderColumn>
  <TableHeaderColumn dataField="price">Product Price</TableHeaderColumn>
</BootstrapTable>
```

Cell Edit

Use `cellEdit` to enable cell editing on your table and this attribute accept a object.
This is object contain these properties:
`mode` : click/dbclick **(required)** to specify which condition will trigger cell editing.
`blurToSave` : true/false **(optional)** if true, when mouse blur on input field will trigger a save on cell, default is false so you need to press ENTER for saving cell.
`afterSaveCell` : callback function **(optional)** accept a callback function, after save cell, this function will be called.

In default condition, all columns is editable except for **key column** `isKey={true}`
If you want some column can't be edited, use `editable={false}` on `<TableHeaderColumn>`

Product ID	Product Name	Product Price
1	Product1	101
2	Product2	102
3	Product3	103
4	Product4	104
5	Product5	105
6	Product6	106

```
function onAfterSaveCell(row, cellName, cellValue){
  console.log("Save cell '"+cellName+"' with value '"+cellValue+"'");
  console.log("Thw whole row :");
  console.log(row);
}
var cellEditProp = {
  mode: "click",
  blurToSave: true,
  afterSaveCell: onAfterSaveCell
}
<BootstrapTable data={products} cellEdit={cellEditProp}>
  <TableHeaderColumn dataField="id" isKey={true}>Product ID</TableHeaderColumn>
  <TableHeaderColumn dataField="name">Product Name</TableHeaderColumn>
  <TableHeaderColumn dataField="price" editable={false}>Product Price</TableHeaderColumn>
</BootstrapTable>
```

Insert & Delete Row

Use `insertRow` to enable the insert row on table.
Use `deleteRow` to enable the delete row on table. If true, you must use `selectRow` to enable row selection.

✚ New 🗑 Delete

	Product ID	Product Name	Product Price
○	1	Product1	101
○	2	Product2	102
○	3	Product3	103
○	4	Product4	104
○	5	Product5	105
○	6	Product6	106

```
var selectRowProp = {
  mode: "radio", // or checkbox
  clickToSelect: true
};
<BootstrapTable data={products} insertRow={true} deleteRow={true} selectRow={selectRowProp}>
  <TableHeaderColumn dataField="id" isKey={true}>Product ID</TableHeaderColumn>
  <TableHeaderColumn dataField="name">Product Name</TableHeaderColumn>
  <TableHeaderColumn dataField="price">Product Price</TableHeaderColumn>
</BootstrapTable>
```

Column Filter

Use `columnFilter` to enable the column filter on table.

Product ID	Product Name	Product Price

1	Product1	101
2	Product2	102
3	Product3	103
4	Product4	104
5	Product5	105
6	Product6	106

id name price

```
<BootstrapTable data={products} columnFilter={true}>
  <TableHeaderColumn dataField="id" isKey={true}>Product ID</TableHeaderColu
mn>
  <TableHeaderColumn dataField="name">Product Name</TableHeaderColumn>
  <TableHeaderColumn dataField="price">Product Price</TableHeaderColumn>
</BootstrapTable>
```

Table Search

Use `search` to enable the column filter on table.
Use `searchPlaceholder` **(optional)** to change the placeholder in search field.

Search

Product ID	Product Name	Product Price

1	Product1	101
2	Product2	102
3	Product3	103
4	Product4	104
5	Product5	105
6	Product6	106

```
<BootstrapTable data={products} search={true}>
  <TableHeaderColumn dataField="id" isKey={true}>Product ID</TableHeaderColu
mn>
  <TableHeaderColumn dataField="name">Product Name</TableHeaderColumn>
  <TableHeaderColumn dataField="price">Product Price</TableHeaderColumn>
</BootstrapTable>
```

Update table data on fly

react-bootstrap-table support the client can update data on fly through a TableDataSet object. See the example in the following or reference About TableDataSet (./docs.html#about-tabledataset).

Click me!

Product ID	Product Name	Product Price
1	Product1	101
2	Product2	102
3	Product3	103
4	Product4	104
5	Product5	105
6	Product6	106

```
var ReactBsTable  = require('react-bootstrap-table');
var BootstrapTable = ReactBsTable.BootstrapTable;
var TableHeaderColumn = ReactBsTable.TableHeaderColumn;
var TableDataSet = ReactBsTable.TableDataSet; //import TableDataSet

var dataSet = new TableDataSet(products);
<BootstrapTable data={dataSet}>
  <TableHeaderColumn dataField="id" isKey={true}>Product ID</TableHeaderColumn>
  <TableHeaderColumn dataField="name">Product Name</TableHeaderColumn>
  <TableHeaderColumn dataField="price">Product Price</TableHeaderColumn>
</BootstrapTable>
// ...
// in any time, you want to update or reset data on table, you can:
dataSet.setData(newProdcuts);
// the table data will be updated auto.
```

Column Hidden

Use `hidden` as attribute in `TableHeaderColumn` to enable the column hidden.

Product ID	**Product Price**
1	101
2	102
3	103
4	104
5	105
6	106

```
<BootstrapTable data={products}>
  <TableHeaderColumn dataField="id" isKey={true}>Product ID</TableHeaderColumn>
  <TableHeaderColumn dataField="name" hidden={true}>Product Name</TableHeaderColumn>
  <TableHeaderColumn dataField="price">Product Price</TableHeaderColumn>
</BootstrapTable>
```

React-bootstrap-table (https://github.com/AllenFang/react-bootstrap-table) is maintained by AllenFang (https://github.com/AllenFang). This page was generated by GitHub Pages (https://pages.github.com) using the Cayman theme (https://github.com/jasonlong/cayman-theme) by Jason Long (https://twitter.com/jasonlong).