

[Pro](#) [Teams](#) [Pricing](#) [Documentation](#)[Sign Up](#)[Sign In](#)[Search](#)

# react-multi-carousel TS

2.8.5 • [Public](#) • Published 6 days ago

[Readme](#)[Code](#) Beta

0 Dependencies

164 Dependents

133 Versions

## react-multi-carousel 🙌

financial contributors 1

all contributors 8

Production-ready, lightweight fully customizable React carousel component that rocks supports multiple items and SSR(Server-side rendering).

quality ★★★★½

downloads 591k/month

documentation yes

Maintained? yes

License MIT

### Hello world!

We are on a very excited journey towards version 3.0 of this component which will be rewritten in hooks/context completely. It means smaller bundle size, performance

improvement and easier customization of the component and so many more benefits.

It would mean so much if you could provide help towards the further development of this project as we do this open source work in our own free time especially during this covid-19 crisis.

If you are using this component seriously, please donate or talk to your manager as this project increases your income too. It will help us make releases, fix bugs, fulfill new feature requests faster and better.

**Become a backer/sponsor** to get your logo/image on our README on Github with a link to your site.

## Features.

- Server-side rendering
- Infinite mode
- Dot mode
- Custom animation
- AutoPlay mode
- Auto play interval
- Supports images, videos, everything.
- Responsive
- Swipe to slide
- Mouse drag to slide
- Keyboard control to slide
- Multiple items
- Show / hide arrows
- Custom arrows / control buttons
- Custom dots
- Custom styling
- Accessibility support
- Center mode.
- Show next/previous set of items partially
- RTL support

## Shoutouts 🙏

Big thanks to **BrowserStack** for letting the maintainers use their service to debug browser issues.

## Documentation

---

### Other important links.

---

- [Code sandbox](#)
- [Contributing](#)
- [Changelog](#)
- [Releases](#)
- [TypeScript usage](#)
- [SSR demo](#)

## Bundle size

---

**Bundle-size.** 2.5kB

## Demo.

---

Documentation is [here](#).

Demo for the SSR <https://react-multi-carousel.now.sh/>

Try to disable JavaScript to test if it renders on the server-side.

Codes for SSR at [github](#).

Codes for the documentation at [github](#).

# Install

---

```
$ npm install react-multi-carousel --save
```

```
import Carousel from 'react-multi-carousel';  
import 'react-multi-carousel/lib/styles.css';
```

## How the SSR mode works?

Codes for SSR at [github](#).

- Demo for the SSR are at [here](#)
- Try to disable JavaScript to test if it renders on the server-side.

Here is a lighter version of the library for detecting the user's device type **alternative**

You can choose to only bundle it on the server-side.

## Minimum working set up.

---

```
import Carousel from "react-multi-carousel";  
import "react-multi-carousel/lib/styles.css";  
const responsive = {  
  superLargeDesktop: {  
    // the naming can be any, depends on you.  
    breakpoint: { max: 4000, min: 3000 },  
    items: 5  
  },  
  desktop: {  
    breakpoint: { max: 3000, min: 1024 },  
    items: 3  
  },  
  tablet: {  
    breakpoint: { max: 1024, min: 464 },  
    items: 2  
  }  
};
```

```
    },
    mobile: {
      breakpoint: { max: 464, min: 0 },
      items: 1
    }
  };
<Carousel responsive={responsive}>
  <div>Item 1</div>
  <div>Item 2</div>
  <div>Item 3</div>
  <div>Item 4</div>
</Carousel>;
```

## Common Usage

---

```
import Carousel from "react-multi-carousel";
import "react-multi-carousel/lib/styles.css";

const responsive = {
  desktop: {
    breakpoint: { max: 3000, min: 1024 },
    items: 3,
    slidesToSlide: 3 // optional, default to 1.
  },
  tablet: {
    breakpoint: { max: 1024, min: 464 },
    items: 2,
    slidesToSlide: 2 // optional, default to 1.
  },
  mobile: {
    breakpoint: { max: 464, min: 0 },
    items: 1,
    slidesToSlide: 1 // optional, default to 1.
  }
};
```

```
    }  
  };  
  <Carousel  
    swipeable={false}  
    draggable={false}  
    showDots={true}  
    responsive={responsive}  
    ssr={true} // means to render carousel on server-side.  
    infinite={true}  
    autoplay={this.props.deviceType !== "mobile" ? true : false}  
    autoplaySpeed={1000}  
    keyboardControl={true}  
    customTransition="all .5"  
    transitionDuration={500}  
    containerClass="carousel-container"  
    removeArrowOnDeviceType={["tablet", "mobile"]}  
    deviceType={this.props.deviceType}  
    dotListClass="custom-dot-list-style"  
    itemClass="carousel-item-padding-40-px"  
  >  
    <div>Item 1</div>  
    <div>Item 2</div>  
    <div>Item 3</div>  
    <div>Item 4</div>  
  </Carousel>;
```

## Custom Arrows.

You can pass your own custom arrows to make it the way you want, the same for the position. For example, add media query for the arrows to go under when on smaller screens.

Your custom arrows will receive a list of props/state that's passed back by the carousel such as the `currentSide`, is dragging or swiping in progress.

## Code

```
const CustomRightArrow = ({ onClick, ...rest }) => {
  const {
    onMove,
    carouselState: { currentSlide, deviceType }
  } = rest;
  // onMove means if dragging or swiping in progress.
  return <button onClick={() => onClick()} />;
};
<Carousel customRightArrow={<CustomRightArrow />} />;
```

## Custom button group.

This is very useful if you don't want the dots, or arrows and you want to fully customize the control functionality and styling yourself.

## Code

```
const ButtonGroup = ({ next, previous, goToSlide, ...rest }) => {
  const { carouselState: { currentSlide } } = rest;
  return (
    <div className="carousel-button-group"> // remember to give it pos
      <ButtonOne className={currentSlide === 0 ? 'disable' : ''} onClick={previous} />
      <ButtonTwo onClick={() => next()} />
      <ButtonThree onClick={() => goToSlide(currentSlide + 1)}> Go to
    </div>
  );
};
<Carousel arrows={false} customButtonGroup={<ButtonGroup />>
  <ItemOne>
  <ItemTwo>
</Carousel>
```

# renderButtonGroupOutside

Passing this props would render the button group outside of the Carousel container. This is done using React.fragment

```
<div className='my-own-custom-container'>
  <Carousel arrows={false} renderButtonGroupOutside={true} customButtons={
    <ItemOne>
    <ItemTwo>
  } />
</div>
```

## Custom dots.

You can pass your own custom dots to replace the default one.

Custom dots can also be a copy or an image of your carousel item. See example in this **one**

The codes for this **example**

You custom dots will receive a list of props/state that's passed back by the carousel such as the currentSlide, is dragging or swiping in progress.

### Code

```
const CustomDot = ({ onClick, ...rest }) => {
  const {
    onMove,
    index,
    active,
    carouselState: { currentSlide, deviceType }
  } = rest;
  const carouselItems = [CarouselItem1, CarouselItem2, CarouselItem3];
```



```
// onMove means if dragging or swiping in progress.
// active is provided by this lib for checking if the item is active
return (
  <button
    className={active ? "active" : "inactive"}
    onClick={() => onClick()}
  >
    {React.Children.toArray(carouselItems)[index]}
  </button>
);
};
<Carousel showDots customDot={<CustomDot />}>
  {carouselItems}
</Carousel>;
```

## renderDotsOutside

Passing this props would render the dots outside of the Carousel container. This is done using React.fragment

```
<div className='my-own-custom-container'>
  <Carousel arrows={false} showDots={true} renderDotsOutside={renderBu
    <ItemOne>
    <ItemTwo>
  </Carousel>
</div>
```

## partialVisible props.

Shows the next items partially, this is very useful if you want to indicate to the users that this carousel component is swipable, has more items behind it.

This is different from the "centerMode" prop, as it only shows the next items. For the centerMode, it shows both.

```
const responsive = {
  desktop: {
    breakpoint: { max: 3000, min: 1024 },
    items: 3,
    partialVisibilityGutter: 40 // this is needed to tell the amount c
  },
  tablet: {
    breakpoint: { max: 1024, min: 464 },
    items: 2,
    partialVisibilityGutter: 30 // this is needed to tell the amount c
  },
  mobile: {
    breakpoint: { max: 464, min: 0 },
    items: 1,
    partialVisibilityGutter: 30 // this is needed to tell the amount c
  }
}

<Carousel partialVisible={true} responsive={responsive}>
  <ItemOne />
  <ItemTwo />
</Carousel>
```

## centerMode

Shows the next items and previous items partially.

```
<Carousel centerMode={true} />
```

## afterChange callback.

This is a callback function that is invoked each time when there has been a sliding.

```
<Carousel
  afterChange={(previousSlide, { currentSlide, onMove }) => {
    doSpeicalThing();
  }}
/>
```

## beforeChange call back

This is a callback function that is invoked each time before a sliding.

```
<Carousel
  beforeChange={(nextSlide, { currentSlide, onMove }) => {
    doSpeicalThing();
  }}
/>
```

## Combine beforeChange and nextChange, real usage.

They are very useful in the following cases:

- The carousel item is clickable, but you don't want it to be clickable while the user is dragging it or swiping it.

```
<Carousel
  beforeChange={() => this.setState({ isMoving: true })}
  afterChange={() => this.setState({ isMoving: false })}
>
  <a
    onClick={e => {
      if (this.state.isMoving) {
```

```

        e.preventDefault();
      }
    }}
    href="https://w3js.com"
  >
    Click me
  </a>
</Carousel>

```

- Preparing for the next slide.

```

<Carousel beforeChange={nextSlide => this.setState({ nextSlide: nextSlide })}
  <div>Initial slide</div>
  <div
    onClick={() => {
      if (this.state.nextSlide === 1) {
        doVerySpecialThing();
      }
    }}
  >
    Second slide
  </div>
</Carousel>

```

## Skipping callbacks

When calling the `goToSlide` function on a `Carousel` the callbacks will be run by default. You can skip all or individual callbacks by passing a second parameter to `goToSlide`.

```

this.Carousel.goToSlide(1, true); // Skips both beforeChange and afterChange
this.Carousel.goToSlide(1, { skipBeforeChange: true }); // Skips only beforeChange

```

```
this.Carousel.goToSlide(1, { skipAfterChange: true }); // Skips only a
```

## focusOnSelect

Go to slide on click and make the slide a current slide.

```
<Carousel focusOnSelect={true} />
```

## Using ref.

```
<Carousel ref={(el) => (this.Carousel = el)} arrows={false} responsive
  <ItemOne />
  <ItemTwo />
</Carousel>
<button onClick={() => {
  const nextSlide = this.Carousel.state.currentSlide + 1;
  // this.Carousel.next()
  // this.Carousel.goToSlide(nextSlide)
}}>Click me</button>
```

## additionalTransfrom Props.

This is very useful when you are fully customizing the control functionality by yourself like this **one**

### Code

For example if you give to your carousel item padding left and padding right 20px. And you have 5 items in total, you might want to do the following:

```
<Carousel ref={el => (this.Carousel = el)} additionalTransfrom={-20 *
```

# Specific Props

Name	Type	Default	Description
responsive	object	{}	Numbers of slides to show at each breakpoint
deviceType	string	' '	Only pass this when use for server-side rendering, what to pass can be found in the example folder
ssr	boolean	false	Use in conjunction with responsive and deviceType prop
slidesToSlide	Number	1	How many slides to slide.
draggable	boolean	true	Optionally disable/enable dragging on desktop
swipeable	boolean	true	Optionally disable/enable swiping on mobile
arrows	boolean	true	Hide/Show the default arrows

Name	Type	Default	Description
renderArrowsWhenDisabled	boolean	false	Allow for the arrows to have a disabled attribute instead of not showing them
removeArrowOnDeviceType	string or array	''	Hide the default arrows at different break point, should be used with responsive props. Value could be mobile or ['mobile', 'tablet'], can be a string or array
customLeftArrow	jsx	null	Replace the default arrow with your own
customRightArrow	jsx	null	Replace the default arrow with your own
customDot	jsx	null	Replace the default dots with your own
customButtonGroup	jsx	null	Fully customize your own control

Name	Type	Default	Description
			functionality if you don't want arrows or dots
infinite	boolean	false	Enables infinite scrolling in both directions. Carousel items are cloned in the DOM to achieve this.
minimumTouchDrag	number	50	The amount of distance to drag / swipe in order to move to the next slide.
afterChange	function	null	A callback after sliding everytime.
beforeChange	function	null	A callback before sliding everytime.
sliderClass	string	'react-multi-carousel-track'	CSS class for inner slider div, use this to style your own track list.
itemClass	string	''	CSS class for carousel item, use this to style your own Carousel item.



Name	Type	Default	Description
			For example add padding-left and padding-right
containerClass	string	'react-multi-carousel-list'	Use this to style the whole container. For example add padding to allow the "dots" or "arrows" to go to other places without being overflowed.
dotListClass	string	'react-multi-carousel-dot-list'	Use this to style the dot list.
keyBoardControl	boolean	true	Use keyboard to navigate to next/previous slide
autoPlay	boolean	false	Auto play
autoPlaySpeed	number	3000	The unit is ms
showDots	boolean	false	Hide the default dot list
renderDotsOutside	boolean	false	Show dots outside of the container

Name	Type	Default	Description
partialVisible	boolean	string	false
customTransition	string	transform 300ms ease-in-out	Configure your own animation when sliding
transitionDuration	number	300	The unit is ms, if you are using customTransition, make sure to put the duration here as this is needed for the resizing to work.
focusOnSelect	boolean	false	Go to slide on click and make the slide a current slide.
centerMode	boolean	false	Shows the next items and previous items partially.
additionalTransfrom	number	0	additional transfrom to the current one.
shouldResetAutoplay	boolean	true	resets autoplay when clicking next, previous

Name	Type	Default	Description
			button and the dots
rewind	boolean	false	if infinite is not enabled and autoPlay explicitly is, this option rewinds the carousel when the end is reached (Lightweight infinite mode alternative without cloning).
rewindWithAnimation	boolean	false	when rewinding the carousel back to the beginning, this decides if the rewind process should be instant or with transition.
rtl	boolean	false	Sets the carousel direction to be right to left

---

## Author

---

 **Yi Zhuang**

- Github: [@YIZHUANG](#)



# Contribute

---

Please read <https://github.com/YIZHUANG/react-multi-carousel/blob/master/contributing.md>

Submit an issue for feature request or submit a pr.

## Local development.

---

- cd app
- npm install
- npm run dev

## Donation

---

If this project help you reduce time to develop, you can give me a cup of coffee :)

## Contributors

---

### Code Contributors

This project exists thanks to all the people who contribute. [[Contribute](#)].

### Financial Contributors

Become a financial contributor and help us sustain our community. [[Contribute](#)]

### Individuals

---

### Organizations









---

Support this project with your organization. Your logo will show up here with a link to your website. [[Contribute](#)]

# License

## Contributors ✨

Thanks goes to these wonderful people (**emoji key**):

<b>Truong Hoang Dung</b> 	<b>Tobias Pinnekamp</b> 	<b>Rajendran Nadar</b> 	<b>Abhinav Dalal</b> 	<b>Oscar Barrett</b> 	<b>Neamat Mim</b> 	<b>Martin Retrou</b> 
<b>Ben Hodgson</b> 	<b>Faizan ul haq</b> 	<b>Adrian3PG</b> 	<b>kuznetsovgm</b> 	<b>Vadim Filimonov</b> 	<b>Romain</b> 	<b>Riley Lundquist</b> 
<b>Paul Deshaies Jr</b> 	<b>Pavel Mikheev</b> 	<b>nev1d</b> 	<b>Mads Vammen</b> 	<b>Jiro Farah</b> 		

This project follows the **all-contributors** specification. Contributions of any kind welcome!

## Keywords

**react typescript carousel react-component server-side rendering  
ssr image-gallery video-gallery slide-show swipe drag touch**

**multiple items gallery slider responsive mobile-friendly  
tablet-friendly breakpoint**

## Install

```
> npm i react-multi-carousel
```



## Repository

 [github.com/YIZHUANG/react-multi-carousel](https://github.com/YIZHUANG/react-multi-carousel)

## Homepage

 [react-multi-carousel.surge.sh](https://react-multi-carousel.surge.sh)

## ± Weekly Downloads

**97,630**

## Version

**2.8.5**

## License

**MIT**

## Unpacked Size

**1.63 MB**

## Total Files

**90**

## Last publish

**6 days ago**

## Collaborators

[>Try on RunKit](#)

 **Report** malware



## Support

[Help](#)

[Advisories](#)

[Status](#)

[Contact npm](#)

## Company

[About](#)

[Blog](#)

[Press](#)

## Terms & Policies

[Policies](#)

[Terms of Use](#)

[Code of Conduct](#)

[Privacy](#)

