

A set of react transition components for basic animations.

📄 MIT License

★ 43 stars 🔗 4 forks

☆ Star

👁 Watch ▼

Code

Issues

Pull requests

Actions

Security

Insights

🔗 master ▼

...

Chris Johnson ...

✓ on 4 Apr 2018 🕒

[View code](#)

☰ README.md

react-animation-components

build passing

A set of React components using [React Transition Group](#) to provide drop in GPU accelerated animations and wrappers for group effects.

[Checkout the Storybook!](#)

- [Installation](#)
- [Animation Components](#)
 - [Fade](#)
 - [Transform](#)
 - [FadeTransform](#)
- [Wrapper Components](#)
 - [Stagger](#)
 - [Random](#)
 - [Loop](#)

Installation

```
npm install react-animation-components
```

Make sure you also have installed the following peer dependencies:

```
"react": "^16.0.0",
"react-dom": "^16.0.0",
"react-transition-group": "^2.2.1",
"prop-types": "^15.6.0"
```

Animation Components

Props available on all animation components

The following are available on any animation component as well as **any valid `Transition` props**. Transitions are set to `appear` and their `timeout` is calculated by combining the `delay` and `duration` by default but can be overwritten.

Key	Description	Example	Type	Default Value
className	Passes className to wrapper <code>div</code>	<code>some-class</code>	<i>string</i>	<code>undefined</code>
delay	Sets the animations <code>transitionDelay</code>	<code>500</code>	<i>number</i>	<code>0</code>
duration	Sets the animations <code>transitionDuration</code>	<code>1000</code>	<i>number</i>	<code>500</code>
style	Passes styles to wrapper <code>div</code>	<pre>{ display: 'flex' }</pre>	<i>object</i>	<code>{}</code>
timingFn	Sets the animations <code>transitionTimingFunction</code>	<code>'ease-in-out'</code>	<i>string</i>	<code>'ease'</code>

Fade

Transitions the wrapped element's opacity from one value to another

Props

Key	Description	Example	Type	Default Value
enterOpacity	The opacity value when <code>in</code> is <code>true</code>	<code>0.85</code>	<i>number</i>	<code>0</code>
exitOpacity	The opacity value when <code>in</code> is <code>false</code>	<code>0.25</code>	<i>number</i>	<code>0</code>

Examples

```
import { Fade } from 'react-animation-components'

<Fade in>
  <h1>I'm transitioning to opacity:1</h1>
</Fade>

<Fade in enterOpacity={0.85}>
  <h1>I'm transitioning to opacity:0.85</h1>
</Fade>

<Fade in={false}>
  <h1>I'm transitioning to opacity:0</h1>
</Fade>

<Fade in={false} exitOpacity={0.25}>
  <h1>I'm transitioning to opacity:0.25</h1>
</Fade>
```

Transform

Transitions the wrapped element from one transform property to another. Any valid `transform` property will work.

Props

Key	Description	Example	Type	Default Value
enterTransform	The transform value when <code>in</code> is <code>true</code>	<code>'translateX(100px)'</code>	<i>string</i>	<code>'none'</code>
exitTransform	The transform value when <code>in</code> is <code>false</code>	<code>'translateX(100px)'</code>	<i>string</i>	<code>'none'</code>

Examples

```
import { Transform } from 'react-animation-components'

<Transform enterTransform="translateX(100px)" in>
  <h1>I'm transitioning from my initial position to 100px right when `in`
  is `true`</h1>
</Transform>

<Transform enterTransform="translateX(100px)"
  exitTransform="translateX(-100px)" in>
  <h1>
    I'm 100px to the left of my initial position and
    I transition 100px right of my initial when `in` is `true`
  </h1>
</Transform>

<Transform enterTransform="rotate(90deg)" in>
  <h1>I transition from initial position to rotate 90deg when `in` is
  `true`</h1>
</Transform>
```

FadeTransform

Composes `Fade` and `Transform`. All top level props are passed to both components. You can also pass props to individual components in the composition.

Props passed to individual components via `fadeProps` or `transformProps` will override any top level props

Props

Key	Description	Example	Type	Default Value
fadeProps	The props that only <code>Fade</code> receives.	<pre>{ enterOpacity: 0.85 }</pre>	<i>object</i>	<code>{}</code>
transformProps	The props that only <code>Transform</code> receives.	<pre>{ enterTransform: 'translateX(100px)' }</pre>	<i>object</i>	<code>{}</code>

Examples

```
import { FadeTransform } from 'react-animation-components'
```

```

<FadeTransform in transformProps={{ enterTransform: 'translateX(100px)' }}>
  <h1>I'm transitioning from my initial position to 100px right when `in`
  is `true`</h1>
</FadeTransform>

<FadeTransform
  in
  transformProps={{
    enterTransform: 'translateX(100px)',
    exitTransform: 'translateX(-100px)'
  }}
>
  <h1>
    I'm 100px to the left of my initial position and
    I transition 100px right of my initial when `in` is `true`
  </h1>
</FadeTransform>

<FadeTransform
  in
  transformProps={{
    exitTransform: 'translateX(-100px)'
  }}
  fadeProps={{
    enterOpacity: 0.85,
  }}
>
  <h1>I transition from `-100px` horizontally of my initial positon and to
  0.85 opacity when `in` is `true`</h1>
</FadeTransform>

```

Wrapper Components

Wrapper components use the inner animation components `onEntered` and `onExited`.
Setting those callbacks inside these wrappers will not work

Stagger

Uses `TransitionGroup` to stagger `delay` on a set of animation components

Props

Key	Description	Example	Type	Default Value
chunk	Used to limit the stagger into "chunks".	5	<i>number</i>	0

Key	Description	Example	Type	Default Value
delay	The amount to separate each stagger by	1000	number	100
duration	A value to set the inner child animations transition duration	800	number	500
in	A boolean to tell the children to mount or unmount	true	boolean	false
onComplete	A function that is called after the last animation finishes	any valid function	function	Function.prototype(noop)
reverse	A boolean to tell the component to reverse how delays are applied	true	boolean	false

Examples

```
import { Fade, Stagger } from 'react-animation-components'

const items = ['first', 'second', 'third', 'fourth', 'fifth'];

<Stagger in>
  {items.map(
    item => (
      <Fade>
        <h1>Each {item} will transition in with an incrementally
larger delay than the previous</h1>
      </Fade>
    )
  )}
```

```

    })
  </Stagger>

  <Stagger chunk={4} in>
    {items.map(
      item => (
        <Fade>
          <h1>
            Each {item} will increment in segments of 4.
            First is 0, Second is 100, Third is 200, Fourth is 0,
            fifth is 100, and so on
          </h1>
        </Fade>
      )
    )}
  </Stagger>

```

Random

Uses `TransitionGroup` to randomize `delay` on a set of animation components

Props

Key	Description	Example	Type	Default Value
duration	A value to set the inner child animations transition duration	800	number	500
in	A boolean to tell the children to mount or unmount	true	boolean	false
maxDelay	Maximum delay possible	5000	number	1500
minDelay	Minimum delay possible	100	number	0

Key	Description	Example	Type	Default Value
onComplete	A function that is called after the last animation finishes	any valid function	<i>function</i>	<code>Function.prototype(noop)</code>
reverse	A boolean to tell the component to reverse how delays are applied	<code>true</code>	<i>boolean</i>	<code>false</code>

Examples

```
import { Fade, Random } from 'react-animation-components'

const items = ['first', 'second', 'third', 'fourth', 'fifth'];

<Random in>
  {items.map(
    item => (
      <Fade>
        <h1>Each {item} will randomly FadeIn between 0 and
1500ms</h1>
      </Fade>
    )
  )}
</Random>

<Random minDelay={1000} maxDelay={5000} in>
  {items.map(
    item => (
      <Fade>
        <h1>Each {item} will randomly FadeIn between 1000ms and
5000ms</h1>
      </Fade>
    )
  )}
</Random>
```

Loop

Loops using the `onEntered` and `onExited` callbacks to toggle `in` on a **single** animation component.

Props

Key	Description	Example	Type	Default Value
in	Initializes the loop when <code>true</code>	<code>true</code>	<i>bool</i>	<code>false</code>
interval	Sets the interval to toggle <code>in</code> . Also sets the <code>duration</code>	<code>1000</code>	<i>number</i>	<code>500</code>
iterations	Maximum number of loops	<code>5.5</code>	<i>number</i>	<code>Infinity</code>
onComplete	Callback that is called when the <code>iterations</code> have been met. Waits an additional <code>interval</code> to ensure its called when the last iteration has completed	any valid function	<i>function</i>	<code>Function.prototype(noop)</code>
onIterate	Callback that is called with the current count each time the loop iterates. Count is incremented by <code>0.5</code>	any valid function	<i>function</i>	<code>Function.prototype(noop)</code>

Examples

```
import { Fade, Loop } from 'react-animation-components'
```

```
<Loop in>
  <Fade>
    <h1>I will Fade in and out repeatedly on 500ms intervals</h1>
  </Fade>
</Loop>

<Loop in iterations={5.5}>
  <Fade>
    <h1>I will Fade in and out repeatedly on 500ms intervals 5.5
times</h1>
  </Fade>
</Loop>
```

Releases 12

 **v3.0.0** Latest
on 4 Apr 2018

[+ 11 releases](#)

Packages

No packages published

Languages

● **JavaScript** 100.0%