

Top 5 React Lazy-Loading Libraries



Isuri Devindi



9 min read



Nov 20, 2024

Updated



TL;DR: *Discover the top 5 React lazy-loading libraries to optimize your applications effortlessly. Explore flexible options like **react-lazyload**, fast solutions like **react-lazy-load**, and more, tailored for seamless integration and enhanced performance.*

Lazy loading is one of the most popular methods to optimize React applications. It allows users to render content only when they need it, resulting in faster initial load time, reduced bandwidth consumption, and less data traffic.

Over the years, various packages were introduced to the React ecosystem to optimize applications through lazy loading. This article will go through the following five lazy-loading libraries popular in the React community:

- [react-lazyload](#)
- [react-lazy-load](#)
- [react-lazy-load-image-component](#)
- [react-lazy-images](#)

- [react-lazy-blur-image](#)

1. react-lazyload: Flexible customization

[react-lazyload](#) can be used to lazy load any type of component in a React application. It is one of the most popular lazy-loading libraries in the React community, supporting decorators and server-side rendering.

react-lazyload has around 892 thousand monthly NPM downloads and 5.4 thousand GitHub stars.

Features of react-lazyload

- Has a set of properties to enable users to customize the functionalities of the component.
- Provides utilities such as **forceCheck** to display hidden content that becomes visible without a resize or scroll event.
- Supports horizontal lazy load out of the box.
- Implements only two event listeners for all lazily loaded components, optimizing the performance further.

Installation

You can install and save the library using NPM as follows.

```
npm install --save react-lazyload
```

Usage

After installation, you can use react-lazyload as follows.

```
import React from 'react';
import LazyLoad from 'react-lazyload';
import LazyComponent from './LazyComponent'; //Component to be lazyloaded.
import Placeholder from './Placeholder'; //Placeholder component.

const App = () => {
  return (
    <div className="container">
      /\*\* Lazy loading images is supported by default without extra configurat
        set height for better experience \*\*/
      <LazyLoad height={200}>
        <img src={"https://picsum.photos/id/235/500/500"} />
      </LazyLoad>

      /\*\* Once this component is loaded, LazyLoad will not care about it anymo
        improving performance \*\*/
      <LazyLoad height={200} once>
```

[Copy](#)

```

    <img src={"https://picsum.photos/id/235/500/500"} />
  </LazyLoad>

  /\*\*Can also load components as below\*\*/
  <LazyLoad height={200} once >
    <LazyComponent />
  </LazyLoad>

  /\*\* This component will be loaded when its top edge is 100px from viewp
  Offset prop is useful to make user ignorant about lazy load effect. \*\*/
  <LazyLoad height={200} offset={100}>
    <img src={"https://picsum.photos/id/235/500/500"} />
  </LazyLoad>

  /\*\*Can use a placeholder component as bellow\*\*/
  <LazyLoad placeholder={<Placeholder/>}>
    <img src={"https://picsum.photos/id/235/500/500"} />
  </LazyLoad>

</div>
);
};

export default App;

```

Note: Check out its well-written [documentation](#) to learn more about react-lazyload and its limitations.



Syncfusion React UI components are the developers' choice to build user-friendly web applications. You deserve them too.

Explore Now

2. react-lazy-load: Incredibly fast!

[react-lazy-load](#) is a React component that can be used to defer loading content in a predictable way. It is comparatively fast, with a minimal bundle size (6KB minified).

react-lazy-load has 676 thousand monthly NPM downloads and 869 GitHub stars.

Features of react-lazy-load

- Automatically detects scrolling containers such as a `<div>` with a scrollbar and therefore can be used even inside a scrolling container.
- Lets users set the threshold to begin content loading from any side of the viewport, using props such as `offsetVertical`, `offsetHorizontal`, `offsetTop`, `offsetBottom`, etc.
- Works in IE8+.

- Has built-in support for debouncing functions.

However, compared to the react-lazyload library, this component has a very limited number of props and therefore provides less flexibility. The main disadvantage is that it doesn't allow users to apply a placeholder for the lazy-loaded component.

Installation

The library can be installed and saved to your React application using the following command.

```
$ npm install --save react-lazy-load
```

Usage

Here's how you can load a component using react-lazy-load.

```
import React from 'react';
import LazyLoad from 'react-lazy-load';

const App = () => {
  <div className='App'>
```

[Copy](#)

```
<div className="container">
  /\*\*With default offset 0, and height set for more clarity\*\*/
  <LazyLoad height={500}>
    <img src={"https://picsum.photos/id/235/500/500"} alt='demonstration1' /
  </LazyLoad>

  /\*\*OffsetBottom allows you to start loading content before reaching the
  <LazyLoad offsetBottom={200}>
    <img src={"https://picsum.photos/id/235/500/500"} alt='demonstration2' /
  </LazyLoad>

  /\*\*OffsetTop allows you to delay content loading after reaching the view
  <LazyLoad offsetTop={200}>
    <img src={"https://picsum.photos/id/235/500/500"} alt='demonstration3' /
  </LazyLoad>
</div>
</div>
</div>

export default App;
```

Note: Check out the react-lazy-load [documentation](#) to learn about the component in more detail.



A to Z about Syncfusion's versatile
React components and their feature

Read Now

set.

3. react-lazy-load-image-component: Compatible with TypeScript declarations

The [react-lazy-load-image-component](#) is an easy-to-use library for lazy loading any type of component. It supports the [IntersectionObserver](#), and you can determine when an element leaves and enters the viewport.

react-lazy-load-image-component has around 324 thousand monthly NPM downloads and 884 GitHub stars.

Features of react-lazy-load-image-component

- The most significant feature of this library is its HOC, `trackWindowScroll`, which allows components to track window scroll positions without using scroll or resize event listeners for every element.
- The lazy-loaded components will have an offset of 100 pixels by default.
- Built-in, on-visible effects such as blur, black and white, and opacity transitions help to improve the user experience.

- Server-side rendering compatible.
- Support for TypeScript declarations.
- A placeholder is provided by default with the same size as the image or component, though it can be customized.

Installation

You can install and save the package using the following command.

```
\\# Yarn
$ yarn add react-lazy-load-image-component
# NPM
$ npm install --save react-lazy-load-image-component
```

Usage

A simple use case of the LazyLoadImage component is shown.

```
import React from 'react';
import { LazyLoadImage } from 'react-lazy-load-image-component';
import Placeholder from './Placeholder'; //Placeholder component.
```

[Copy](#)

```
//Import the css file if using blur, black and white, or opacity effects.
import 'react-lazy-load-image-component/src/effects/blur.css';

const App = () => {
  <div className='App'>
    <div className="container">
      /\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\*Can use a placeholder for the component\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\*/
      <LazyLoadImage
        alt="demonstration1"
        placeholder={<Placeholder/>}
        width = {100}
        height = {100}
        src={"https://picsum.photos/id/235/500/500"}
      />

      /\\\\\\\\*Using the opacity effect\\\\\\\\*/
      <LazyLoadImage
        alt="demonstration2"
        effect="opacity"
        src={"https://picsum.photos/id/235/500/500"}
      />
    </div>
  </div>
}

export default App;
```

Note: You can find several implementation examples and their limitations in the

[documentation](#) of the library.

4. react-lazy-images: Fallback strategies for SEO- or JS-disabled environments

[react-lazy-images](#) is a flexible library that provides components and utilities to lazy load images in React. It gives full presentational control for the caller using render props.

react-lazy-images has around 51,000 monthly NPM downloads and 263 GitHub stars.

Features of react-lazy-images

- Uses IntersectionObserver with polyfill to improve performance.
- Provides fallback strategies for SEO- and JavaScript-disabled environments.
- Supports server-side rendering.
- Supports background images and works with horizontal scrolling.

Installation

The library package is distributed via NPM and can be installed as follows.

```
\\# Yarn
$ yarn add react-lazy-images
# NPM
$ npm install --save react-lazy-images
```

Usage

The following example shows how the basic functionality of the component can be implemented in React.

```
import React from 'react';
import { LazyImage } from "react-lazy-images";

const App = () => {
  <div className='App'>
    <div className="container">
      /\*\*Basic implementation.
      It is important that you pass on the ref in placeholder,
      otherwise the detection of the element intersecting is impossible.
      /\*\*/
    </div>
  </div>
}
```

[Copy](#)

```

<LazyImage
src={"https://picsum.photos/id/235/500/500"}
alt={"demonstration"}
debounceDurationMs={800}
placeholder={({ imageProps, ref }) => (
  <img
    ref={ref}
    src={"https://picsum.photos/id/235/500/500?blur=10"}
    alt={imageProps.alt}
    style={{ width: "100%" }}
  />
)}
actual={({ imageProps }) => (
  <img {...imageProps} style={{ width: "100%" }} />
)}
/>

</div>
</div>
}

export default App;

```

In addition to being a flexible component, react-lazy-images has an easy-to-understand source code and [documentation](#), explaining implementation details in depth with possible problems and their solutions.

5. react-lazy-blur-image: Provides a lightweight gray placeholder

[react-lazy-blur-image](#) is the ideal library to lazy load images into a low-resolution placeholder. By default, this component displays a lightweight, gray placeholder and is replaced with an actual placeholder when the component is about to reach the viewport. This placeholder will be replaced only when the actual image is lazily loaded completely.

react-lazy-blur-image has around 1,000 monthly NPM downloads.

Features of react-lazy-blur-image

- It provides a minimalistic approach to lazy load images, providing the perfect UX and performance balance.
- The component accepts only two props for customization: **src** and **style**.
- Component can use **styled-components** to transition an image from the placeholder.

Installation

You can install and save the package using the following commands.

```
\\# Yarn
$ yarn add react-lazy-blur-image
# NPM
$ npm install --save react-lazy-blur-image
```

Usage

Here's how you can lazy load an image using this library.

```
import React from 'react';
import LazyImage from 'react-lazy-blur-image';

const App = () => {
  <div className='App'>
    <div className="container">
      /\*\*Basic implementation\*\*/
      <LazyImage
        placeholder={"https://picsum.photos/id/235/500/500?blur=10"}
        uri={"https://picsum.photos/id/235/500/500"}
        render={({src,style}) => <img src={src} style={style} alt="demonstration"/>
      />

    </div>
  </div>
}
```

[Copy](#)


```
export default App;
```

Note: Check out the [documentation](#) to learn more about react-lazy-blur-image.



See the possibilities for yourself with live demos of Syncfusion React components.

Try Now

Wrap up

Among the many libraries available for lazy loading, this article reviewed 5 of the most popular and often-used libraries in the React community. Following is the summary of statistics for those libraries.



Summary of statistics of React lazy loading libraries

While each library has its pros and cons, it's your duty as a developer to select the best library for the requirements of your application to get the maximum benefit.

The [Syncfusion React](#) suite offers over 80 high-performance, lightweight, modular, and responsive UI components in a single package. This library is the only suite you'll ever need to construct a complete application.

If you have any questions or comments, you can contact us through our [support forums](#), [support portal](#), or [feedback portal](#). We are always happy to assist you!

Related blogs

- [Top 10 React Component Libraries Every Developer Should Know](#)
- [Everything You Should Know About React 18](#)
- [Create Dynamic Forms in React Using React Hook Forms](#)
- [Unit Testing in React with Jest and Enzyme Frameworks](#)



MEET THE AUTHOR

Isuri Devindi

I'm a software developer who's willing to explore out of her comfort zone and try to achieve perfection in any task assigned to me.

Undergraduate | University of Peradeniya | Faculty of Engineering