

## Blog

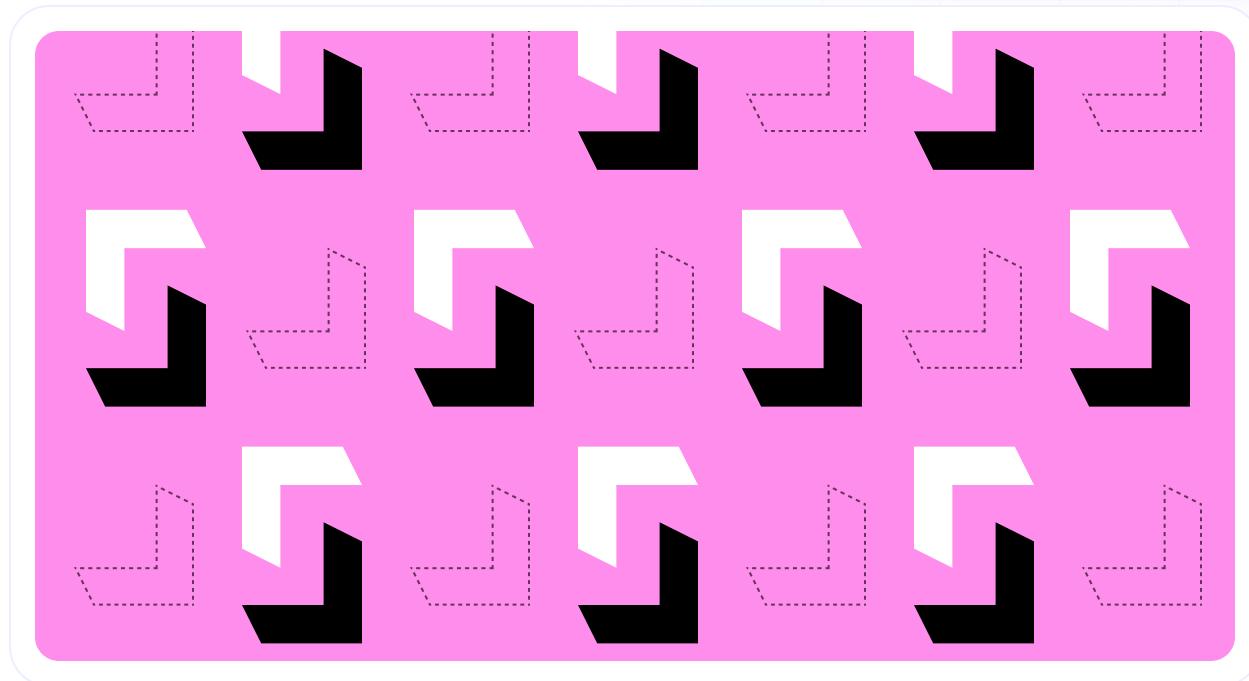
[Blog](#)[Viewpoint](#)

# How to Create Apple's Liquid Glass Effects in React



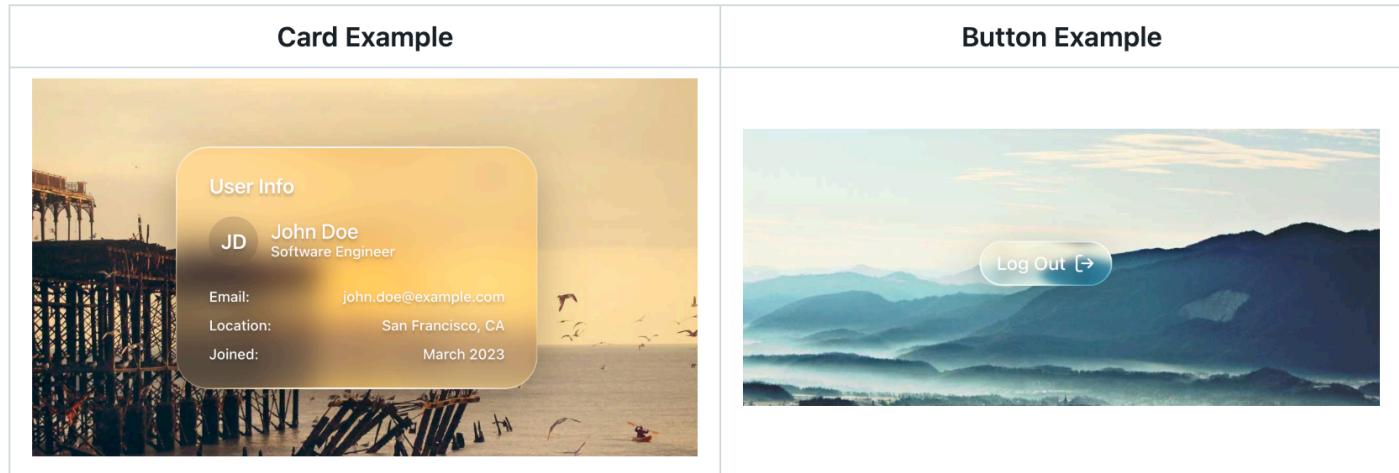
Lynn Mikami

14 June 2025



Apple has always been at the forefront of user interface design, and one of their most captivating recent effects is the "liquid glass" look. This effect, characterized by its fluid, ielly-

In this article, we'll explore how to recreate this stunning effect in your React applications using the `liquid-glass-react` library.



This library provides a highly customizable and easy-to-use component that encapsulates the complexity of the effect, allowing you to add a touch of Apple's design magic to your projects with minimal effort.

**GitHub - rdev/liquid-glass-react: Apple's Liquid Glass effect for React**

Apple's Liquid Glass effect for React. Contribute to rdev/liquid-glass-react development by creating an account on GitHub.

 GitHub • rdev

**rdev/liquid-glass-react**

Liquid Glass effect for React

3 Contributors   3 Issues   1k Stars   85 Forks



Want a great API Testing tool that generates beautiful API Documentation?

Want an integrated, All-in-One platform for your Developer Team to work together with [maximum productivity](#)?

Apidog delivers all your demands, and [replaces Postman at a much more affordable price!](#)

## How to Generate A Public API Documentation [Under 5 Minutes Guide]



[Sign Up for Free](#)

 [Download Now](#)

[For Windows or Linux >](#)

 Privacy protected

 Security guaranteed with no ads

## Getting Started

Before we dive into the intricacies of the liquid glass effect, let's get the `liquid-glass-react` library set up in our project.

## Installation

To install the library, simply run the following command in your terminal:

```
npm install liquid-glass-react
```

Or if you're using Yarn:

```
yarn add liquid-glass-react
```

## Basic Usage

Once the library is installed, you can start using the `LiquidGlass` component in your React application. Here's a basic example of how to use it:

```
import React from 'react';
import LiquidGlass from 'liquid-glass-react';

function App() {
  return (
    <div className="App">
      <LiquidGlass>
        <div style={{ padding: '20px' }}>
          Hello, Liquid Glass!
        </div>
      </LiquidGlass>
    </div>
  );
}

export default App;
```

This will render a basic liquid glass container around the "Hello, Liquid Glass!" text. While this is a good starting point, the true power of the library lies in its extensive customization options.

## Core Concepts

To fully appreciate and effectively use the `liquid-glass-react` library, it's helpful to understand the core concepts that power the effect. The liquid glass effect is primarily achieved through a combination of SVG filters, displacement maps, and chromatic aberration.

## SVG Filters

SVG (Scalable Vector Graphics) filters are a powerful feature of SVG that allows you to apply a wide range of effects to your graphics. In the case of the liquid glass effect, SVG filters are used to create the distortion and blur that give the component its characteristic look.

The `liquid-glass-react` library uses a series of `fe` (filter effect) primitives to achieve the desired effect. These include:

- `feGaussianBlur` : This primitive is used to create the background blur that is a key component of the glass effect.
- `feImage` : This primitive is used to load the displacement map, which is essential for creating the distortion.
- `feDisplacementMap` : This is the star of the show. It uses the pixel values from the displacement map to distort the source graphic.
- `feColorMatrix` : This primitive is used to manipulate the color channels of the image, which is how the chromatic aberration effect is created.
- `feBlend` : This primitive is used to combine the different layers of the effect, such as the red, green, and blue channels for the chromatic aberration.
- `feComposite` : This primitive is used to combine the final effect with the original source graphic.

## Displacement Maps

A displacement map is a grayscale image that is used to distort another image. The `feDisplacementMap` filter uses the color values of the displacement map to shift the pixels of the source graphic. Lighter areas of the displacement map will push the pixels of the source graphic in one direction, while darker areas will push them in the opposite direction.

The `Liquid-glass-react` library comes with three built-in displacement maps:

- `standard` : A standard, smooth displacement map.
- `polar` : A displacement map with a polar coordinate-based distortion.
- `prominent` : A displacement map with a more pronounced, "bubbly" effect.

You can also provide your own custom displacement map, or even generate one dynamically using a shader.

## Chromatic Aberration

Chromatic aberration is an optical effect that occurs when a lens fails to focus all colors to the same point. This results in a "fringing" of colors around the edges of objects. While this is generally considered an undesirable effect in photography, it can be used to create a stylish and futuristic look in digital graphics.

The `Liquid-glass-react` library simulates chromatic aberration by splitting the image into its red, green, and blue channels, and then displacing each channel by a slightly different amount. This creates a subtle but effective color fringing effect that adds to the overall "glossy" feel of the component.

## The `LiquidGlass` Component

Now that we have a basic understanding of the concepts behind the liquid glass effect, let's take a closer look at the `LiquidGlass` component and its props.

The `LiquidGlass` component is the main component of the library. It's a highly customizable component that allows you to control every aspect of the liquid glass effect.

Here are some of the key props that you can use to customize the component:

- `displacementScale` : Controls the intensity of the distortion effect. A higher value will result in a more distorted, "wobbly" look.
- `blurAmount` : Controls the amount of blur applied to the background.
- `saturation` : Controls the saturation of the background.
- `aberrationIntensity` : Controls the intensity of the chromatic aberration effect.
- `elasticity` : Controls how much the component "stretches" towards the mouse cursor. This creates a fun, interactive effect.
- `cornerRadius` : Controls the corner radius of the component.
- `mode` : Controls which displacement map is used. You can choose between `standard` , `polar` , `prominent` , and `shader` .
- `overLight` : A boolean that, when true, makes the glass darker for better visibility on light backgrounds.

## Customization

The real fun begins when you start to play with the props of the `LiquidGlass` component. By combining different props, you can create a wide range of unique and interesting effects.

Here's an example of a more customized `LiquidGlass` component:

```
import React from 'react';
import LiquidGlass from 'liquid-glass-react';

function App() {
  return (
    <div className="App">
      <LiquidGlass
        displacementScale={100}
```

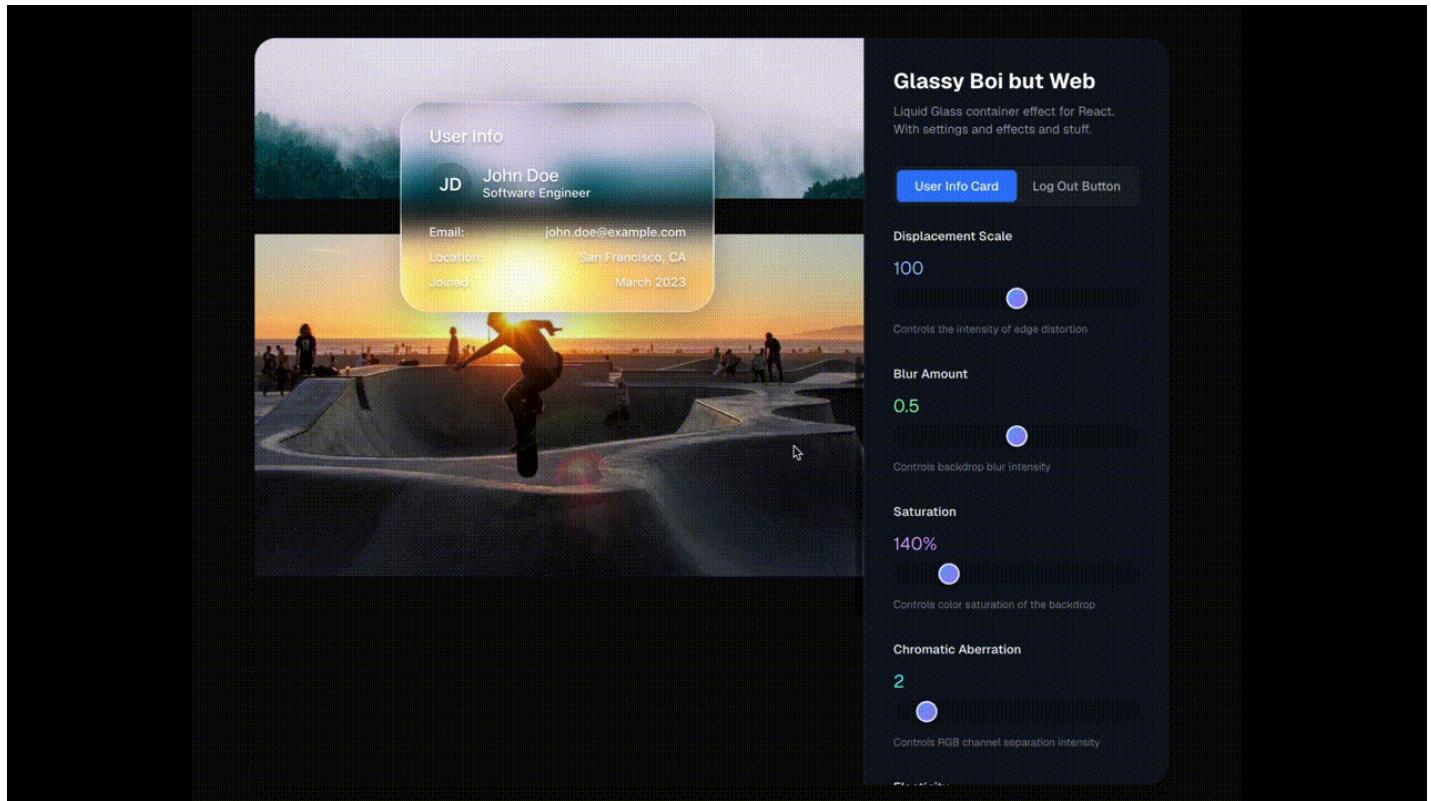
```
blurAmount={0.5}
saturation={140}
aberrationIntensity={2}
elasticity={0.35}
cornerRadius={32}
mode="polar"
overLight={false}

>
<div style={{ padding: '20px' }}>
  Customized Liquid Glass
</div>
</LiquidGlass>
</div>
);

}

export default App;
```

In this example, we've increased the `displacementScale` to create a more pronounced distortion, and we've set the `mode` to `polar` to use the polar displacement map. We've also added a bit of `elasticity` to make the component react to the mouse cursor.



## Conclusion

The `liquid-glass-react` library is a powerful and easy-to-use tool for creating stunning liquid glass effects in your React applications. Whether you're a seasoned designer or a developer looking to add a bit of flair to your projects, this library has something to offer.

With its extensive customization options and advanced features, the `liquid-glass-react` library is the perfect tool for creating the next generation of user interfaces. So what are you waiting for? Give it a try and see what you can create!



Want a great API Testing tool that generates beautiful API Documentation?

Want an integrated, All-in-One platform for your Developer Team to work together with [maximum productivity](#)?

Apidog delivers all your demands, and [replaces Postman](#) at a much more affordable price!

[Sign Up for Free](#)[!\[\]\(6e934896f25e6ce1b0dbb50c23abc197\_img.jpg\) Download Now](#)[For Windows or Linux >](#) Privacy protected Security guaranteed with no ads

## Explore more

**Minimax M1 API**

### How to Run Minimax M1 via API: A Complete Guide

MiniMax M1, developed by a Shanghai-based AI startup, is a groundbreaking open-weight, large-scale hybrid-attention reasoning model. With a 1 million token...

19 June 2025

**Minimax Hailuo 02**

### Minimax Hailuo-02: Better than Google Veo 3?

Explore Hailuo AI's groundbreaking video generation with Hailuo 02, offering unmatched realism and physics simulation. Compare it to Veo 3 and discover wh...

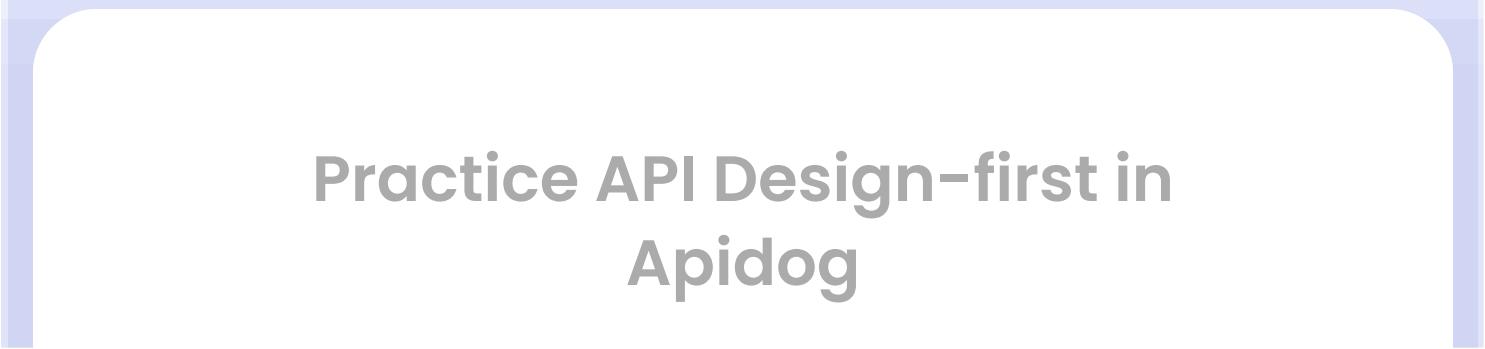
19 June 2025

**Kimi-Dev-72B**

### Is Moonshot AI's Kimi-Dev-72B the Best Coding Model Yet?

Discover Moonshot AI's Kimi-Dev-72B, a state-of-the-art open-source coding LLM with a 60.4% SWE-bench Verified resolve rate. Learn its technical architecture,...

19 June 2025



# Practice API Design-first in Apidog

Discover an easier way to build and use APIs

Sign up for free



All-in-One API Platform: Design, Debug, Mock, Test, and Document

🌐 English ▾



## PRODUCTS

[Download Apidog](#)

[Pricing](#)

[Apidog Europe](#)

[API](#)

[Changelog](#)

[Wall of love](#)

## RESOURCES

[Help Docs](#)

[Getting Started](#)

[Video Tutorials](#)

[Blog](#)

[Newsletter](#)

[Articles](#)

## SUPPORT

[About us](#)

[Request a Demo](#)

[Migration](#)

## COMPARE

[vs Postman](#)

[vs Insomnia](#)

[vs Hoppscotch](#)

Contact us

vs RapidAPI

Term of use

vs Stoplight

Security and compliance

vs Readme

Service status



©2025 Apidog, Inc.