

April 17th, 2026

An open-source email editor you can embed in your own app, plus a new collection of templates.



We're excited to announce [React Email 6.0](#), featuring:

- [Open-Source Editor](#)
- [Build your own Extensions](#)
- [New Templates](#)
- [Unified Package](#)

0:00 / 1:23

Get started today or check the [upgrade instructions](#) below.

```
npm i react-email@latest @react-email/ui@latest
```

React Email now has **2M weekly downloads** on [npm](#), that is a **108% increase** since the last major release 5 months ago.

We would like to thank all the [196 open source contributors](#) who made this possible.

## Open-Source Editor

The centerpiece of React Email 6 is a **new open-source visual editor**, available as a standalone package.

```
npm i @react-email/editor
```

You can embed the editor directly in your app with a few lines of code.

```
import { EmailEditor } from '@react-email/editor'  
  
export default function MyEditor() {  
  return <EmailEditor />  
}
```

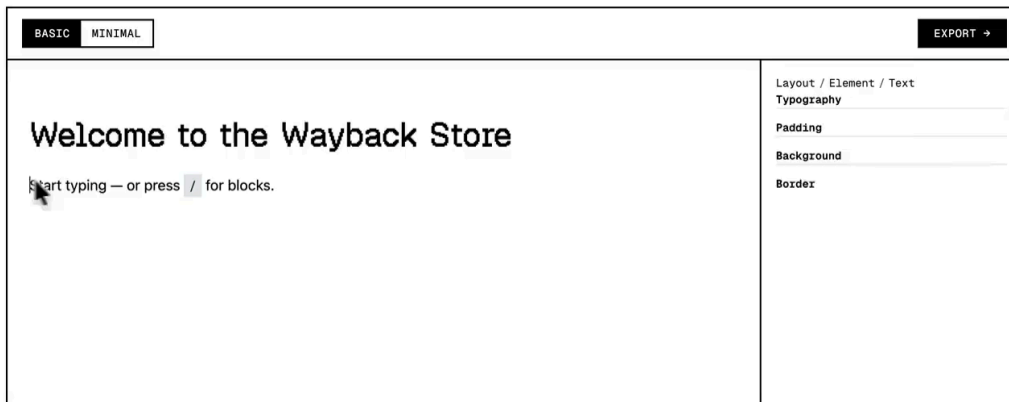
The editor outputs semantically correct, email-ready HTML that renders correctly across every major inbox provider.



TOOLS / EMAIL BUILDER

# Email Builder

Compose email-ready HTML with React  
Email's visual editor.

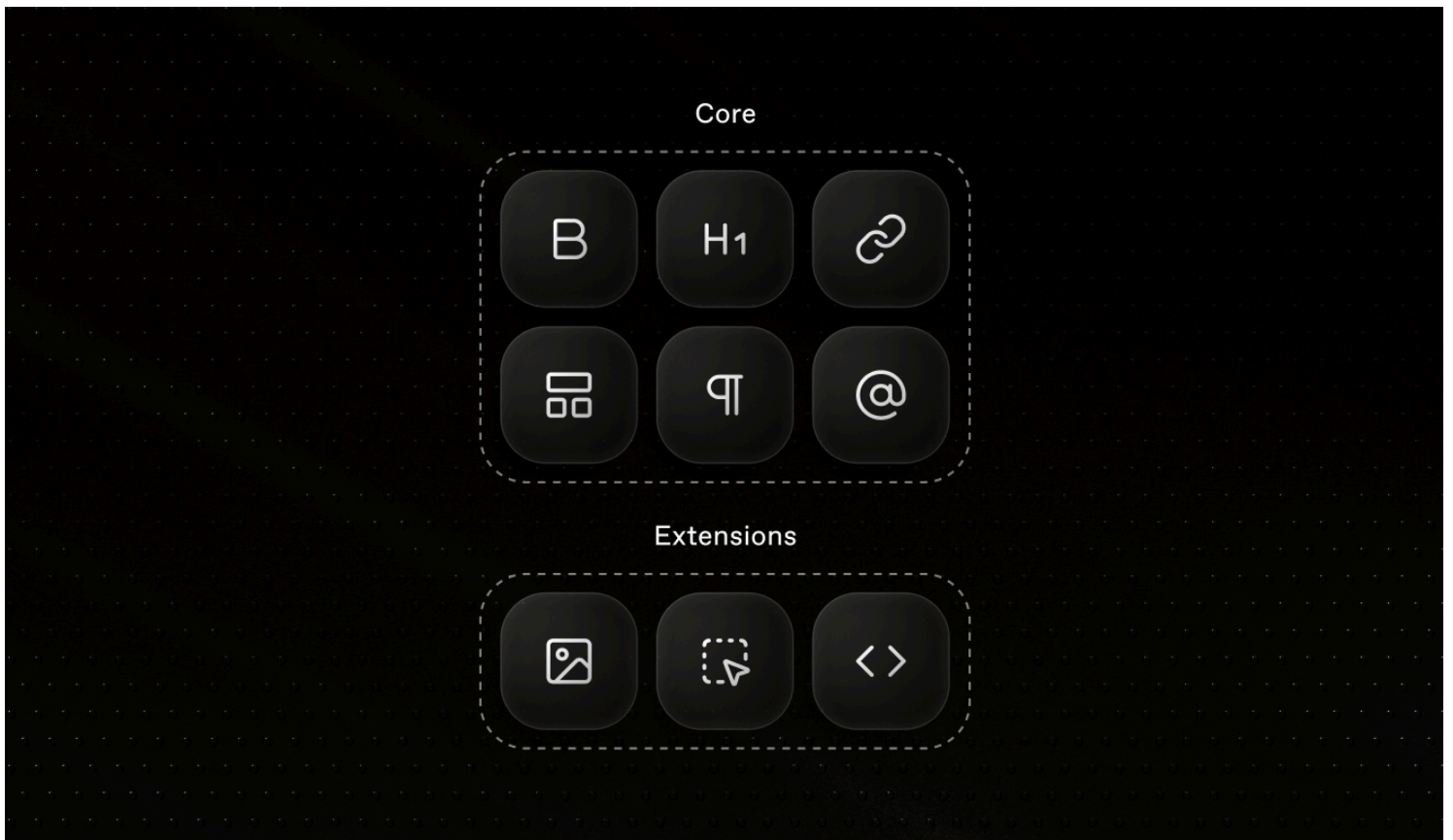


## Editor Architecture

Built on the fundamental learnings from React Email, the editor is designed to be easy to use, extensible, and flexible.

The editor is built in two layers.

1. **Core** works out of the box with no configuration.
2. **Extensions** custom features you can build using the composable API



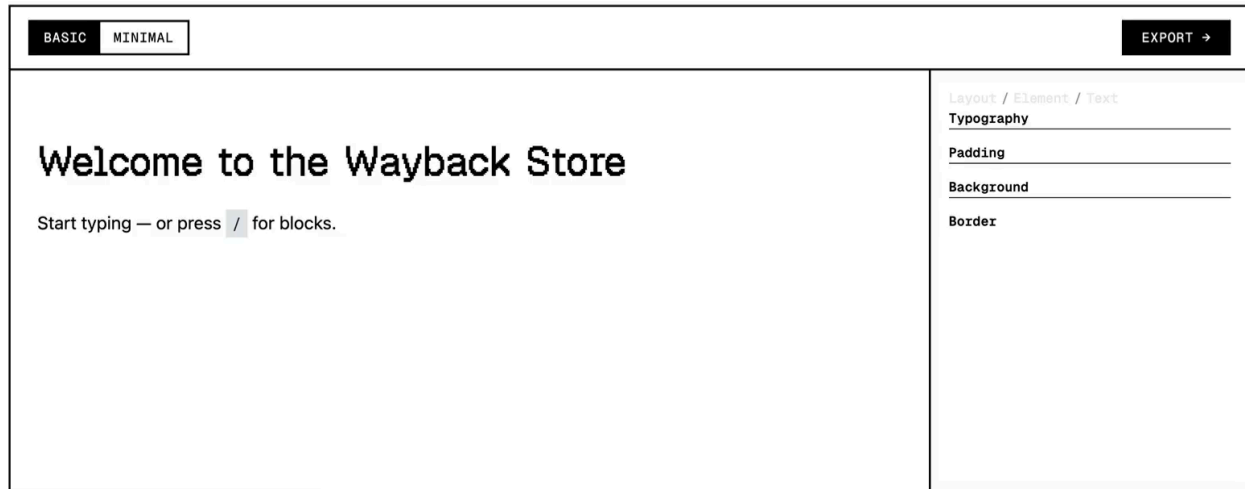
This architecture allows the core to remain simple and focused, while providing a composable API for building your own extensions.

To style the editor, import a default theme to get started quickly.

```
import '@react-email/editor/themes/default.css'
```

You can also build your own theme to style the editor to look and feel like your own app.

Compose email-ready HTML with React  
Email's visual editor.



## Build your own Extensions

The composable API exposes `EmailNode` so you can build any custom block your users need: uploading images to a CDN, embedding social posts, rendering charts inline in an email.

```

export function EmailBuilder() {
  const [mounted, setMounted] = useState(false);
  const [theme, setTheme] = useState<Theme>("basic");
  const [exportResult, setExportResult] = useState<ExportResult | nul

  const editor = useEditor(
    {
      extensions: [[StarterKit, EmailTheming.configure({ theme })]],
      content: STARTER_CONTENT,
      immediatelyRender: false,
    },
    [theme],
  );

  useEffect(() => {
    setMounted(true);
  }, []);
}

```

The opportunities are as endless as your imagination.

Each custom node defines both its HTML representation and its React Email output via `renderToReactEmail`.

```

import { EmailNode } from '@react-email/editor/core';
import { mergeAttributes } from '@tiptap/core';

const Callout = EmailNode.create({
  name: 'callout',
  group: 'block',
  content: 'inline*',

  parseHTML() {
    return [{ tag: 'div[data-callout]' }];
  },

  renderHTML({ HTMLAttributes }) {
    return [
      'div',
      mergeAttributes(HTMLAttributes, { 'data-callout': '' }),
      0,
    ];
  },
});

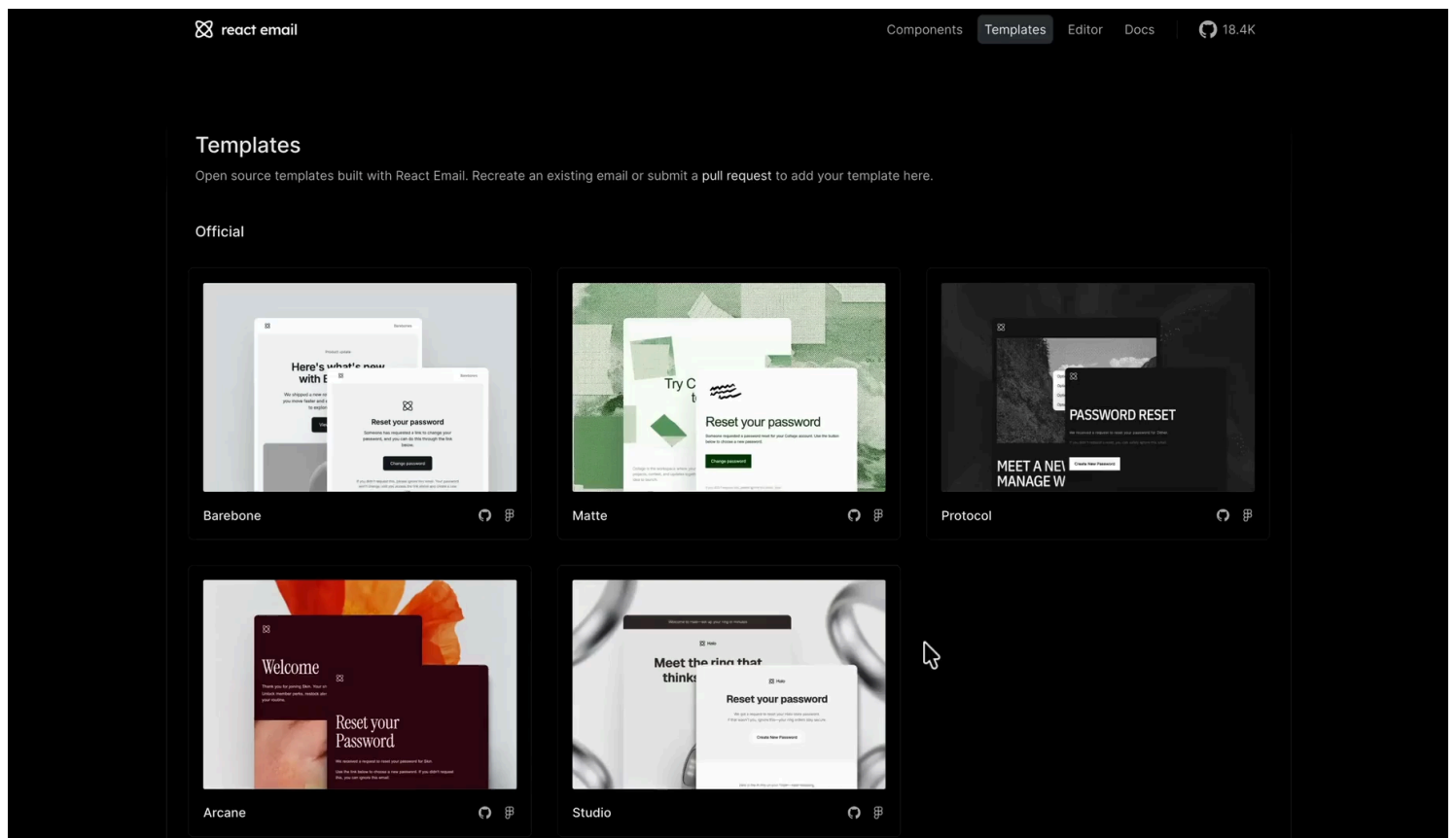
```

```
renderToReactEmail({ children, style }) {  
  return (  
    <div style={{ ...style, padding: '12px 16px', backgroundColor: '#f4f4f5'  
      {children}  
    </div>  
  );  
},  
});
```

For more help, view [editor examples](#) or read the [full editor documentation](#).

## New Templates

React Email 6 also ships with a new collection of templates created by our friends at [Character Studio](#) for common use cases, including authentication flows and e-commerce sequences.



Use them as a starting point, or pull individual sections into your own templates. The templates are available as React Email templates or as Figma files (Authentication

templates, [E-commerce templates](#)).

## Unified Package

To simplify the development experience, we've unified all React Email components into a single package. This means you can now import everything you need from `react-email` without worrying about multiple packages.

```
import { Button, Container, Html, Heading, Tailwind } from 'react-email'
```

This does not include the Editor, which needs to be installed separately.

We've also moved what was `@react-email/preview-server` into what's now `@react-email/ui`.

## Upgrade instructions

Here's how to update from React Email 5.0 to 6.0.

### A) Remove the old packages:

```
npm rm @react-email/components @react-email/preview-server
```

### B) Install the new packages:

```
npm i react-email@latest @react-email/ui@latest
```

### C) Update your imports:

Instead of importing from `@react-email/components`:

```
import { Button, Html, Head, render } from "@react-email/components";
```

You can now import from `react-email`:

```
import { Button, Html, Head, render } from "react-email";
```

For a thorough explanation, see the [updating guide](#).

## Conclusion

We're excited to see the new possibilities that the open-source editor and the new templates will bring to the community.

If you have any questions, feel free to reach out to us on [GitHub](#).



2261 Market Street #5039  
San Francisco, CA 94114



All systems operational

Features

Email API

SMTP

Resources

Changelog

Pricing

Inbound

Security

Automations

SOC 2

Audiences

GDPR

Broadcasts

Brand

Templates

Webhooks

Company

Help

About

Contact

Blog

Support

Careers

Status

Clubs

Migrate

Customers

Knowledge base

Humans

Legal policies

Philosophy

Community

Events

Insiders

Open source

Wallpapers