

About Forge

Get started

- › Introduction to Forge
- › Migration guides
- › Learn

Plan & design

- › App architecture
- › Platform quotas and limits
- ▼ User interface
 - Overview
 - [Build with UI Kit](#)
 - Extend UI with custom options
 - Design tokens and theming
 - Internationalization
 - Understanding the UI modifications module
 - Jira full page modules

- › Legal & privacy
- › App distribution

About UI Kit



You must be on @forge/react major version 10 or higher to use the latest version of UI Kit components.

To upgrade your app to the latest version, run `npm install --save @forge/react@latest` in your terminal.

Upgrading to the latest version may contain [breaking changes](#) for existing UI Kit apps, as *all existing component APIs have been updated*.

UI Kit has enhanced performance and features, such as native rendering and ease of use. With everything happening directly in the browser, your app doesn't need to call a separate server-side function for every state change. With more React features you can now access capabilities directly from React.

Let's look at what UI Kit offers:

Components

Explore [UI Kit components](#), which are reusable building blocks designed to streamline the development of user interfaces.

ON THIS PAGE

[Components](#)

React runtime

API requests

Known limitations and issues

Build

- › Development life cycle
- › App capabilities
 - › Compute
 - › Storage
 - › Remotes
- › User interface
 - UI Kit
 - Frontend bridge
 - Display conditions
 - Reference
- › Observability
- Automation with Forge
- › Trust and security
- › Enterprise development
- › Forge releases and deprecation policy

Manage

- Overview
- › Observability
- › Access
- › Distribution

React runtime

UI Kit runs the app in a React runtime, which enables you to use some React features.

Hooks

To import hooks to your app:

```
import React, { <hook name>} from  
'react';
```

You should be able to use any React hooks that do not rely on an underlying DOM node or depend on any synchronous operations in the browser's event loop. Hooks are now imported from `react`.

Supported hook types

- `useState`
- `useEffect`
- `useContext`
- `useReducer`
- `useCallback`
- `useMemo`
- `useRef`
- `useDebugValue`
- `useDeferredValue`
- `useId`

For more information on Hooks, see the [Hooks API Reference ↗](#) official page.

Context

Forge UI Kit allows you to use the React context in your apps. For more information, see [Context - React ↗](#) documentation.

JSX support

UI Kit supports JSX syntax, enabling developers to write declarative and highly readable code.

Components	JSX support
Components imported from @forge/react	✓
Function components composed of @forge/react components	✓
Context providers	✓
HTML	✗

API requests

Making API requests in UI Kit is the same as in Custom UI. Whether a request is made from the client side or from the lambda depends on the request type.

Atlassian app fetch requests

Atlassian app fetch requests can be made in the frontend through `@forge/bridge`, i.e. in your App.js file. See [requestJira](#), [requestConfluence](#) and [requestBitbucket](#) for documentation.

Non-Atlassian app fetch requests and the storage API

These requests will need to go through the lambda, and so you will need to set up a Resolver.

See [Forge resolver](#).

App context

Retrieve Atlassian app context via the `getContext` method on `@forge/bridge`. See [Forge bridge view](#).

```
1 import { view } from '@forge/bridge';
2 const context = await view.getContext()
```

Known limitations and issues

Limitations

- You won't have access to any of the underlying DOM, so features that depend on that will not work. This includes portals and forwarding refs. You also still cannot use arbitrary HTML, and are restricted to using the components exported from `@forge/react`.
- UI Kit is not supported in [Forge workflow condition](#), [Forge workflow validator](#), and [Forge workflow post function](#) modules.

Issues

- Using `TypeScript` for resolver files causes deployment failures. You will get an error like **TS2304: Cannot find the name "Forge UI"** when doing a `forge deploy`. You can work around this by adding a rule into the `tsconfig.json` file to only include the `src` file.

