

Component Library

- [Running & Testing in the Component Library Sandbox](#)
- [Things we want document](#)
- [Nautilus UI](#)

Running & Testing in the Component Library Sandbox

Initial Component Library Setup

- Pull down and open the Atlantis-Components-Library repo
- Get authenticated to Font Awesome globally by inputting these commands into your command line:
 - `npm config set "@fontawesome:registry" https://npm.fontawesome.com/`
 - `npm config set "//npm.fontawesome.com/:_authToken" XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX`
 - Replace the placeholder with the real auth token, which you can find in the Shared Development Last Pass account under Font Awesome, or by asking someone with access to it.

Making Component Changes and Testing

- The actual component definitions are in the `efc-atlantis-components` directory. You can make changes there, and then see them reflected in the sandbox by using Rollup. To do that, navigate to the `efc-atlantis-components` directory in your command line and run:
 - `npm install` or `npm i` (On first time running)
 - `npm run build`
 - Behind the scenes, this is a script running `rollup -c` and it is packaging everything up for the sandbox to reference
- If you rerun the sandbox after packaging your changes, you should see them reflected there
- *Please note that we have seen regular inconsistencies after compiling so you may need to rebuild and/or restart the sandbox a few times to see your changes. This is a factor in our decision to move to the Nautilus UI project, along with a long wait time for builds and refreshing the sandbox.

Running the Sandbox

- Make sure your packages are up to date and then run the app via Vite. To do that, navigate in your command line to the `revver-comp-sandbox` directory and input these commands:
 - `npm install` or `npm i` (On first time running)
 - `npm run dev`

- After a few seconds the app should be running, with a localhost port you can visit: <http://localhost:5173/>

Related Docs

- Bootstrap: [Bootstrap · The most popular HTML, CSS, and JS library in the world. \(getbootstrap.com\)](#)
- Vanilla Extract: [vanilla-extract — Zero-runtime Stylesheets-in-TypeScript.](#)
- Reactstrap: [Home/Installation - Page · Reactstrap](#)
- React-Virtuoso (used in the VirtualizedFullTable): [Getting Started with React Virtuoso | React Virtuoso](#)

Things we want document

- Vanilla Extract
- Bootstrap strategy (bootstrap variables and overriding style)
- what specific styles (like colors and fonts) should exclusively be managed in component library

Nautilus UI

Nautilus UI

- Nautilus UI is located in the Atlantis-Component-Library repo alongside efc-atlantis-components and revver-comp-sandbox and is the location for new components (pending POC completion).

Initial Setup

- Pull down and open the Atlantis-Components-Library repo
- Get authenticated to Font Awesome globally by inputting these commands into your command line:
 - `npm config set "@fortawesome:registry" https://npm.fontawesome.com/`
 - `npm config set "//npm.fontawesome.com/:_authToken" XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX`
 - Replace the placeholder with the real auth token, which you can find in the Shared Development Last Pass account under Font Awesome, or by asking someone with access to it.

Adding a component

- Add a folder with the component name under the "src" folder
- Add the component, story, index, and component specific styling files in the same folder.
 - Example:
 - Button
 - index.ts
 - Button.tsx
 - Button.stories.ts
 - button.scss
- In the component level index.ts file export the component
 - `export { Button } from './Button';`
- In the library level index.ts file add the component export as well
 - `export * from './Button';`

Run Storybook

- Make sure you're at the level where you can see the ".storybook" folder and the various config files

-Atlantis-Component-Library\nautilus-ui
- In your command line simple run:

```
npm run storybook
```

- This will automatically spin up the project Storybook on <http://localhost:6006/>
- Here's the documentation for more details on Storybook. [Get started with Storybook • Storybook docs](#)

Build the library

```
npm run build
```

Build the NPM package

Once the PR is approved and merged into main the pipeline will automatically update the NPM package. (This is still a work in progress)

Team Process for Moving from EFC Component Library to Nautilus UI

After the POC is complete this section will get updated, but for now:

- A list of candidate components will be created that anyone can work on moving over during ILs or as a possible sand item.
 - Items from the EFC Component Library
 - Items that were created in Atlantis-FrontEnd while waiting on Nautilus UI that are intended for the library.
- Any new multi-use components should go directly into Nautilus UI going forward.