rush

Visualizing Rush projects using rush-plugin-skott

Rush Hour (West) - Feb 2023 - Antoine Coulon

What it is `rush-plugin-skott`

`rush graph`

It's a third-party plugin allowing to visualize a Rush monorepo graph.

Part of krush (https://github.com/antoine-coulon/krush), which aims to be include a set of Rush plugins, one at a time.

The plugin mainly includes one command now which is `rush graph`.

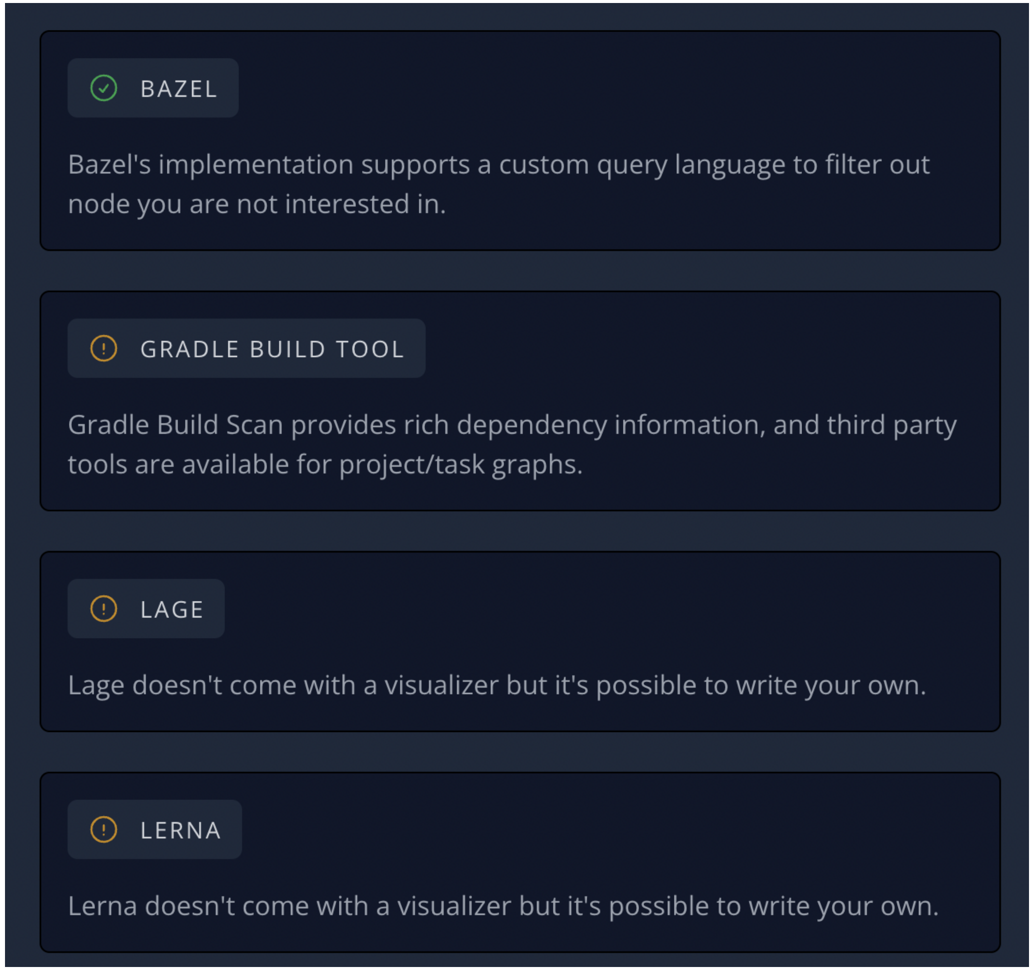
By starting this command, the plugin automatically generates the graph and opens an interactive webapp.

Benefits of having a way to visualize the monorepo graph

- Having a big picture of the architecture of the monorepo, showing any kind of metadata that could be useful (e.g. third-party dependencies, bundle sizes)
- Facilitate the overview about projects dependencies, which projects depend on which other projects to optimize the toolchain. At scale, unwanted/implicit dependencies can be non trivial to track

State of the monorepo ecosystem 1/2

Source: https://monorepo.tools/



✓ BAZEL

Bazel's implementation supports a custom query language to filter out node you are not interested in.

! GRADLE BUILD TOOL

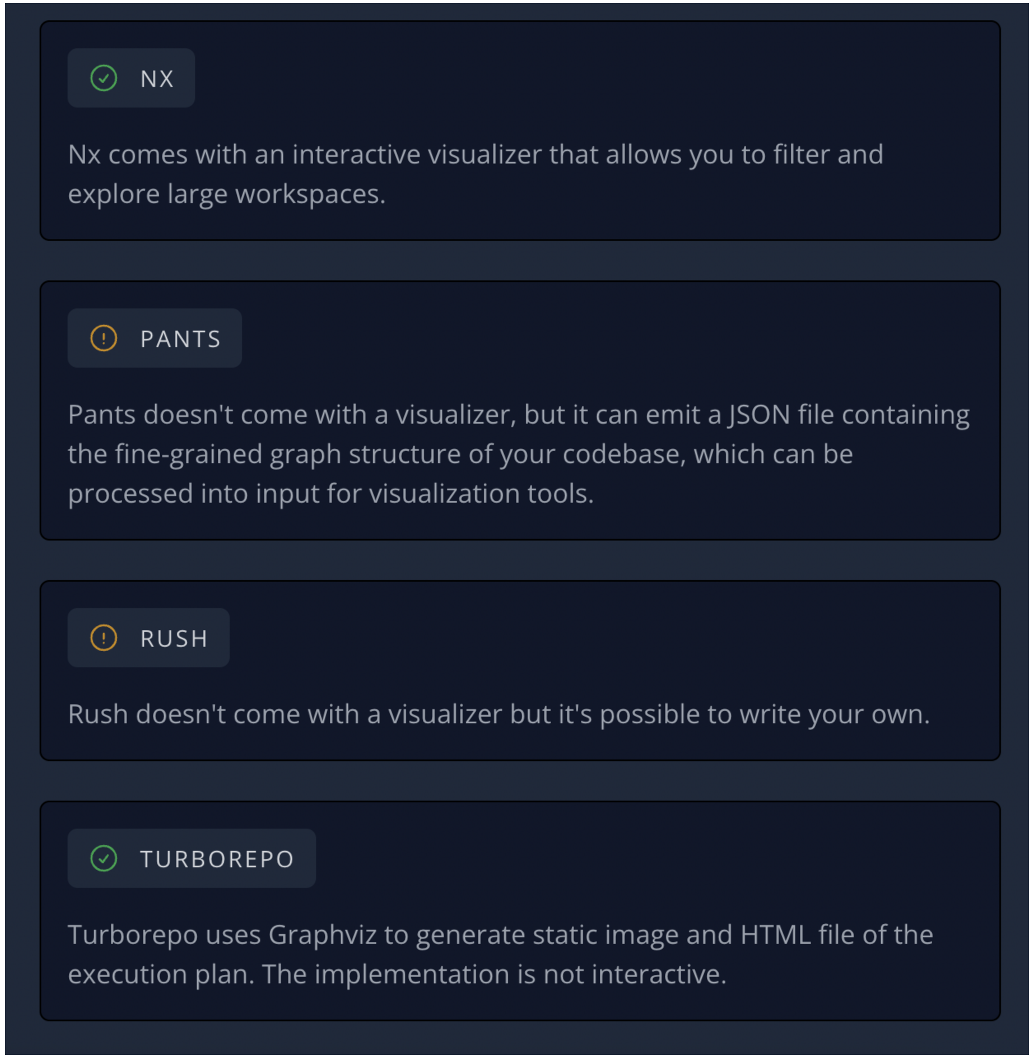
Gradle Build Scan provides rich dependency information, and third party tools are available for project/task graphs.

! LAGE

Lage doesn't come with a visualizer but it's possible to write your own.

! LERNA

Lerna doesn't come with a visualizer but it's possible to write your own.



✓ NX

Nx comes with an interactive visualizer that allows you to filter and explore large workspaces.

! PANTS

Pants doesn't come with a visualizer, but it can emit a JSON file containing the fine-grained graph structure of your codebase, which can be processed into input for visualization tools.

! RUSH

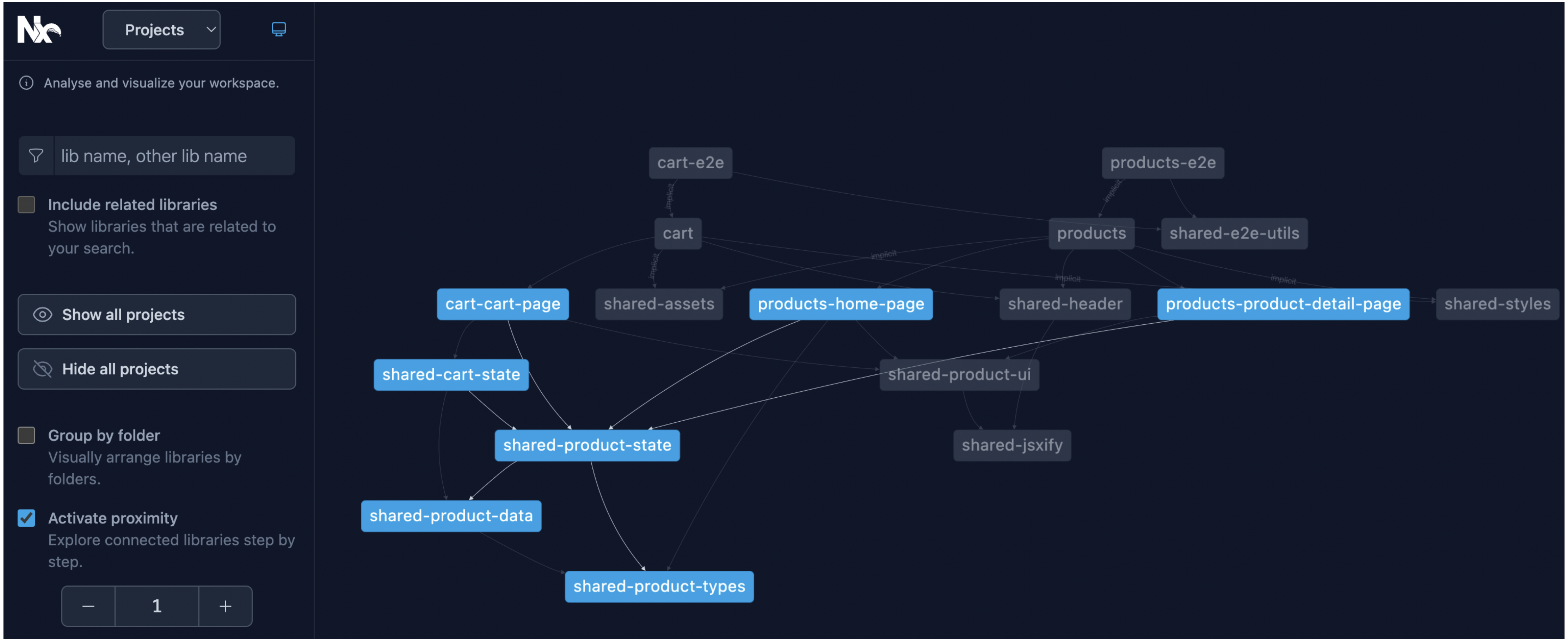
Rush doesn't come with a visualizer but it's possible to write your own.

✓ TURBOREPO

Turborepo uses Graphviz to generate static image and HTML file of the execution plan. The implementation is not interactive.

State of the monorepo ecosystem

Nx has a builtin `graph` command that does this.



Analyse and visualize your workspace.

lib name, other lib name

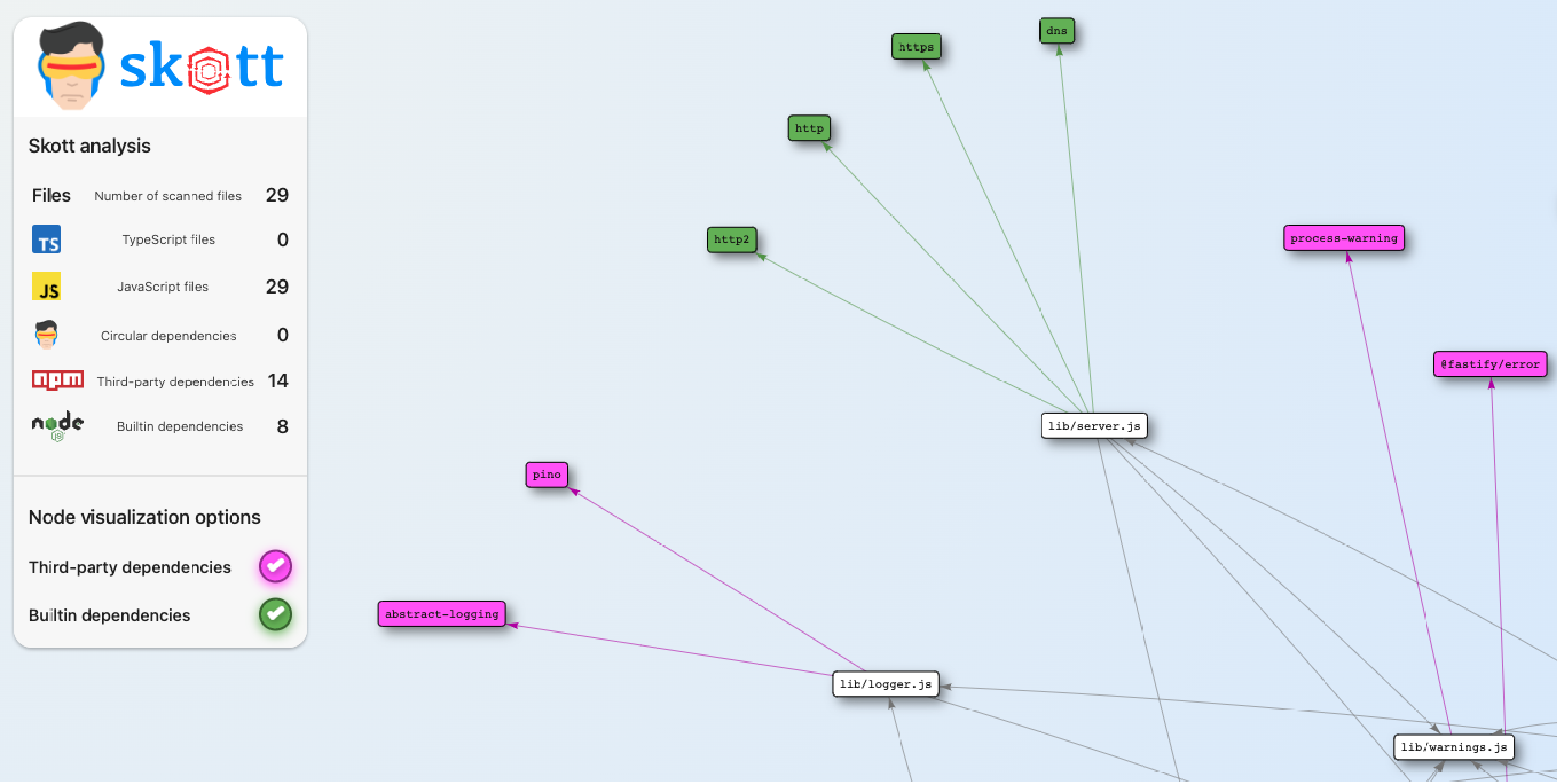
□ Include related libraries
Show libraries that are related to your search.

□ Group by folder
Visually arrange libraries by folders.

☑ Activate proximity
Explore connected libraries step by step.

– 1 +

Introducing Skott, a tiny tool analyzing JavaScript/TypeScript source files to build graphs.



skott

Skott analysis

Files	Number of scanned files	29
TS	TypeScript files	0
JS	JavaScript files	29
	Circular dependencies	0
npm	Third-party dependencies	14
node	Builtin dependencies	8

Node visualization options

Third-party dependencies ✓

Builtin dependencies ✓

Starting a Skott plugin for Rush (PoC)

- Few weeks old (not stable)
- Embeds a custom dependency resolver for Rush (using the `@rushstack/rush-sdk`) *~ 200 LOC*
- Uses the `skott-webapp` package (which is the default webapp for Skott)
- Mainly relies on JavaScript/TypeScript *source files analysis* (detecting import statements) at the moment. It will also either use the internal Rush graph or use some package.json analysis to find workspace dependencies as well to reproduce the Rush behavior (and to ensure the same graph is generated for both the visualization and all the things that rely on the internal graph such as `incremental` commands).

Quick demo using a tiny Rush monorepo

A little showcase on a real-world monorepo

What could/should be improved/implemented 1/2

- Dependencies should be created by looking at the workspace dependencies either via package.json or the internal Rush graph API. We could also imagine to get more information about where the dependencies are used by running Skott analysis so that we also know if the dependency is used and if it is know precisely where.

Command

- Support the Rush project selectors (i.e. - --from, --to, --only, --to-except) suggested by **@iclanton**
- Watch mode
- Support other basic arguments: http settings (port, host, auto browser open...)

What could/should be improved/implemented 2/2

Web application

- The Skott webapp could be swapped or made fully customizable/adaptable for Rush needs (UI theme, logos, etc). At the moment, it was designed to fit Skott's needs but could evolve and become flexible
- Graph layout could be adapted to monorepos as they tend to have a flatter structure (file-level graph is a mess to represent)
- Add filters (by tags, by names, etc)
- Display metadata for each project (bundle size, associated scripts...)
- Interactive search
- ... many more

State of the plugin

- Early stage, any feedback/idea is welcome
- Implement collected feedbacks and suggestions in the third-party plugin for now (including Skott aswell if it needs to be extended)
- Ensure that the plugin is consistent with the graph creation, by also using internal Rush graph generated instead of only an external static analysis tool (i.e. `skott`)
- If the plugin ends up fitting well with the expectations, I'd be happy to land PRs to add that as a builtin command

rush

Thanks for listening

Feature proposal issue on the Rush monorepo (#3945):

https://github.com/microsoft/rushstack/issues/3945

`Krush` monorepo (including `rush-plugin-skott` package location):

https://github.com/antoine-coulon/krush

`Skott` monorepo (`skott` + `skott-webapp` location):

https://github.com/antoine-coulon/skott

`Demo sample` https://github.com/antoine-coulon/rush-skott-demo