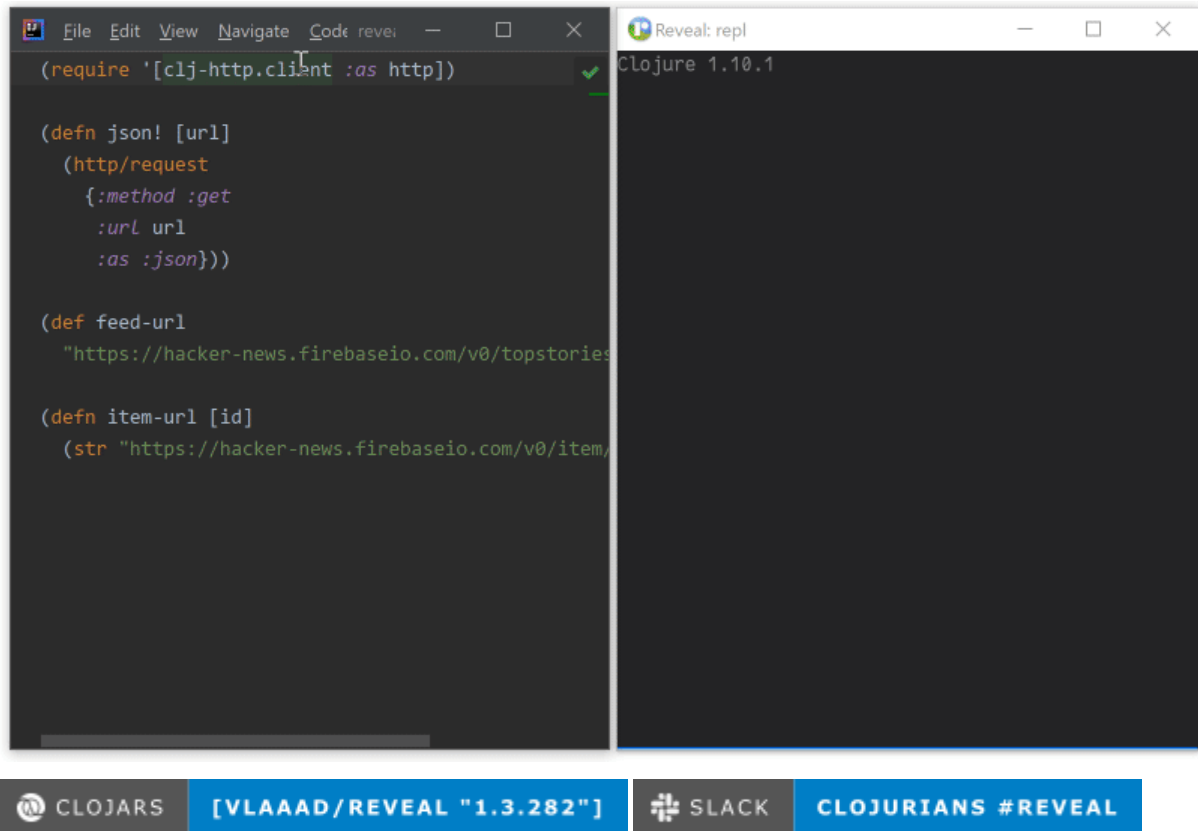




Reveal: Read Eval Visualize Loop for Clojure



Rationale

Repl is a great window into a running program, but the textual nature of its output limits developer's ability to inspect the program: a text is not an object, and we are dealing with objects in the VM.

Reveal aims to solve this problem by creating an in-process repl output pane that makes inspecting values as easy as selecting an interesting datum. It recognizes the value of text as a universal interface, that's why its output looks like a text: you can select it, copy it, save it into a file. Unlike text, reveal output holds references to printed values, making inspecting selected value a matter of opening a context menu.

Unlike datafy/nav based tools, Reveal does not enforce a particular data representation for any given object, making it an open set — that includes datafy/nav as one of the available options. It does not use datafy/nav by default because in the absence of inter-process communication to datafy is to lose.

Not being limited to text, Reveal uses judicious syntax highlighting to aid in differentiating various objects: text `java.lang.Integer` looks differently depending on whether it was produced from a symbol or a class.

Reveal Pro

Reveal aims to be an extensible tool suitable for helping with development of any Clojure program. Reveal Pro provides a set of extensions that improve developer experience by providing more tools, so you can focus on your problem with data and knowledge you need, available as soon as you need it.

Documentation

You can find overview, setup instructions and more at vlaaad.github.io/reveal.

Versioning, stability, public and internal code

Reveal uses `1.MAJOR.REVISION` versioning where: - `1` is a static prefix for compatibility with semantic versioning. Reveal should never introduce breaking changes, so if an update broke something, please file a bug report; - `MAJOR` is a number that is incremented when there are significant changes or improvements to Reveal; - `REVISION` is a commit number from the beginning of a history.

Reveal's compatibility promise applies to its public API: `vlaaad.reveal` ns, everything else is implementation detail that is subject to change. Reveal's UI and controls might change in the future.