

REPL Driven Development

Contents

1. [With Leiningen](#)
2. [Starting the Application from the REPL](#)
3. [Connecting to the nREPL](#)
4. [With Boot](#)
5. [Starting the Application from the REPL](#)

A typical workflow for developing Clojure applications is to connect the editor to a REPL running an instance of the application. Luminus provides two ways to connect to the REPL.

With Leiningen

Starting the Application from the REPL

When you run the REPL in the project directory it will start in the `user` namespace located in the `env/dev/clj/user.clj` file. This namespace provides helper functions `start`, `stop`, and `restart` that are used to manage the state of the application components.

To start the HTTP server and any other components such as databases, run the `start` function:

```
(start)
```

Run the following command to start the HTTP server:

```
(mount/start #'<app>.core/http-server)
```

Connecting to the nREPL

Luminus also provides an embedded [nREPL](#) that can be used to connect the editor to a running instance of the server. A default nREPL port is set the `env/dev/resources/config.edn` configuration file:

Build Tool:

Topics

[Your First Application](#)

- **REPL Driven Developm**

[Application Profiles](#)

[HTML Templating](#)

[Static Assets](#)

[ClojureScript](#)

[Routing](#)

[RESTful Services](#)

[Request types](#)

[Response types](#)

[Websockets](#)

[Middleware](#)

[Sessions and Cookies](#)

[Input Validation](#)

[Security](#)

[Component Lifecycle](#)

[Database Access](#)

[Database Migrations](#)

[Logging](#)

[Internationalization](#)

[Testing](#)

[Server Tuning](#)

[Environment Variables](#)

[Deployment](#)

[Useful Libraries](#)

[Sample Applications](#)

[Upgrading](#)

[Clojure Resources](#)

Books

```
{:dev true
 :port 3000
 ;; when :nrepl-port is set the application starts the nREPL server on
 :nrepl-port 7000}
```

When you run your application using `lein run` it will create a network REPL on the port `7000` and you will be able to connect your editor to it on `localhost:7000`. Note that when the `NREPL_PORT` environment variable is set then nREPL will also be available in production and can be used to inspect the application the same way you would in development.

Please see the [deployment section](#) for the instructions on enabling the nREPL for production.

With Boot

Starting the Application from the REPL

With Boot, running the HTTP server in the REPL requires a few extra steps. In the command line, run `boot dev repl` to start the REPL in the "dev" configuration. Then, in the REPL, run `(in-ns 'user)` followed by `(load "user")` to load the helper functions found in `env/dev/clj/user.clj`. Since you are already in the `user` namespace, you have immediate access to these functions. Now you can run `(start)` to run your application in the REPL.