

# Clojure

sachin

2013-09-26 Thu

## Contents

<b>1</b>	<b>lein</b>	<b>2</b>
1.1	Lein configuration . . . . .	2
1.2	Upgrade . . . . .	2
<b>2</b>	<b>project/app</b>	<b>2</b>
2.1	create . . . . .	2
2.2	REPL . . . . .	3
<b>3</b>	<b>library</b>	<b>3</b>
3.1	create . . . . .	3
3.2	REPL . . . . .	4
<b>4</b>	<b>dependencies</b>	<b>4</b>
<b>5</b>	<b>compiling</b>	<b>5</b>
<b>6</b>	<b>running</b>	<b>5</b>
<b>7</b>	<b>connecting to REPL server</b>	<b>5</b>
<b>8</b>	<b>generating docs from libraries</b>	<b>5</b>
<b>9</b>	<b>notes</b>	<b>6</b>
9.1	Difference between :use and :require . . . . .	6
<b>10</b>	<b>references</b>	<b>6</b>

## 1 lein

- Download **lein** binary file from github and copy it to your ~/bin/ directory.
- Add ~/bin PATH in your ~/.bashrc
- Make the binary executable

```
chmod a+x ~/bin/lein
```

- Now run the self installer

```
lein
```

### 1.1 Lein configuration

- lein version 2 uses the idea of *profiles*. The location of file is ~/.lein/profiles.clj
- sample profiles.clj file

```
1:  {:user {:plugins [[lein-difftest "1.3.8"]
2:                                [lein-marginalia "0.7.1"]
3:                                [lein-pprint "1.1.1"]
4:                                [lein-swank "1.4.5"]
5:                                [lein-exec "0.3.0"]
6:                                [lein-ring "0.8.7"]]]}}
```

### 1.2 Upgrade

- Upgrade lein to latest stable version using

```
lein upgrade
```

## 2 project/app

### 2.1 create

- create a project using

```
lein new my-project
```

- create an app using

```
lein new app my-stuff
```

- create a noir project(**noir** is a plugin web framework)

```
lein new noir my-web-project
```

## 2.2 REPL

- Go to your project directory and run **lein**

```
cd my-stuff  
lein repl
```

- inside **repl** prompt, run **(-main)**

```
my-stuff.core=> (-main)
```

- If you have done some change in a code, reload it

```
my-stuff.core=> (require 'my-stuff.core :reload)
```

- run a program using **lein**

```
lein run
```

- if you are happy with your app, package it into an executable **jar**

```
lein uberjar
```

- and run **jar** file using

```
java -jar target/my-stuff-0.1.0-standalone.jar
```

## 3 library

### 3.1 create

```
lein new default my-lib
```

## 3.2 REPL

```
$ lein repl
...

user=> (require 'my-lib.core)
nil
user=> (ns my-lib.core)
nil
my-lib.core=> (my-func 3)
9
```

## 4 dependencies

- add project description to ~/.lein/profiles.clj or app/project.clj

```
1:  {:user {:plugins [[lein-difftest "1.3.8"]
2:                                [lein-marginalia "0.7.1"]
3:                                [lein-pprint "1.1.1"]
4:                                [lein-swank "1.4.5"]
5:                                [lein-exec "0.3.0"]
6:                                [lein-ring "0.8.7"]]]}}
```

- or you can have it specific to the project

```
1:  (defproject perfect-clojure "0.1.0-SNAPSHOT"
2:    :description "A simple clojure app to test my environment"
3:    :url "http://clojuremadesimple.co.uk"
4:    :license {:name "Eclipse Public License"
5:              :url "http://google.co.uk"}
6:    :dependencies [[org.clojure/clojure "1.3.0"]]
7:    :dev-dependencies [[midje "1.4.0"]
8:                       [autodoc "0.9.0"]
9:
10:   :plugins [[lein-swank "1.4.4"]]]
11:  )
```

- set or unset proxy and run

```
lein deps
```

## 5 compiling

- Build an executable jar

```
lein deps
lein uberjar
```

## 6 running

```
java -jar target/my-stuff-0.1.0-SNAPSHOT-standalone.jar
```

## 7 connecting to REPL server

- A new REPL server is started at certain PORT when we invoke

```
lein repl
```

- We can connect to existing server using

```
lein repl :connect nrepl://localhost:PORT
```

## 8 generating docs from libraries

- install marginalia

```
cd ~/.lein
touch profiles.clj
```

- add following line to profiles.clj (marginalia version may be different)

```
{:user {:plugins [[lein-marginalia "0.7.1"]]]}}
```

- then, in your project

```
cd /path/to/project/
```

- generate docs

```
lein marg
```

- browse docs in path using web-browser

`file:///path/to/my-proj/docs/uberdoc.html`

## 9 notes

### 9.1 Difference between `:use` and `:require`

```
(ns whatever
  (:use [some.namespace :only [vars you want]]))
```

```
(ns whatever
  (:require [some.namespace :as sn]))
```

## 10 references

- <http://www.unexpected-vortices.com/clojure/brief-beginners-guide/>
- <http://clojuremadesimple.co.uk/>