

# Structuring Code in Clojure

---



**Zachary Bennett**

Software Engineer

@z\_bennett\_ zachbennettcodes.com





Used to opinionated frameworks?

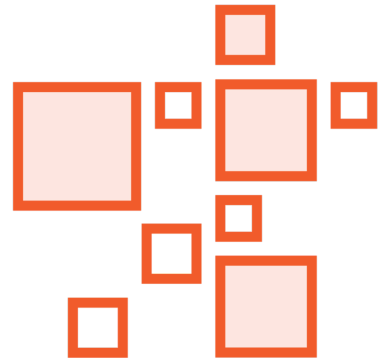
**Clojure can be disorienting if you're used to frameworks like Ruby on Rails or Java's Spring**



Keep it simple



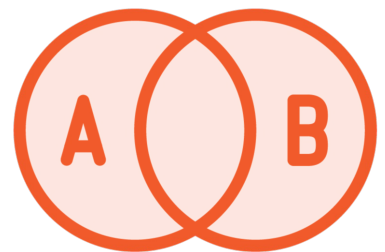
# Clojure Project Principles



**Simple, logical namespaces**



**If your project grows, use a structuring tool**



**Keep your directory structure as simple as possible**



# Namespaces

---



# Namespace

**You can think of a namespace in Clojure like a shelf in a library. But instead of containing books, namespaces are like a loose collection of symbols!**



`:: Creating namespaces!`

`; Create/declare the current namespace  
(ns wired-brain-coffee.utilities)`

- ◀ **The “ns” macro creates a new namespace and then uses it**
- ◀ **The part of the namespace name before the “.” denotes the directory that the corresponding file resides within.**
- ◀ **Dashes in the namespace == underscores in the actual directory name!**
- ◀ **The part of the namespace name after the “.” denotes the file name i.e. “utilities.clj”**

Make sure your directories/files  
match up with the right namespace  
name!





```
;; Using namespaces!
```

```
; Bring in another namespace and alias it  
(require '[clojure.set :as set])
```

```
; Referencing an aliased namespace  
(set/union set1 set2)
```

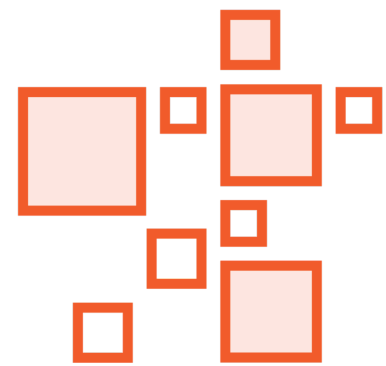
- ◀ **Using “require” allows you to load a library or libraries from a given namespace**
- ◀ **The :as keyword allows you to alias the imported namespace**
  
- ◀ **Functions within the aliased namespace can then be used accordingly**

# Setting Up a Clojure Project With Leiningen

---



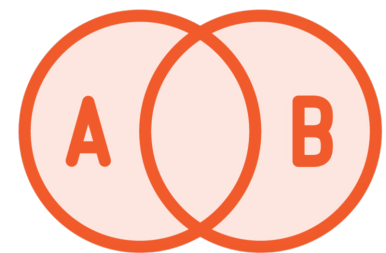
# Why Leiningen?



**Easily setup a new Clojure project with a structure**



**Bring in third-party dependencies**



**Automated/declarative project deployment**



# Leiningen Basics

## Projects

A collection of  
structured Clojure  
files

## Tasks

Commands you can  
run to create or  
manage a project

## Templates

Project structures you  
can pick from when  
creating a new project



# Demo



## Setting up a Clojure project

- Use Leiningen to bootstrap a new app
- Look at the generated namespaces



# Package Management

---



# Package Management

**The process of handling project dependencies that are available either locally or via a remote repository.**



Dependencies are hard...  
Leiningen helps!





```
52      :dependencies [[org.clojure/clojure "1.3.0"]  
53                      [org.jclouds/jclouds "1.0" :classifier "jdk15"]  
54                      [net.sf.ehcache/ehcache "2.3.1" :extension "pom"]
```

```
104     :repositories ["java.net" "https://download.java.net/maven/2"]  
105                  ["sonatype" {:url "https://oss.sonatype.org/content/repositories/releases"
```

## Declarative Dependency Management

**Leiningen makes package management quite easy via the declarations available in the project.clj file that is located in every Leiningen Clojure project.**

# Demo



## Using Third-Party Dependencies

- Alter the project.clj file
- Use our new dependency

