

Clojure in practice

```
(loop [software current-software]
  (recur (-> software get-feedback build ship)))
```

About us

→ **Christian Tomoscheit**

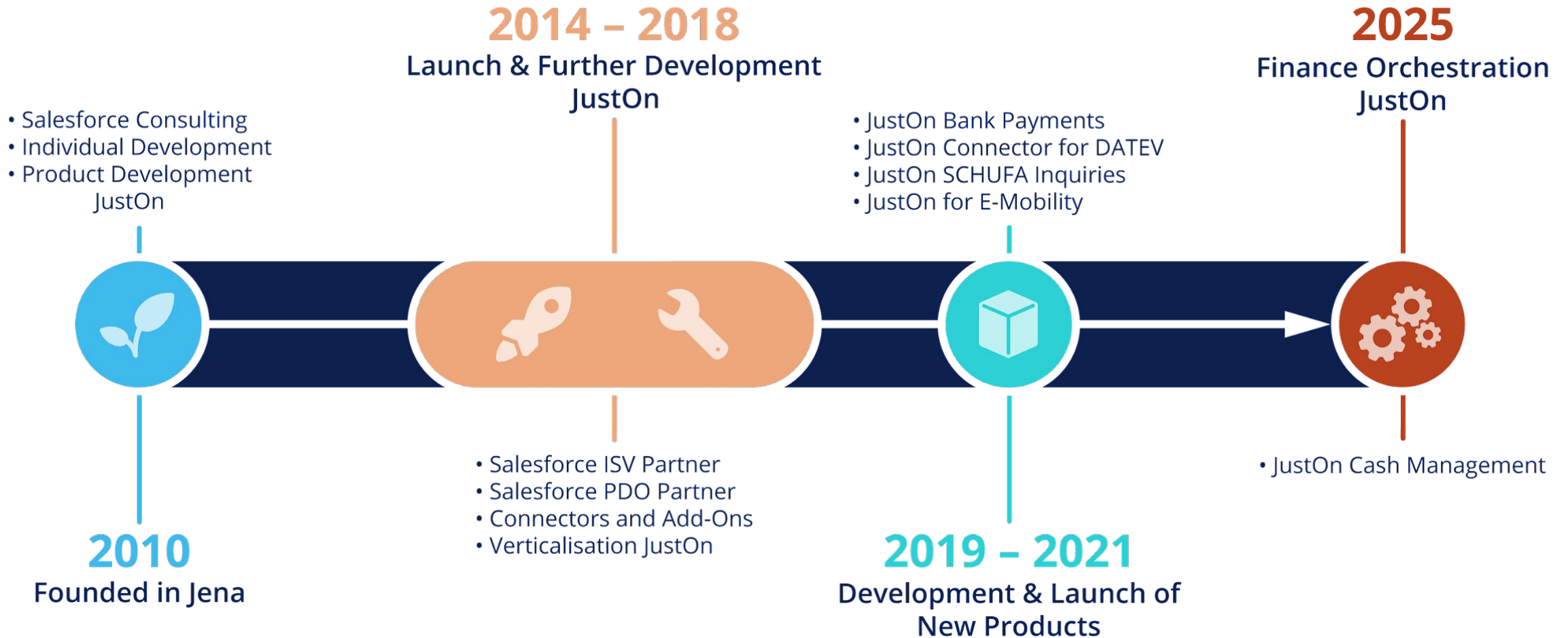
- ◆ Software Developer at JustOn

→ **Alexander Johannes**

- ◆ Head of Development at JustOn



Company history



Our software automates finance processes



Credit Check



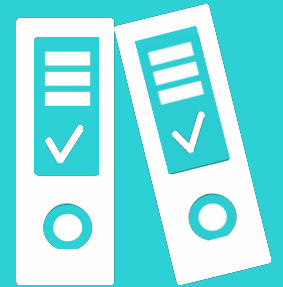
Billing & Invoice
Management



Cash
Management



Dunning &
Collection



Accounting

Why did we chose Clojure for our business?



&



Why did we chose Clojure for our business?

- Gained experience with in-house tools and small projects
- Good fit for our business domain
- Easy to attract qualified and interested people
- Job satisfaction

```
(ns juston.modules.ebics.h005.transaction.upload-xml
  (:require [juston.modules.ebics.h005.transaction.xml-utils :as utils.xml]
            [juston.modules.ebics.xml-validation :as xml-validation]
            [juston.modules.ebics.schema :as ebics.schema]
            [juston.modules.ebics.h005.transaction.orders :as orders]
            [juston.modules.ebics.crypto.defs :as crypto-defs]
            [clojure.data.xml :as xml]
            [buddy.core.codecs :as codecs]
            [buddy.core.hash :as hash]
            [clj-http.util :refer [deflate]]
            [juston.lib.crypto.utils :as utils]
            [juston.lib.crypto.aes-utils :as aes]
            [juston.lib.crypto.rsa-utils :as rsa]
            [juston.common.schema.utils :refer [validate]]
            [clojure.string :as str]
            [juston.base.config :refer [env]]
            [juston.utils.logging :as log]))
```



Why Clojure as Language and Technology?

→ the deal for us:

- ◆ Hosted on the JVM
- ◆ Rapid prototyping
- ◆ Great stability & backwards compatibility
- ◆ Functional programming
- ◆ Same language for frontend and backend



Getting fluent in Clojure

- Most common used functions/macros (more than 50 times):
 - ◆ **functions:** def, defn, fn, let
 - ◆ **thread macros:** ->, ->>
 - ◆ **conditionals:** if, when, cond
 - ◆ **strings:** str, string?
 - ◆ **collections:** assoc, assoc-in, dissoc, conj, update, get, get-in, merge, into
 - ◆ **collections:** map, reduce, filter
 - ◆ **logical:** or, and, not
 - ◆ **exceptions:** try, throw, ex-info
 - ◆ **misc:** nil?, some?, count, first, vals, do
 - ◆ **calling function:** partial, apply



Clojure hands on

```
Connected to the target VM, address: '127.0.0.1:52704', transport: 'socket'
```

```
Connecting to local nREPL server...
```

```
Clojure 1.11.1
```

```
nREPL server started on port 52706 on host localhost - nrepl://localhost:52706
```

```
Loading src/juston/example1.clj... 2022-11-03 13:23:53
```

```
.275:INFO::nREPL-session-d863ae2e-65d6-4706-8d6e-071e5ce9709b: Logging initialized @11672ms to  
org.eclipse.jetty.util.log.StdErrLog
```

```
done
```

```
⇒ {:total-net 150.00M, :total-tax 28.50M, :grand-total 178.50M}
```

- Part 1: Clojure basics
- Part 2: Little App Example
- available at <https://bitbucket.org/JustOn/clojure-talk>



Most loved and used Clojure libraries

- **How we started:** Luminus Template
- **buddy:** security libraries
- **cheshire:** JSON serialization & deserialization
- **clj-http:** client for HTTP requests
- **hugsql:** SQL files to define database functions
- **malli:** Data-driven Schemas
- **reitit:** data-driven router
- **mount:** manage application state
- **ring:** web applications library
- **re-frame:** cljs framework for building user interfaces



Going big - lessons learned

- Building infrastructure is hard
 - ◆ software localization (fluent)
 - ◆ running background tasks (job framework)
 - ◆ logging
 - ◆ interfacing Salesforce



Going big - lessons learned

- Maintainable code is still a lot of work
 - ◆ documentation
 - ◆ code structure
 - ◆ schemas
 - ◆ tooling
 - ◆ testing

```
(deftest mappedchars-relaxed-test
  (testing "Problematic characters mapping to SEPA-safe characters in relaxed mode"
    (is (= "acCeeEgGhhHHiiIIjJkllLorRsStTuuUUwWyY"
          (sepa-utils/sanitize-string-relaxed "aĉĈēeĚğĜhĥĤHıİİjĴkŁorŖšŦtFuüÜŰŵŴÿŶ")))))
```



Clojure Ecosystem

- DEV environments
 - ◆ IntelliJ IDEA + Cursive
 - ◆ Emacs + Cider, VS Code + Calva
- Build tools → tools.deps vs Leiningen
- Unified Docker environment for running tests
 - ◆ on DEV computer
 - ◆ in CI pipelines



Clojure Community

→ Social Channels

- ◆ Clojurians Slack Channel → <http://clojurians.net/>
- ◆ Clojure user groups → https://clojure.org/community/user_groups#europe

→ Conferences

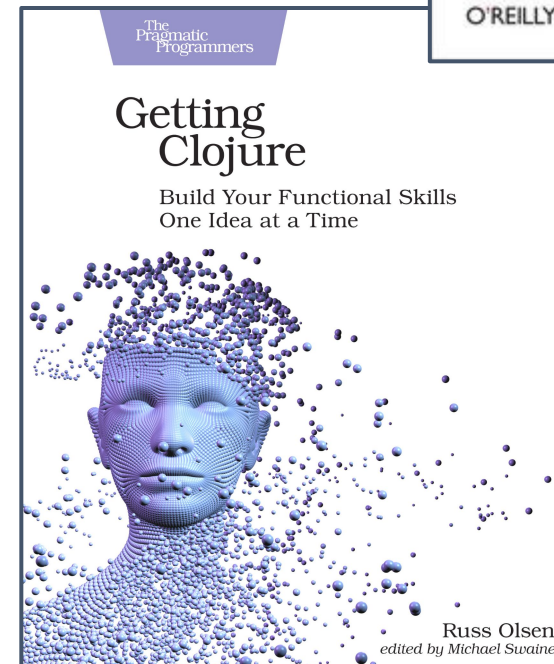
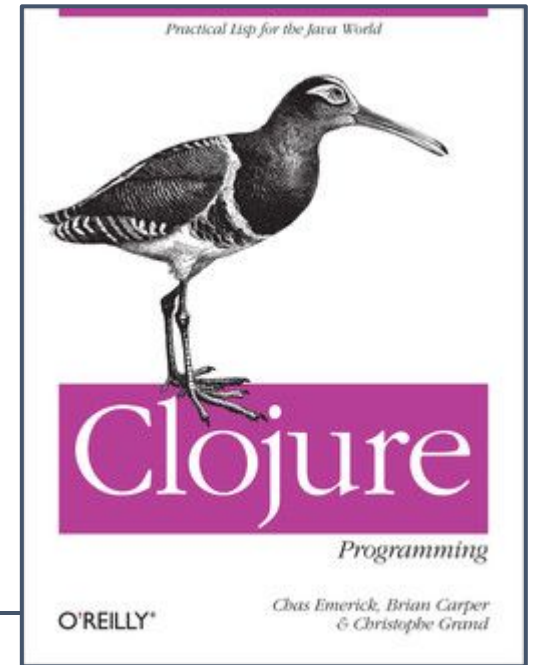
- ◆ ClojureD → <https://clojured.de/>
- ◆ Clojure Events → <https://clojure.org/community/events>



Clojure resources

→ Books

- ◆ Clojure Programming (O'Reilly Media 2012)
- ◆ Getting Clojure (Pragmatic Programmers, 2018)



Clojure resources

→ Blogs/Podcasts

◆ Clojure Deref →

<https://clojure.org/news/news>



OVERVIEW REFERENCE API RELEASES GUIDES COMMUNITY DEV NEWS

Clojure Deref (Oct 28, 2022)

Clojure Deref (Oct 21, 2022)

Clojure Deref (Oct 14, 2022)

Clojure Deref (Oct 10, 2022)

Clojure Deref (Oct 3, 2022)

Clojure Deref (Sep 26, 2022)

Clojure Deref (Sep 16, 2022)

Clojure Deref (Sep 9, 2022)

Clojure Deref (Sep 2, 2022)

Clojure Deref (Aug 26, 2022)

Clojure Deref (Aug 19, 2022)

Clojure Deref (Aug 12, 2022)

Clojure Deref (Aug 5, 2022)

Clojure Deref (July 30, 2022)

Clojure Deref (July 15, 2022)

Clojure Deref (Oct 28, 2022)

28 October 2022

Alex Miller

Welcome to the Clojure Deref! This is a weekly link/news roundup for the Clojure ecosystem. ([@ClojureDeref](#) [RSS](#))

Podcasts and videos

- [E82 Rewrites with Orestis Markou \(part 1\)](#) - ClojureStream podcast
- [Clojure visual-tools meeting 13 - interactive datavis: Shiny, Clojurescript Bayesian animations - Sci Cloj](#)
- [666: Episode of the Beast](#) - Parens of the Dead
- [Type conversion and generating big Fibonacci numbers in Clojure](#) - Clojure Diary

Blogs and articles

- [Make Clojure development tasks simple and consistent](#) - Practicalli
- [Deploying a Clojure project with neil and tools.build](#) - Michiel Borkent
- [Automating Salesforce DX using Babashka](#) - NextDoc
- [The Joy of Static Analysis](#) - Pitch

Libraries and tools

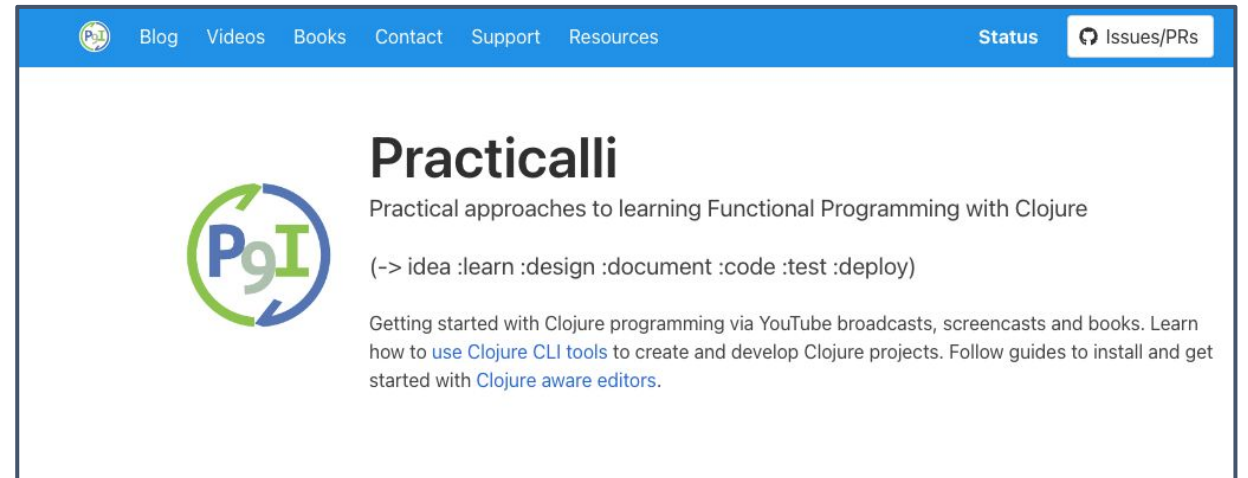
New releases and tools this week:



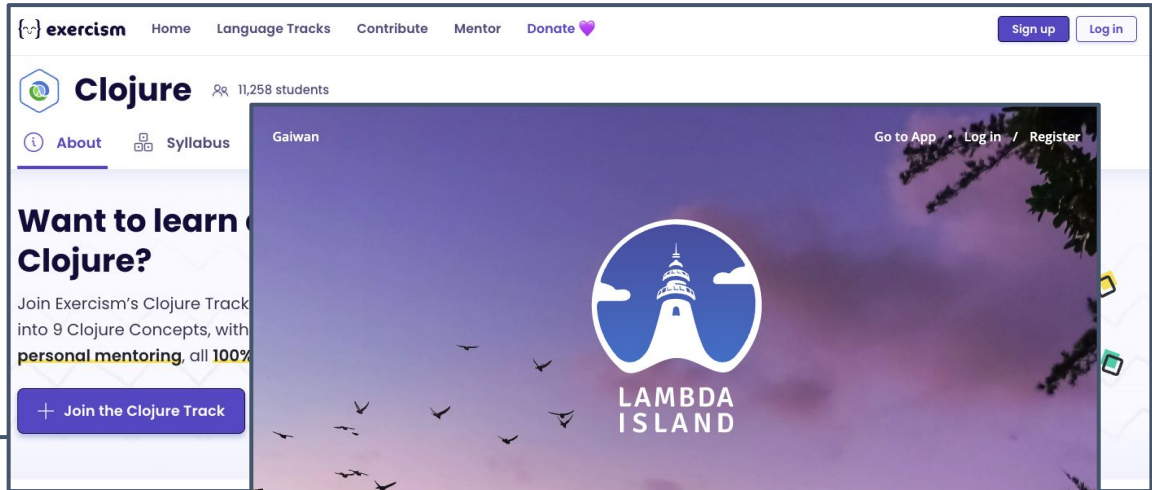
Clojure resources

→ Learning platforms


- ◆ <https://practical.li/>
- ◆ <https://exercism.org/tracks/clojure>
- ◆ <https://lambdaisland.com/>



The screenshot shows the Practical.li website. The header includes navigation links for Blog, Videos, Books, Contact, Support, and Resources, along with a Status indicator and a link to Issues/PRs. The main content area features the Practical.li logo (a green circle with 'PgI' inside) and the title 'Practical.li'. Below the title is a subtitle 'Practical approaches to learning Functional Programming with Clojure' and a tagline '(-> idea :learn :design :document :code :test :deploy)'. A paragraph of text describes the resources available, including YouTube broadcasts, screencasts, books, and guides for Clojure CLI tools and editors.



The screenshot shows the Exercism website's Clojure track page. The header includes navigation links for Home, Language Tracks, Contribute, Mentor, and Donate, along with Sign up and Log in buttons. The main content area features the Exercism logo and the title 'Clojure' with a search icon and '11,258 students'. Below the title are links for About and Syllabus. A large banner for 'Want to learn Clojure?' is visible, with a button to 'Join the Clojure Track'.



The screenshot shows the Lambda Island website. The header includes navigation links for Home, Language Tracks, Contribute, Mentor, and Donate, along with Sign up and Log in buttons. The main content area features the Lambda Island logo (a blue circle with a white lighthouse) and the title 'Clojure & ClojureScript education.' The background is a sunset scene with birds flying.



Contact us



→ ct@juston.com
→ aj@juston.com

JustOn GmbH
Mälzerstraße 3
07745 Jena
Deutschland

T: +49 3641 5598580
M: service@juston.com
www.juston.com