

# **Web Applications with Javalin**

Manish Chakravarty

# Web Applications with Javalin

Manish Chakravarty

This book is for sale at <http://leanpub.com/javalin>

This version was published on 2018-08-05



Leanpub

This is a [Leanpub](#) book. Leanpub empowers authors and publishers with the Lean Publishing process. [Lean Publishing](#) is the act of publishing an in-progress ebook using lightweight tools and many iterations to get reader feedback, pivot until you have the right book and build traction once you do.



This work is licensed under a [Creative Commons Attribution-NonCommercial 4.0 International License](#)

*For Misha and Roop*

# Contents

<b>Chapter One: Introduction to Javalin.</b> . . . . .	<b>1</b>
What is Javelin? . . . . .	1
Philosophy . . . . .	1
Summary . . . . .	5

# Chapter One: Introduction to Javalin.

## What is Javelin?

Javalin is a web framework runs on top of Jetty, one of the most used and stable web-servers on the JVM. You can configure the Jetty server fully, so you can easily get SSL and HTTP2 and everything else that Jetty has to offer. You can use Javalin with either Java or Kotlin languages

Javalin started as a fork of the Java and Kotlin web framework [Spark](#)<sup>1</sup>, but quickly turned into a ground-up rewrite influenced by [koa.js](#)<sup>2</sup>. Both of these web frameworks are inspired by the modern micro web framework grandfather: [Sinatra](#)<sup>3</sup>, so if you're coming from Ruby then Javalin shouldn't feel *too* unfamiliar.

## Philosophy

Like Sinatra, Javalin is not aiming to be a full web framework, but rather just a lightweight REST API library (or a micro framework, if you must). There is no concept of MVC, but there is support for template engines, WebSockets, and static file serving for convenience. This allows you to use Javalin for both creating your RESTful API backend, as well as serving an `index.html` with static resources (in case you're creating an SPA). This is practical if you don't want to deploy an apache or nginx server in addition to your Javalin service.

---

<sup>1</sup><http://sparkjava.com>

<sup>2</sup><http://koajs.com/#application>

<sup>3</sup><http://www.sinatrarb.com/>

If you wish to use Javalin to create a more traditional website instead of a REST APIs, there are several template engine wrappers available for a quick and easy setup.

## API design

All the methods on the Javalin instance return `this`, making the API fully fluent. This will let you create a declarative and predictive REST API, that will be very easy to reason about for new developers joining your project.

## Java and Kotlin interoperability

Javalin is both a Kotlin web framework and a Java web framework, meaning the API is being developed with focus on great interoperability between the two languages. The library itself is written primarily in Kotlin, but has a few core classes written in Java to achieve the best interoperability between the two languages. Javalin is intended as a “foot in the door” to Kotlin development for companies that already write a lot of Java.

When moving a Javalin project from Java to Kotlin, you shouldn't need to learn a new way of doing things. To maintain this consistent API for both languages is an important goal of the project.

## Sailient features of Javalin

### Simple

Unlike other Java and Kotlin web frameworks, Javalin has very few concepts that you need to learn. You never have to extend a class and you rarely have to implement an interface.

## Lightweight

Javalin is just a few thousand lines of code on top of Jetty, which means its performance is almost equivalent to pure Jetty. It also means it's very easy to reason about the source code.

## Active

A new version of Javalin has been released twice a month (on average) since the first version. Don't worry though, every version is backwards compatible. PRs and issues are reviewed swiftly, normally every week.

## Interoperable

Other Java and Kotlin web frameworks usually offer separate version for each language. Javalin is being developed with interoperability in mind, so apps are built the same way in both Java and Kotlin.

## Flexible

Javalin is designed to be simple and blocking, as this is the easiest programming model to reason about. However, if you set a Future as a result, Javalin switches into asynchronous mode.

## Why should I use Javalin over Spring Boot or Vert.x?

Javalin is a microservice framework with a built in app server (making it easy to produce *fat jars*). Unlike Vert.x this has a simple programming model and is easier to code, debug and maintain. Spring Boot is "heavy" framework and typically requires a lot of setup for simple API app. The binaries produced are also fatter.

Javalin helps you write API based applications easily with minimal overhead and gets the job done. You also have full access to the power of JVM threading and asynchronous calls. It is, of course, possible to use all existing Java libraries, thus making Javalin an easy fit in your standard Java development ecosystem.

## Can I write asynchronous code with Javalin?

Yes you can. Just set a `CompletableFuture<String>` or `CompletableFuture<InputStream>` as your result:

```
1  import io.javalin.Javalin
2
3  fun main(args: Array<String>) {
4      val app = Javalin.start(7000)
5      app.get("/") { ctx -> ctx.result(getFuture()) }
6  }
7
8  // hopefully your future is less pointless than this:
9  private fun getFuture() = CompletableFuture<String>().apply {
10 ly {
11     Executors.newSingleThreadScheduledExecutor().schedule {
12         ({ this.complete("Hello World!") }, 1, TimeUnit.SECONDS)
13     }
14 }
```

## How “Fast” is Javalin?

This really depends on your application. Javalin apps will be “as fast” as a standard Java application

## How easy is Javalin to Learn?

Javalin is a micro-framework which really does not require any additional knowledge if you are already an existing Java developer who has developed a web app before. Please note that this is *not* an introductory book and there are several excellent online resources and books available which teach Java and Java Web Programming basics. It is recommended to familiarize oneself with these concepts before proceeding.

## Summary

Javalin is a fascinating micro-framework. I really see it as a sort of NodeJS of the JVM. Something that is needed with the abundance of opinionated and heavy-weight frameworks out there.