



KARSTEN SILZ
25 AUGUST 2020



LEARNING BY CODING: BETTER JAVA PROJECTS FASTER WITH JHIPSTER



SLIDES & CODE

[HTTPS://BPF.LI/PJH4](https://bpf.li/pjh4)

DO YOU PICK TECHNOLOGIES &
TOOLS FOR JAVA PROJECTS?

CAN YOU LEARN FROM
RUNNING CODE?



JHIPSTER

BETTER JAVA PROJECTS FASTER
BY LEARNING FROM RUNNING CODE

WHY BELIEVE ME?

**JAVA DEVELOPER WHO PICKED
TECHNOLOGIES & TOOLS
FOR 21 YEARS**

**JAVA DEVELOPERS TODAY PICK MORE
TECHNOLOGIES & TOOLS MORE OFTEN
LEARNING BY CODING WITH JHIPSTER
MAKES THAT BETTER AND FASTER!**



**ADAM THE
ARCHITECT**



**THERESE
THE TESTER**



**STEVE'S SIDE
PROJECT**

ADAM THE ARCHITECT

Evaluates how to replace existing JEE monolith

- ▶ Microservices vs monolith
- ▶ NoSQL vs. relational databases
- ▶ Reactive back-end vs servlets
- ▶ Single Page Application (SPA) UI
- ▶ Run in Kubernetes
- ▶ Monitor application



THERESE THE TESTER

Better test a Spring Boot application with Angular UI

- ▶ Move from Maven to Gradle
- ▶ Combine Spring Boot & Angular Git projects
- ▶ Write Angular tests: Component, browser, load
- ▶ Run tests with Github Actions



STEVE'S SIDE PROJECT

Host small **Java** side application quickly on **Internet**, learn about **SPA**

- ▶ Quickly build Java back-end
- ▶ SPA UI
- ▶ Responsive layout
- ▶ Spring Security in Javascript
- ▶ Internationalization
- ▶ Host Java application on the Internet



SYSTEMIC EVIDENCE FOR
MORE TECHNOLOGIES &
TOOLS MORE OFTEN?

SHIFT LEFT, DEVOPS

CODE

Developers

TEST

QA

Developers

RUN

Ops

Developers

SUPPORT

Support

Developers

SHIFT LEFT

DEVOPS

MICROSERVICES

BUSINESS LOGIC

SPRING

JAVA 8

MYSQL

MICROSERVICE 1

**WE ARE FREE
TO CHOOSE
DIFFERENT
JAVA STACKS!**

BUSINESS LOGIC

MICRONAUT

JAVA 11

MONGODB

MICROSERVICE 2

NEW USER INTERFACES

JAVA BACK-END

Java
Application

Java Build
Environment

USER INTERFACES

Web
Application

Web Build
Environment

iOS
Application

iOS Build
Environment

Android
Application

Android Build
Environment

CLOUD

OUR ENVIRONMENT

Components

Application

Organization

Commercial

Open Source

WE SPEND **TIME** TO
CODE, TEST, RUN &
SUPPORT THIS

FREE TO
SUBSTITUTE
TIME FOR MONEY!

CLOUD VENDOR

Components

We don't care!

VENDOR CODES, TESTS,
RUNS & SUPPORTS
THIS FOR **MONEY**

**JAVA DEVELOPERS TODAY PICK MORE
TECHNOLOGIES & TOOLS MORE OFTEN**

**LEARNING BY CODING WITH JHIPSTER
MAKES THAT BETTER AND FASTER!**

MORE OFTEN: HIGHER UPDATE SPEED

JDK: From 3-5 years to 3 years (LTS) and 6 months (non-LTS)

Spring Boot: From 12 months to 6 months

Typescript ("Javascript with types"): 2-3 months

MORE OFTEN: MORE ALTERNATIVES

JDK: From Sun/Oracle JDK to multiple distributions (AdoptJDK HotSpot/OpenJ9, Amazon, SAP, Zulu, ...)

Framework: From "Spring, what else?" to Eclipse MicroProfile/Micronaut/Quarkus/... and "JEE comeback?"

Javascript: There's always multiple libraries for everything, new ones appear all the time

**JAVA DEVELOPERS TODAY PICK MORE
TECHNOLOGIES & TOOLS MORE OFTEN**

**LEARNING BY CODING WITH JHIPSTER
MAKES THAT BETTER AND FASTER!**

POLL TIME!

COMPARED TO 5 YEARS AGO, DO
YOU SPEND MORE TIME PICKING
TECHNOLOGIES & TOOLS IN
YOUR JAVA PROJECTS TODAY?

Royalty Free Music from [Bensound](#)

**JAVA DEVELOPERS TODAY PICK MORE
TECHNOLOGIES & TOOLS MORE OFTEN**

**LEARNING BY CODING WITH JHIPSTER
MAKES THAT BETTER AND FASTER!**

ARTICLES

**RUNNING
CODE**

CONSULTANTS

BOOKS

LEARNING

**ON-SITE
TRAINING**

VIDEOS

TALKS

**ONLINE
TRAINING**

HOW DO WE GET RUNNING
CODE OF UNKNOWN
TECHNOLOGIES & TOOLS?



JHIPSTER

OUR SUPERSTAR DEVELOPER

**PICKS, CONFIGURES, AND UPDATES
BEST TECHNOLOGIES & TOOLS**

WRITES ALL THE “PLUMBING CODE”

WHAT DOES JHIPSTER DO EXACTLY?

Open-source command line tool

1. Picks & configures technologies & tools
2. Generates full Spring Boot monolith or microservice application
3. Generates CRUD front-end & back-end for our entities
4. Generates CI files, deploys application
5. Updates application & entities

JHIPSTER PROS

Free

Complete project with UI,
build, tests & monitoring

Uses our own data model

Production level code

Up-to-date libraries

JHIPSTER CONS

Limited selection

Slow Spring Boot updates

UI not consumer-friendly

WHAT CAN WE LEARN ABOUT TECHNOLOGY/ TOOL FROM RUNNING CODE?

Configure for our environment

Fully



Best practices

Some



Low complexity technology/tool

Fully



High complexity technology/tool

Some



Comparison of technologies/tools

A lot



LEARNING FROM RUNNING CODE IS
NOT ENOUGH IN MOST CASES...

...BUT IT LEADS TO
BETTER RESULTS FASTER!

JHIPSTER DEMO

APPLICATION

GENERATION

JHipster picked and configured
technologies & tools

JHipster started project & entity
CRUD generation

We picked architecture (monolith)

We picked technologies & tools
(like PostgreSQL or Angular)



WHAT DID WE PROVE?

WHAT DOES JHIPSTER DO EXACTLY?

Open-source command line tool

- ✓ Picks & configures technologies & tools
- 2. Generates full Spring Boot monolith or microservice application
- 3. Generates CRUD front-end & back-end for our entities
- 4. Generates CI files, deploys application
- 5. Updates application & entities

ADAM THE ARCHITECT

Evaluate how to replace existing JEE monolith

- ✓ Microservices vs monolith
- ✓ NoSQL vs. relational databases
- ✓ Reactive back-end vs servlets
- ✓ Single Page Application (SPA) UI
 - ▶ Run in Kubernetes
 - ▶ Monitor application



THERESE THE TESTER

Better test a Spring Boot application with Angular UI

- ✓ Move from Maven to Gradle
 - ▶ Combine Spring Boot & Angular Git projects
- ✓ Write Angular tests: Component, browser, load
 - ▶ Run tests with Github Actions



STEVE'S SIDE PROJECT

Host small Java side application quickly on Internet, learn about SPA

- ▶ Quickly build Java back-end
- ✔ SPA UI
- ▶ Responsive layout
- ▶ Spring Security in Javascript
- ▶ Internationalization
- ▶ Host Java application on the Internet



IS JHIPSTER SAFE TO USE?

Stable project: 1.0 in Sep 2014, currently at 6.10.1 (Jul 2020)

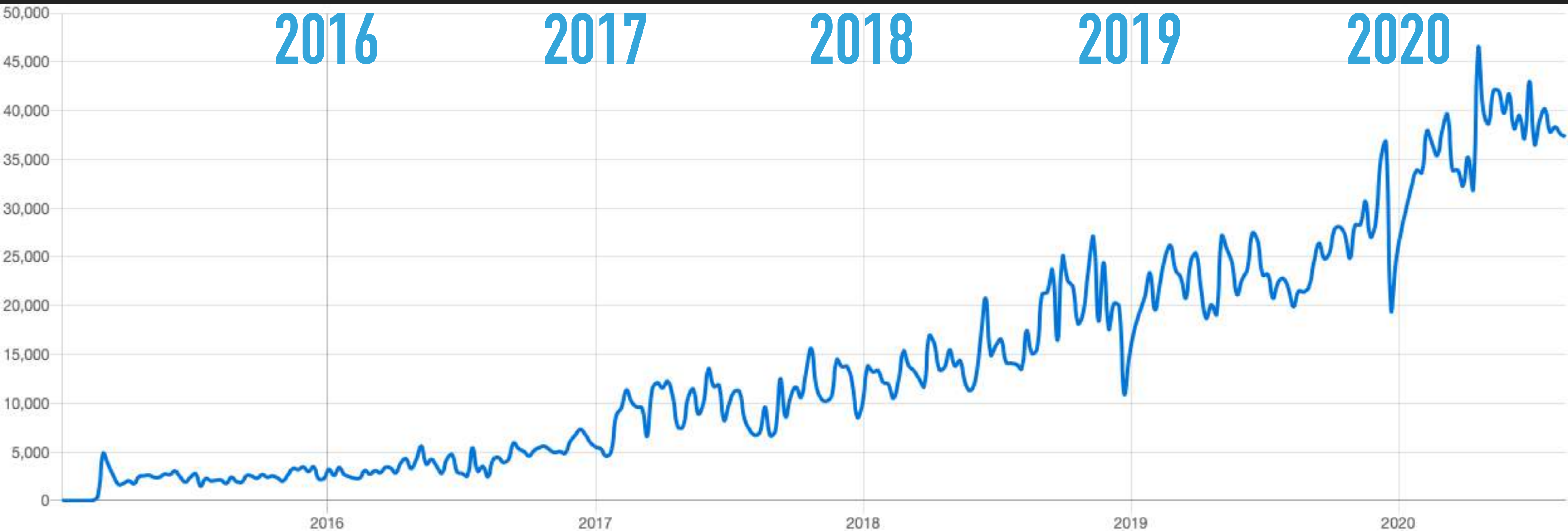
Unlike Grails, JHipster adds no proprietary layers, just 1 Maven & 1 NPM library (security & internationalization)

JHipster creates best practice code

We could stop using JHipster after generating our application

JHipster lead developer "Java Developer Advocacy manager" at Microsoft since 2019

JHIPSTER WEEKLY DOWNLOADS: 37,000



<https://www.npmtrends.com/generator-jhipster>

HOW CAN WE GET JHIPSTER?

Install it with `npm` from **Node** (LTS version)

Also needed: **Java** 8/11-14 and **Git** client

Recommended: **Docker**

DATA MODEL: JHIPSTER DOMAIN LANGUAGE (JDL)

```
entity Product {  
  name String required minlength(4) maxlength(100)  
  price BigDecimal required min(0)  
  size ProductSize required  
  image ImageBlob  
  specifications Blob  
  description TextBlob  
}
```

```
relationship OneToMany {  
  ProductCategory{product} to Product{productCategory(name)}  
}
```

```
service * with serviceImpl  
dto * with mapstruct  
paginate * with pagination
```

ENTITIES WITH ATTRIBUTES
& CONSTRAINTS (BECOME
JPA ENTITIES)

ENTITY RELATIONSHIPS
(1:1, 1:M, M:1, M:N)

CONFIGURATION OPTIONS

JHIPSTER DEMO

USER INTERFACE, CI, DEPLOYMENT

JHipster finished project & entity
CRUD generation



We saw the application

We configured CI with Github Actions

We started to configure deployment to
Kubernetes and Google App Engine

WHAT DID WE PROVE?

WHAT DOES JHIPSTER DO EXACTLY?

Open-source command line tool

- ✓ Picks & configures technologies & tools
- ✓ Generates full Spring Boot monolith or microservice application
- ✓ Generates CRUD front-end & back-end for our entities
- ✓ Generates CI files, deploys app

5. Updates application & entities

ADAM THE ARCHITECT

Evaluate how to replace existing JEE monolith

- ✓ Microservices vs monolith
- ✓ NoSQL vs. relational databases
- ✓ Reactive back-end vs servlets
- ✓ Single Page Application (SPA) UI
- ✓ Run in Kubernetes
- ✓ Monitor application



THERESE THE TESTER

Better test a Spring Boot application with Angular UI

- ✓ Move from Maven to Gradle
- ✓ Combine Spring Boot & Angular Git projects
- ✓ Write Angular tests: Component, browser, load
- ✓ Run tests with Github Actions



STEVE'S SIDE PROJECT

Host small Java side application quickly on Internet, learn about SPA

- ✓ Quickly build Java back-end
- ✓ SPA UI
- ✓ Responsive layout
- ✓ Spring Security in Javascript
- ✓ Internationalization
- ✓ Host Java application on the Internet





**ADAM THE
ARCHITECT**



**THERESE
THE TESTER**



**STEVE'S SIDE
PROJECT**

WHAT DOES JHIPSTER DO EXACTLY?

Open-source command line tool

- ✓ Picks & configures technologies & tools
 - ✓ Generates full Spring Boot monolith or microservice application
 - ✓ Generates CRUD front-end & back-end for our entities
 - ✓ Generates CI files, deploys app
5. Updates application & entities

Updates

JHipster can update libraries and code for **application** and **entities**

- ▶ JHipster **re-generates** them on separate Git branch
- ▶ We need to manually **merge** that branch

Yes, your code **changes** can get **overridden**

How to avoid that is **barely documented**



HOW OFTEN DOES JHIPSTER UPDATE?

Spring Boot: 6 months for 2.1, 3.5 months for 2.2, 2.3 not yet (first 6 month release cycle, out since May 2020)

Angular: 1 month for 9.0, 4 days for Angular 10

Alternatives:

- ▶ Angular build tool: From `yarn` to `npm`
- ▶ Javascript test runner: From Karma to Jest

WHAT DOES JHIPSTER DO EXACTLY?

Open-source command line tool

- ✓ Picks & configures technologies & tools
- ✓ Generates full Spring Boot monolith or microservice application
- ✓ Generates CRUD front-end & back-end for our entities
- ✓ Generates CI files, deploys app
- ✓ Updates application & entities

**WE PROVED THAT JHIPSTER HELPS
ADAM, THERESA & STEVE**

**WE PROVED THAT JHIPSTER
DOES WHAT IT SHOULD**

**JAVA DEVELOPERS TODAY PICK MORE
TECHNOLOGIES & TOOLS MORE OFTEN
LEARNING BY CODING WITH JHIPSTER
MAKES THAT BETTER AND FASTER!**



JHIPSTER

BETTER JAVA PROJECTS FASTER
BY LEARNING FROM RUNNING CODE

The End



SLIDES & CODE

[HTTPS://BPF.LI/PJH4](https://bpf.li/pjh4)