



Get Hip with JHipster!

Matt Raible • <http://raibledesigns.com>

Photos by  McGinity Photo



RAIBLE DESIGNS
ENTERPRISE OPEN SOURCE CONSULTING



Who is Matt Raible?

Father, Skier, Mountain
Biker, Whitewater Rafter

Web Framework Connoisseur

Founder of AppFuse



Bus Lover

Blogger on raibledesigns.com

What about YOU?

Are you a Java Developer?

How long have you been doing web development?

Do you like JSF? Do you like GWT?

Do you like JavaScript?



Do you want to be a hip Java Developer?

Java 8

Parallel Collections

JSR 310 Date and Time API

Functional Interfaces with default method

Lambda Expressions (a.k.a. Closures)

```
// sort a list by lastName
List<Person> persons = ....;
persons.sort((p1, p2) ->
    p1.getLastName().compareTo(p2.getLastName()));
```

Nashorn JavaScript Engine



Hipster

n. noun

1. One who is exceptionally aware of or interested in the latest trends and tastes.



Latest trends in JavaLand?

Annotations

Environments: dev, test, production

Microservices

Container-less Deployment

Monitoring



Spring Boot

Create stand-alone Spring applications

Embeds Tomcat, Jetty or Undertow directly

Provides opinionated 'starter' POMs to simplify your Maven configuration

Also supports/promotes using Gradle



Spring Boot

Automatically configures Spring whenever possible



Provides production-ready features such as metrics, health checks and externalized configuration

Absolutely no code generation and no requirement for XML configuration

Grails 3.0 is based on Spring Boot

SPRING INITIALIZR @ start.spring.io



Spring Boot Demo

Latest trends in Web Development?

JavaScript MVC Frameworks

CSS 3 with Animations

Mobile First

Front-End Optimization

REST APIs

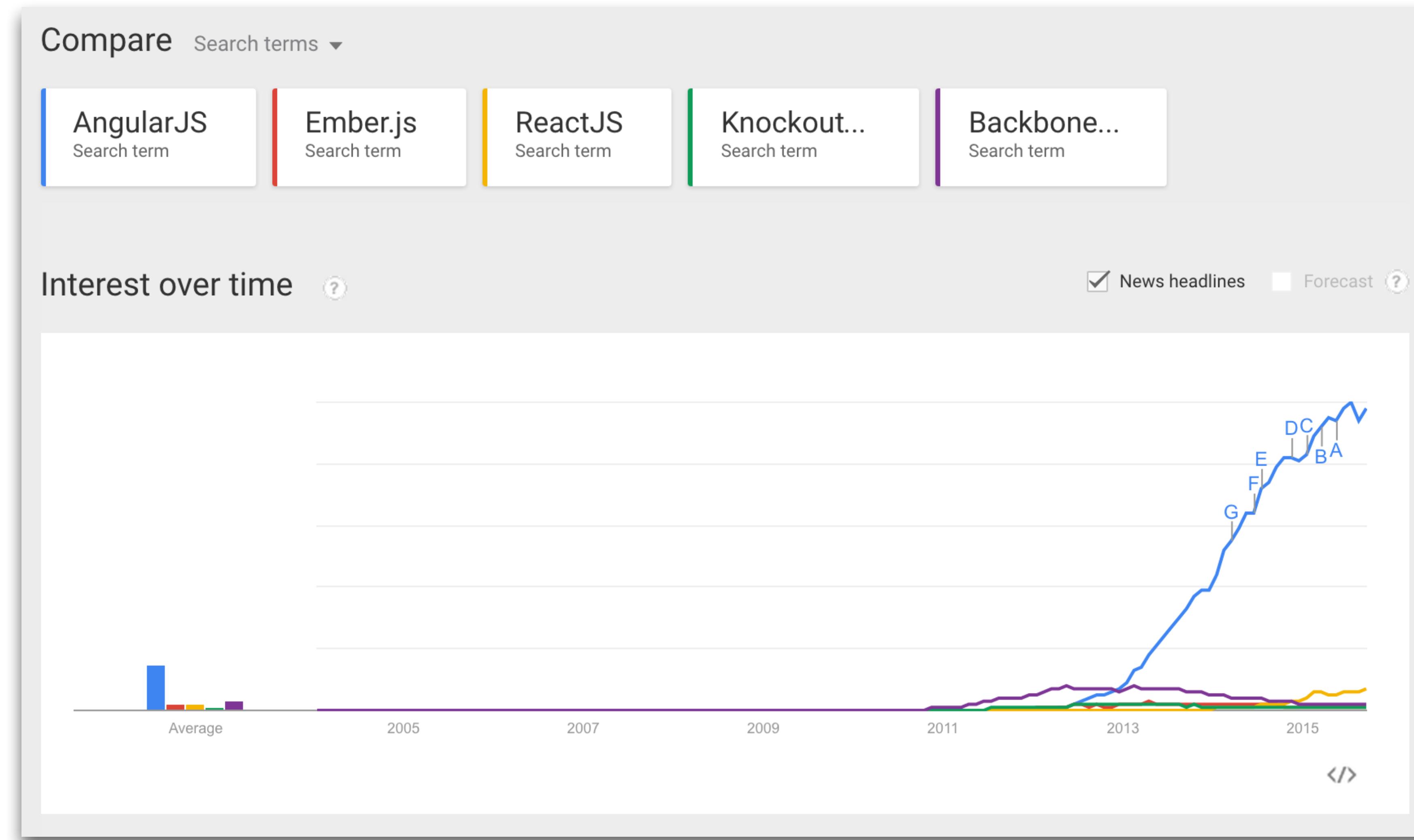




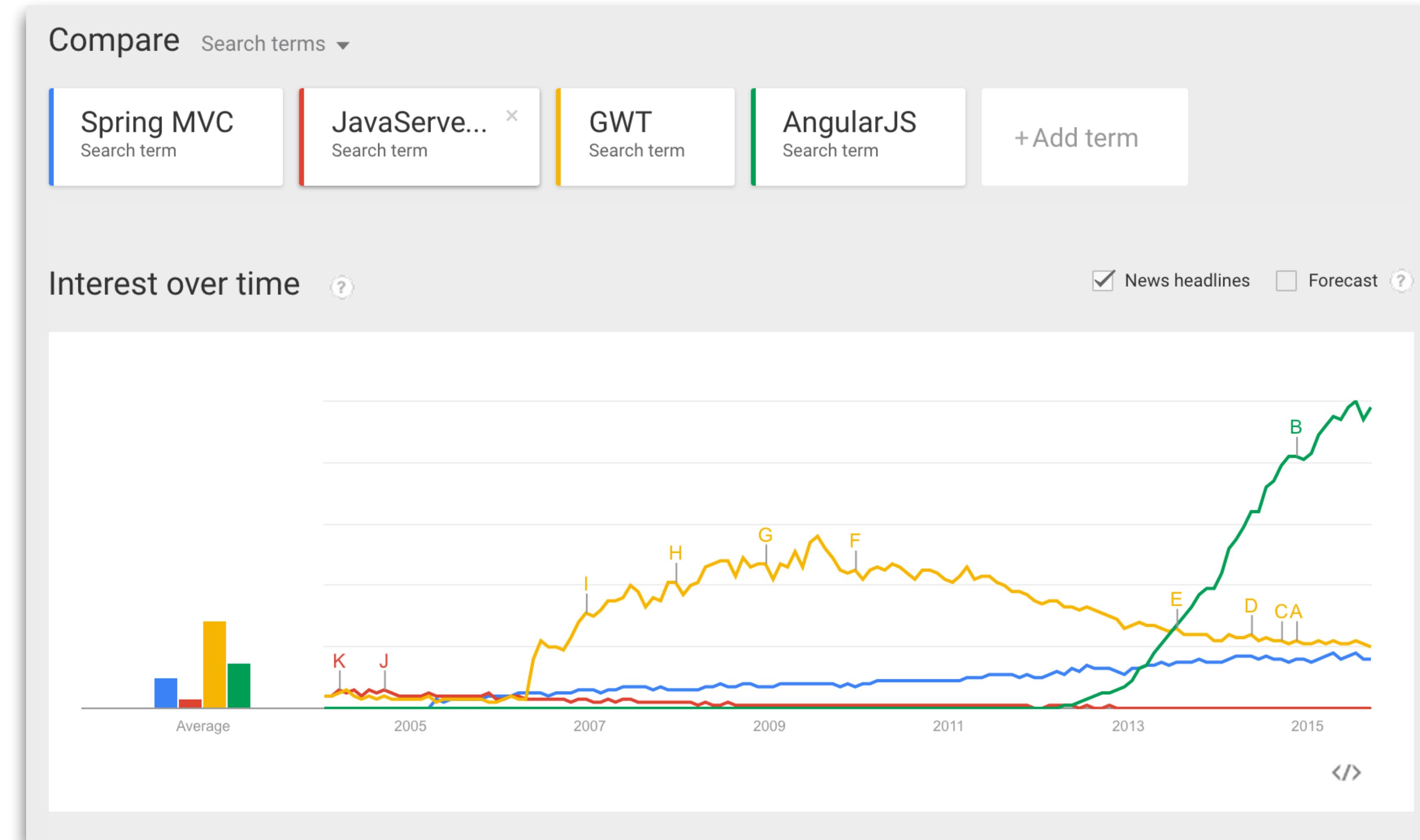
ANGULARJS

Google

Google Trends



Google Trends



The most starred project on GitHub?

Search Search

 **Repositories** 1,208,784

 **Code**

 **Issues**

 **Users**

We've found 1,208,784 repository results Sort: Best match ▾

twbs/bootstrap CSS ★ 88,810 ⚡ 37,206

The most popular HTML, CSS, and JavaScript framework for developing responsive, mobile first projects on the web.

Updated 8 hours ago 

vhf/free-programming-books ★ 45,134 ⚡ 10,379

 Freely available programming books

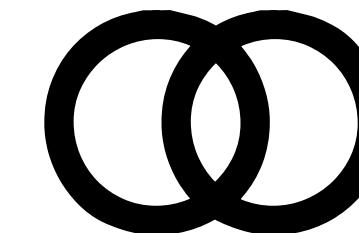
Updated 23 hours ago 

angular/angular.js JavaScript ★ 44,106 ⚡ 20,051

HTML enhanced for web apps

Languages	Count
JavaScript	238,670
Python	126,490
Java	108,622
Ruby	103,529
PHP	93,183
C	57,203
C++	50,630
Objective-C	42,560
Shell	34,696

Wouldn't it be hip if...?



HTML



CSS



JHipster

<http://jhipster.github.io/>

Greetings, Java Hipster!



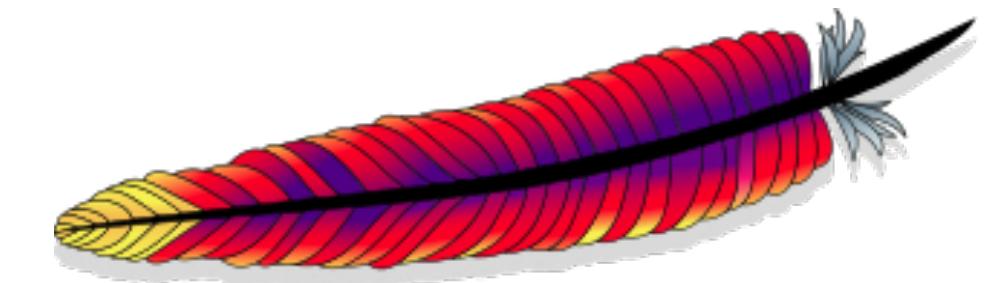
JHipster is a
Yeoman generator,



used to create a
Spring Boot + AngularJS
project.



JHipster Project



Web Statistics, October 2015

275K Page Views

10K Downloads

GitHub Statistics

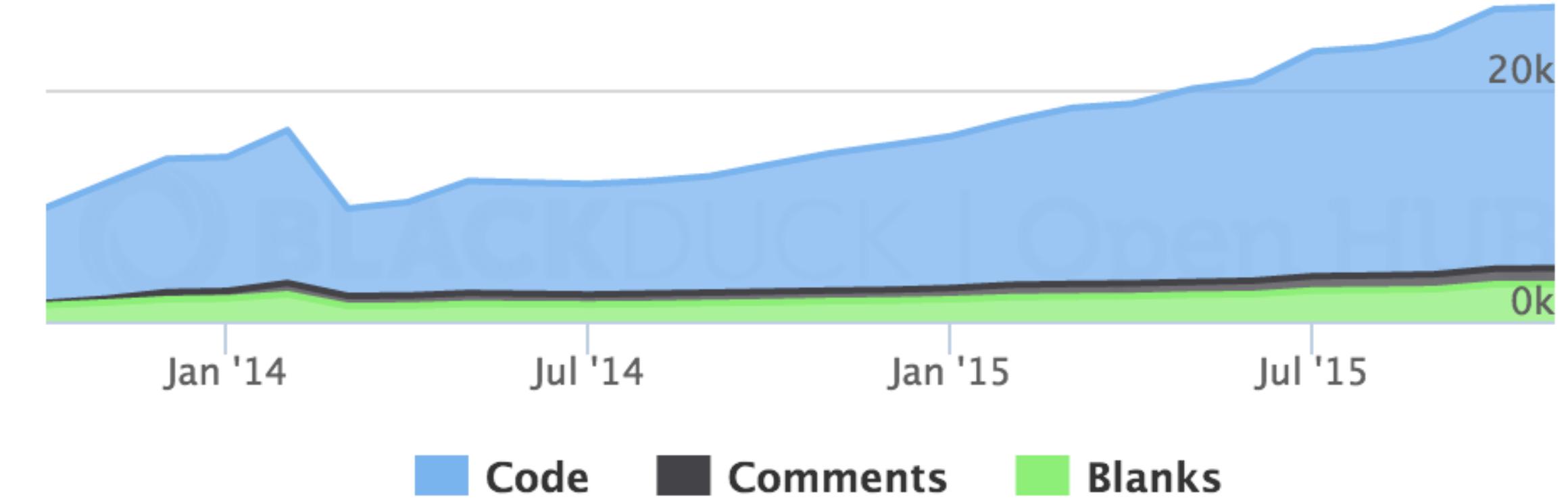
2961 Stars

159 Contributors

Languages

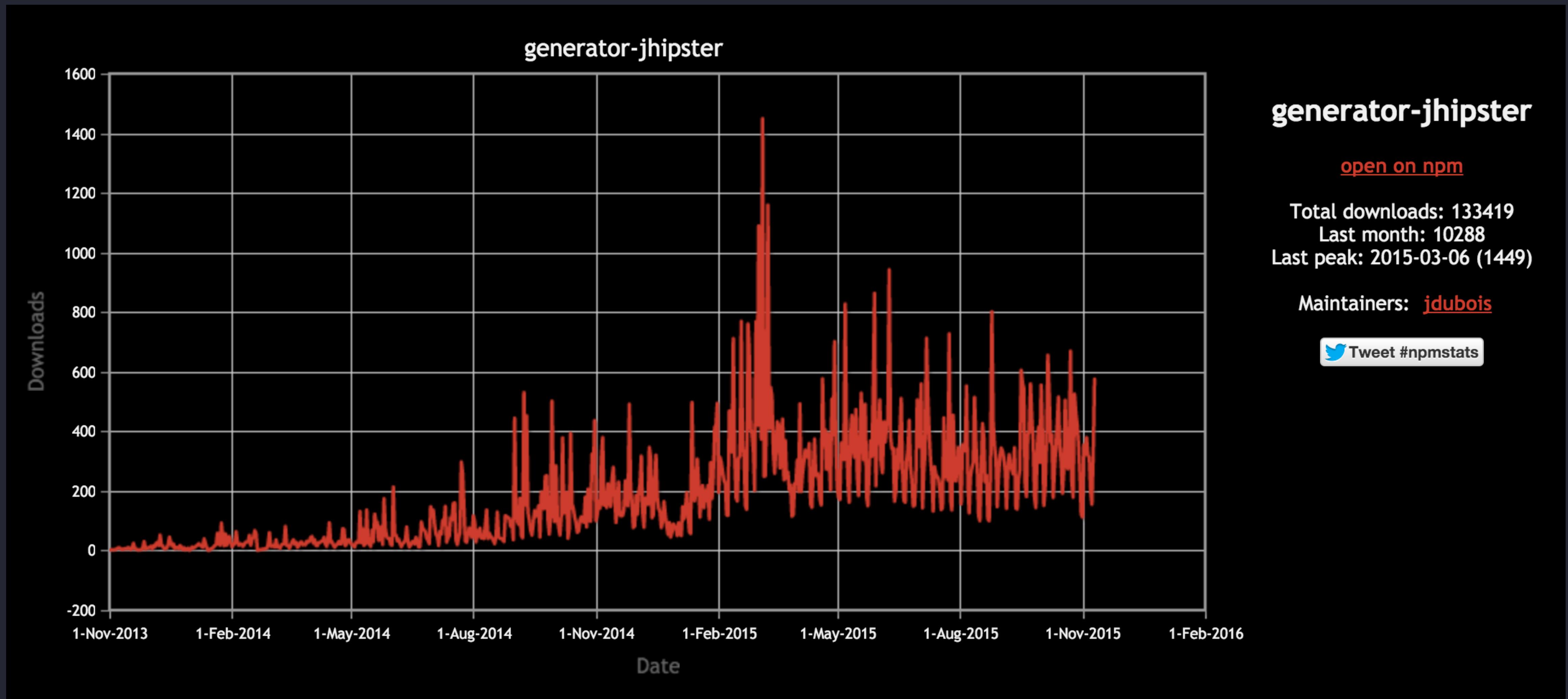


Lines of Code



<https://www.openhub.net/p/generator-jhipster>

JHipster Downloads





JHipster

Foundational Frameworks

Spring Boot

Spring Security

AngularJS

Bootstrap

Metrics

Project Options

Java 7 or Java 8

Maven or Gradle

Authentication Type: cookie-based (with Social), token-based or OAuth2

Type of Database: SQL or NoSQL

Caching: EhCache or Hazelcast

Grunt or Gulp.js

JHipster Features

Authentication and Security

Spring Boot Metrics Dashboard

Bootstrap and AngularJS Integrated

REST API with Spring MVC

JPA Support and NoSQL Support

CRUD Generation

Internationalization



Yeoman

<http://yeoman.io/>

The web's scaffolding tool for modern webapps

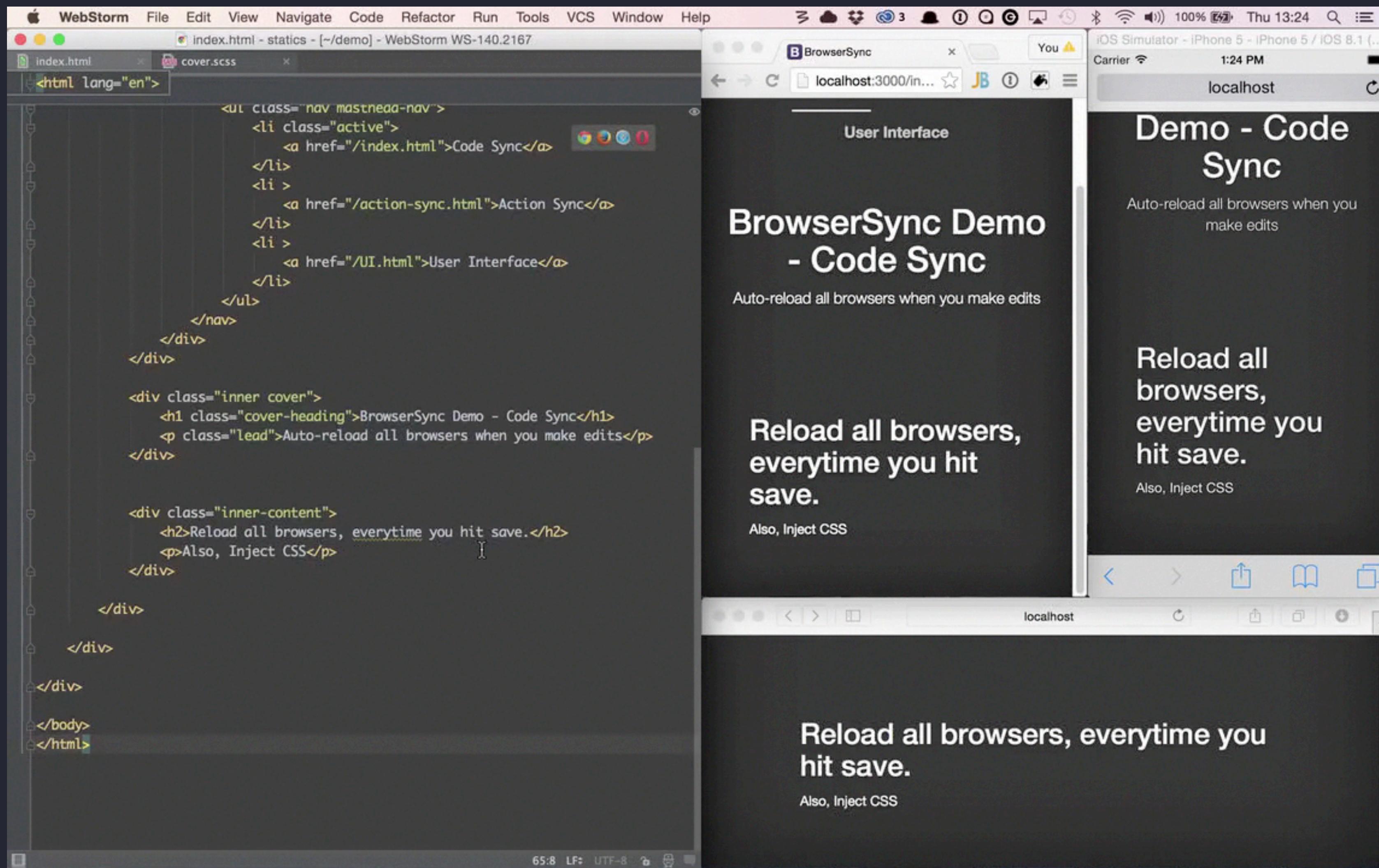
Helps you kickstart new projects

Promotes the Yeoman workflow



Browsersync

<http://www.browsersync.io/>



Demo Time!



Demo Goals

Generate a basic blog application

Look at its configuration

Generate the CRUD entities

Limit blogs to current user, allow HTML

Deploy to

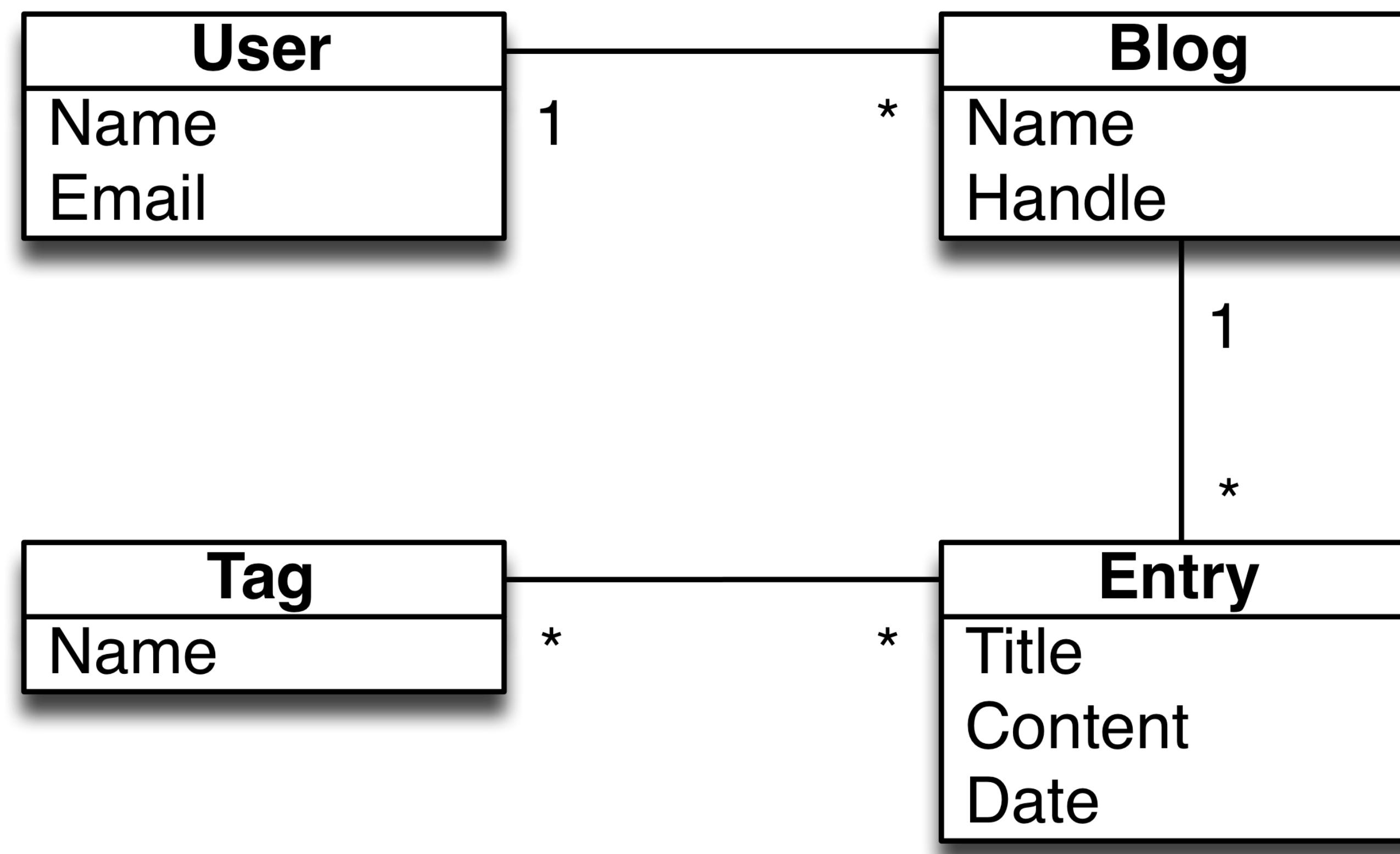


heroku



... all in 30 minutes!

Demo Application



Load Testing with Gatling



High Performance

Scala, Akka and Netty

Nice HTML-based Reporting

Scenario Recorder

```
val scn = scenario("Test the Blog entity")
    .exec(http("First unauthenticated request")
        .get("/api/account")
        .headers(headers_http)
        .check(status.is(401)))
    .pause(10)
    .exec(http("Authentication")
        .post("/api/authenticate")
        .headers(headers_http_authentication)
        .formParam("username", "admin")
        .formParam("password", "admin")
        .check(jsonPath("$.token").saveAs("x_auth_token")))
    .pause(1)
    .exec(http("Authenticated request")
        .get("/api/account")
        .headers(headers_http_authenticated)
        .check(status.is(200)))
    .pause(10)
    .repeat(2) {
        exec(http("Get all blogs")
            .get("/api/blogs")
            .headers(headers_http_authenticated)
            .check(status.is(200)))
        .pause(10 seconds, 20 seconds)
        .exec(http("Create new blog")
            .put("/api/blogs"))
    }
}
```

What about UI Testing?



Protractor
end to end testing for AngularJS

Jasmine

<http://www.jhipster-book.com/#!/news/entry/adding-protractor-tests>

How sweet is JHipster?!

AngularJS, Java 8, Microservices - what more could you want?



Angular 2.0

```
<input type="text" [value]="firstName">  
<button (click)="addPerson()">Add</button>  
  
<input type="checkbox" [checked]="someProperty">
```

Concepts Eliminated in Angular 2.0

Controllers

Directive Definition Object

\$scope

angular.module

jqLite



Angular 2: Template Syntax



The image shows a split-screen development environment. On the left, a WebStorm code editor displays the file `hello.js`. It contains the following code:

```
@Component({
  selector: 'hello'
})
@Template({
  inline: `
    <div>
      <label>Name:</label>
      <input type="text" #ref>
      <hr>
      <h1>Hello {{ref.value}}!</h1>
    </div>
    <button id="button" (custom)="onCustom()" (click)="onClick(ref)">Click me</button>
  `
})
export class Hello{
  constructor(){
    var button = document.querySelector("body /deep/ #button")
    setInterval(function(){
      button.dispatchEvent(new Event("custom"));
    }, 1000);
  }

  onClick(value){
    console.log(value);
  }

  onCustom(){
    console.log("custom");
  }
}
```

A large green play button is overlaid on the code editor. On the right, a browser window shows the resulting application. The URL is `localhost:9000`. The page has a form with a label "Name:" and an input field. Below it is a large "Hello !" heading. At the bottom is a button labeled "Click me". A developer tools sidebar shows a log entry: `41 custom VM17002:25`.

<https://egghead.io/lessons/angularjs-angular-2-template-syntax>

The JHipster Mini-Book

Written with Asciidoctor

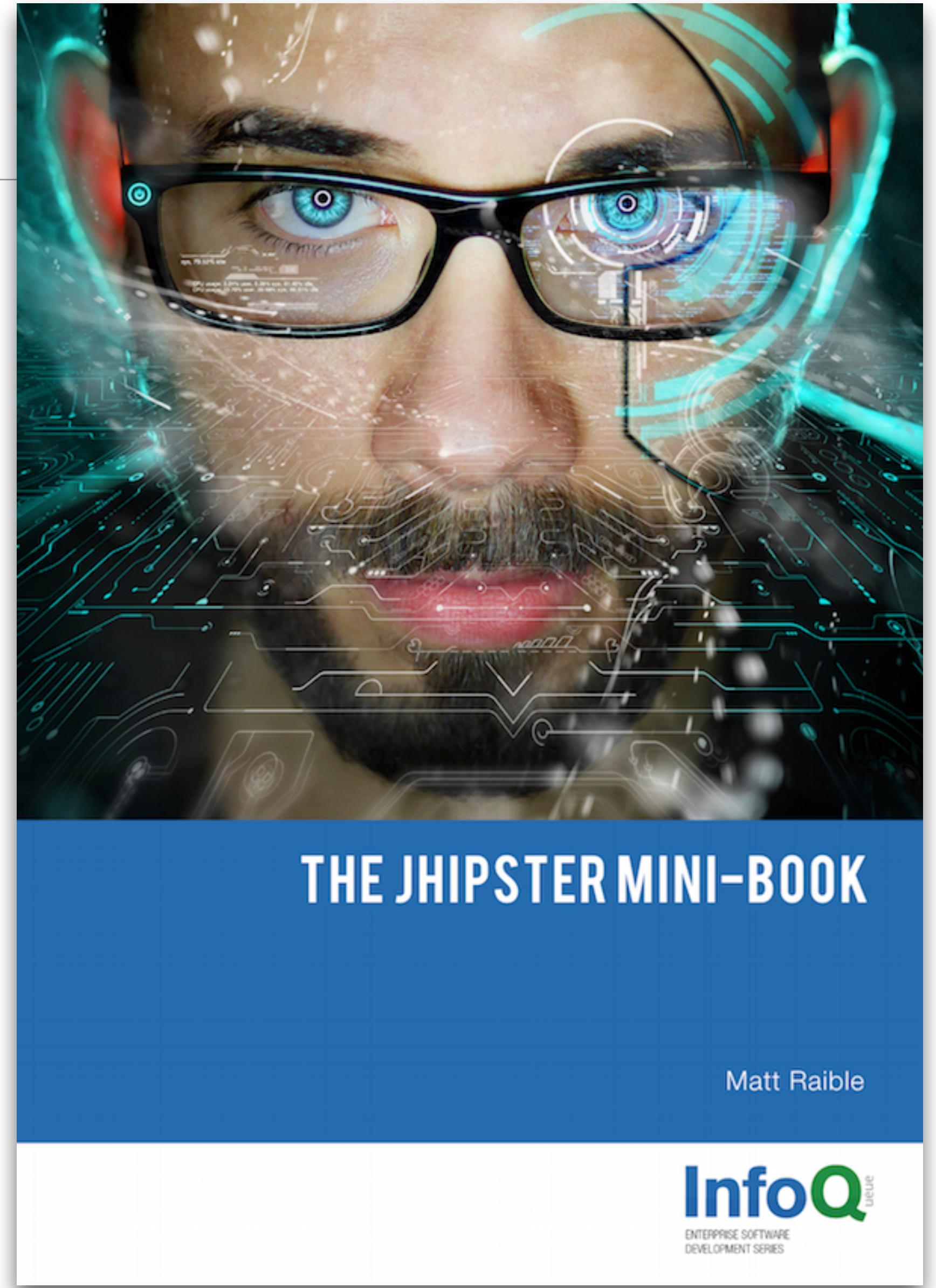
Quick and to the point, 130 pages

Developed a Real World App:

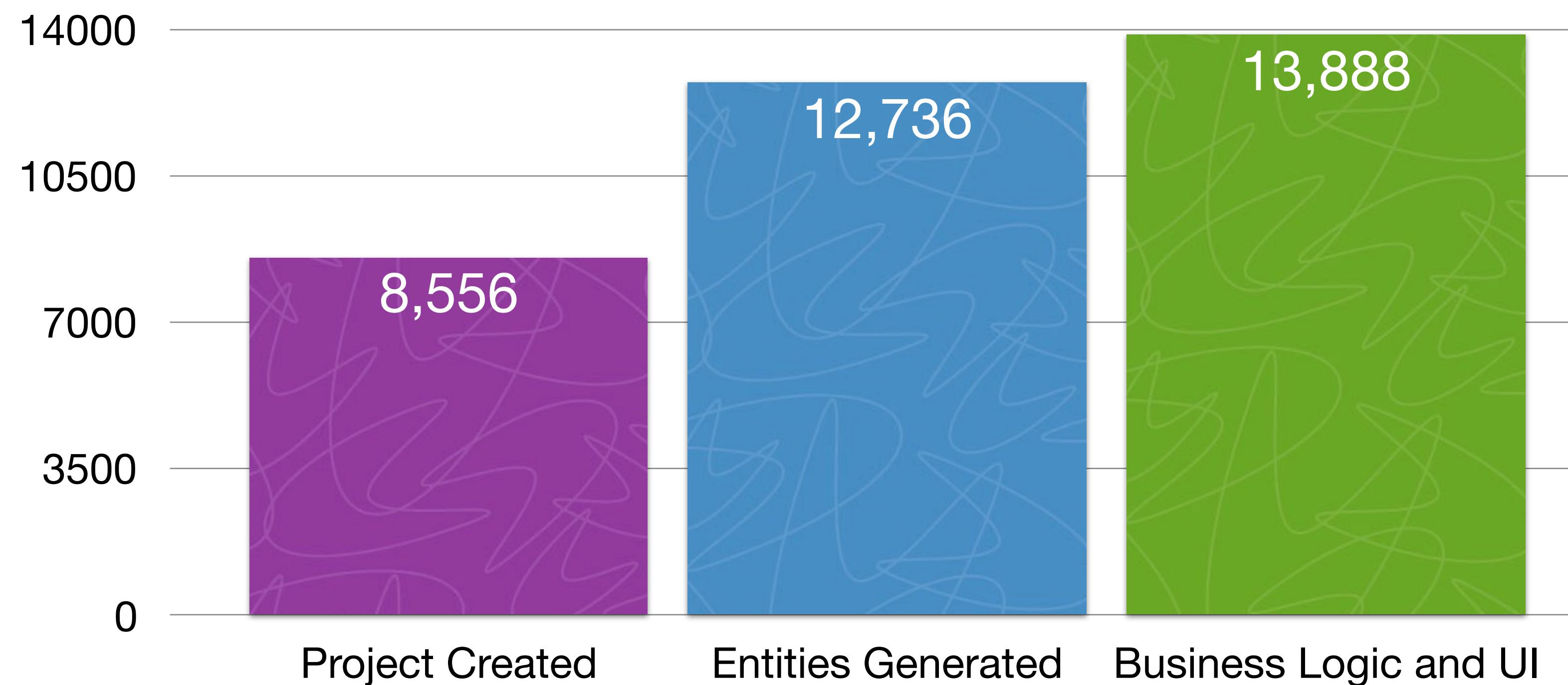
www.21-points.com

Free Download from **InfoQ**

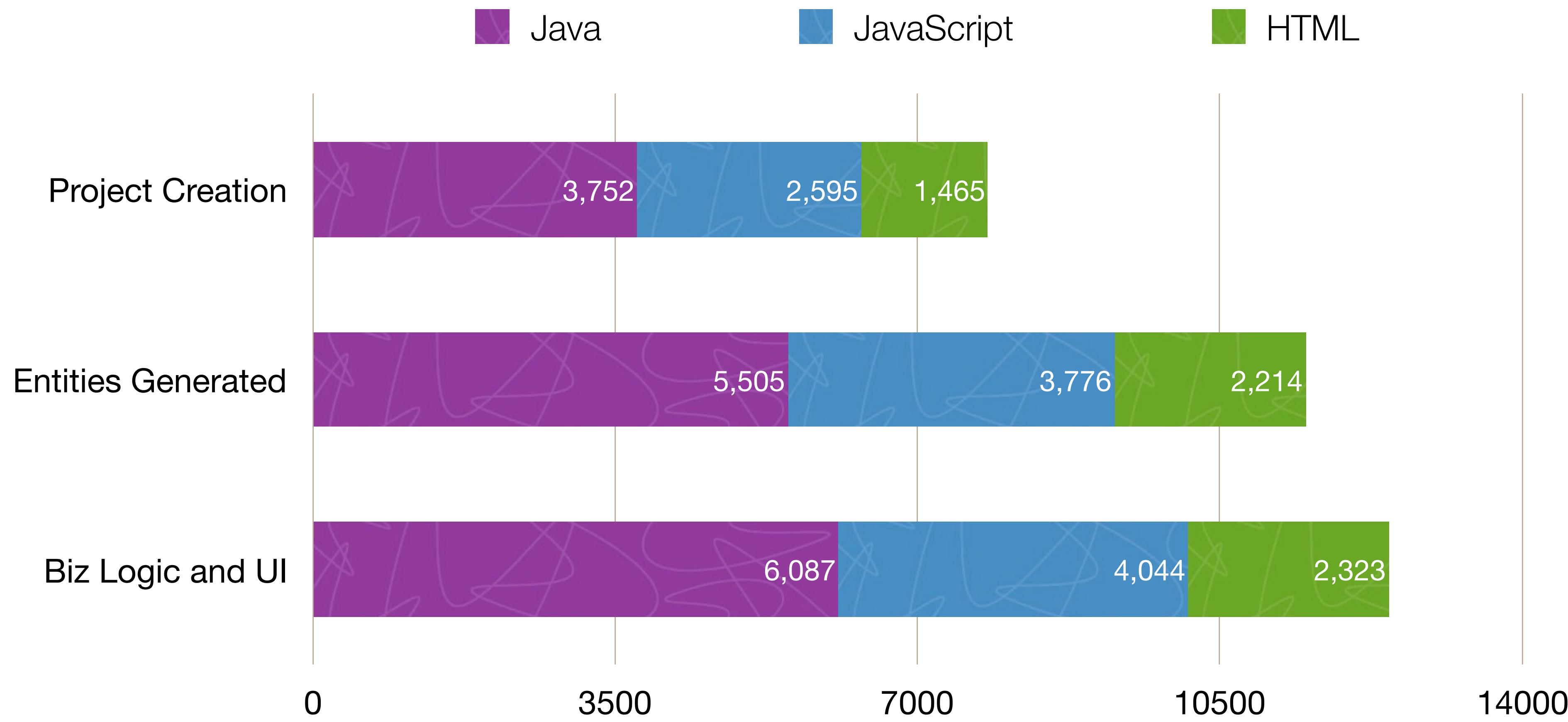
<http://www.infoq.com/minibooks/jhipster-mini-book>



Lines of Code in 21-Points



Lines of Code by Language



Learn More

 https://twitter.com/java_hipster

 <http://jhipster.github.io/>

 <https://github.com/jhipster/generator-jhipster>

 <https://www.youtube.com/watch?v=R3jm2qmqltI>

 <https://angularjs.org>

The JHipster Mini-Book

1.0 Release on October 30



<http://jhipster-book.com>



<http://21-points.com>



[@jhipster_book](https://twitter.com/jhipster_book)



Questions?

Stay hip by following me!

 <http://raibledesigns.com>

 [@mraible](https://twitter.com/mraible)

Presentations

 <http://slideshare.net/mraible>

Code

 <http://github.com/mraible>

