



Hands-on lab

JHipster

the best way to breed a new
webapp

by Peter Lindh & Rikard Thulin

What you will get...

In this hands-on-lab you will learn how to leverage and love the JHipster stack, combining best of breed Java backend with a modern single-page application (SPA) JavaScript frontend and the tooling to get you and your team productive.

We will develop a full stack application, from scratch, and deploy it in the cloud.

What you should have prepared...

Laptop with the following tools pre-installed:

- Java 8
- Maven 3
- Git
- Node.js

Peter Lindh

Software-Geek, Entrepreneur



email: peter.lindh at squeed.com

twitter: [@peterlindh](#)

Rikard Thulin

Javaforum JUG leader, Entrepreneur,
with Java since 1.0



email: rikard.thulin at squeed.com

twitter: [@Rikard_Thulin](#)

plug

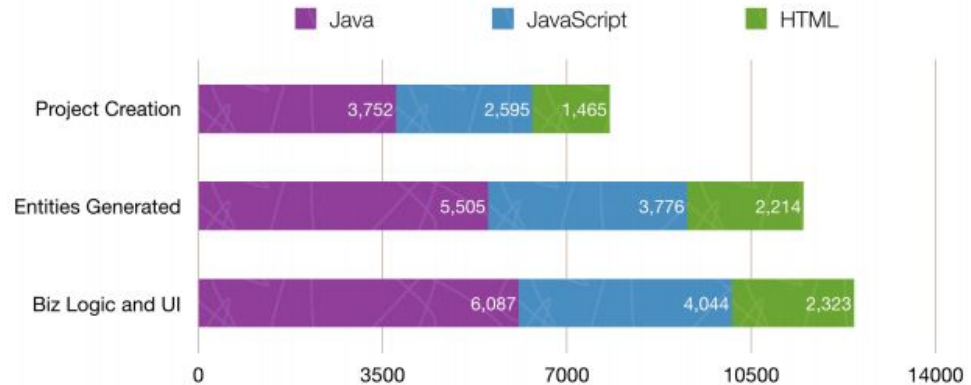
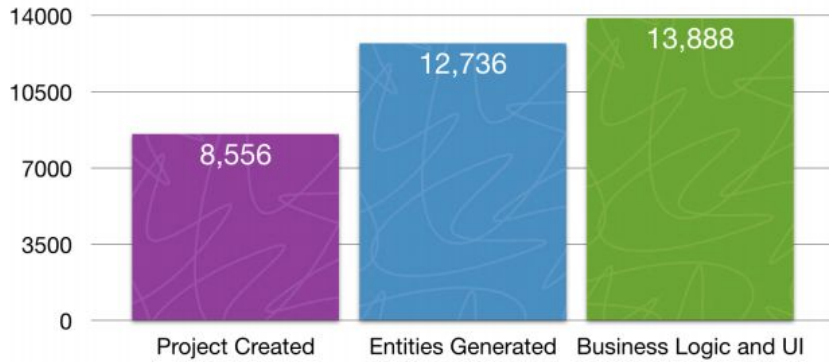
Squeed is a knowledge company caring about learning.
It is in our [awesome] **culture** and we do it with **passion**

JHipster rationale to exist

“We can guarantee you that having everything working together smoothly, with security, caching and monitoring, isn't easy...” -- [Official JHipster Slides](#)

JHipster rationale to exist

“I [Matt Raible] did some quick calculations to see how many lines of code JHipster had produced. [...] I only had to write 1,152 lines of code. JHipster did the rest for me, generating 91.7% of the code” -- Matt Raible [\[The JHipster Mini-book\]](#)



hipster

hipstə/
noun



a person who follows the latest trends and fashions, especially those regarded as being outside the cultural mainstream.

hipster

hipstə/
noun



a person who follows the latest trends and fashions, especially those regarded as being outside the cultural mainstream.

JHipster

jayhipstə/
noun

a **generator** who follows the **latest technology** trends and **best practices**, especially those regarded as being outside the technology mainstream.

Is JHipster hip enough? - self assessment

The project is about 2 years old (2016/02):

- 3,394 GitHub stars and 187 contributors
- 12,000 downloads / month
- Featured on Websites everywhere (Infoworld, SD Times, JavaWorld, Reddit...)
- And it's growing rapidly

yo 10.000 feet overview

JHipster is a Yeoman generator used to create a Spring Boot + AngularJS project



+



+



=



Client Side Options



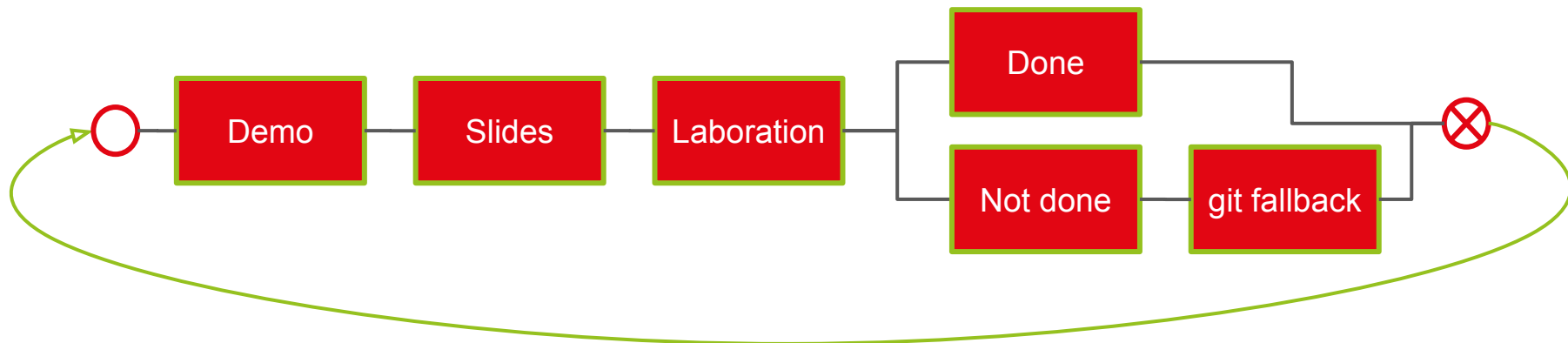
Server Side Options



JHipster hands-on lab topics

- Installation
- Generating a JHipster application
- Creating entities and relations
- Customize the user interface
- Adding a Service
- Deploy to cloud
- Testing

How we will do this



We track your progress in lab stages using: <http://presentation-poll.meteor.com/poll/JHipsterHOL>

JHipster hands-on lab topics

- **Installation**
- Generating a JHipster application
- Creating entities and relations
- Customize the user interface
- Adding a Service
- Deploy to cloud
- Testing

Installation (download Internet and its friends to...)

- Download and install Node.js (<http://nodejs.org>)
- `npm install -g yo`
- `npm install -g bower`
- `npm install -g grunt-cli`
- `npm install -g generator-jhipster`
- `git clone https://github.com/squeed-com/jhipsterhol.git`

Lab instructions:

<https://github.com/squeed-com/jhipsterhol/wiki/Installation>

Yeoman



The web's scaffolding tool for modern webapps

Yeoman helps you to kickstart new projects, prescribing best practices and tools to help you stay productive.

Yeoman provides a generator ecosystem. A generator is basically a plugin that can be run with the `yo` command to scaffold complete projects or useful parts.

```
jhipsterhol — -bash — 80x20
$ npm install -g yo
```

Note: On Unix / Linux you might need to use 'sudo' when executing 'npm'

Bower - a package manager for the web



Web sites are made of lots of things — frameworks, libraries, assets, and utilities. Bower manages all these things for you.

```
jhipsterhol — -bash — 80x20
$ npm install -g bower
```

Note: On Unix / Linux you might need to use 'sudo' when executing 'npm'



Grunt

Grunt gives you automation for repetitive tasks such as

- background execution of tests
- “live editing” of your code
- compiling Sass
- compression and optimization of resources

```
jhipsterhol — -bash — 80x20
$ npm install -g grunt-cli
```

Note: On Unix / Linux you might need to use ‘sudo’ when executing ‘npm’

generator-jhipster

The Yeoman JHipster generator



```
jhipsterhol — -bash — 80×20
$ npm install -g generator-jhipster
```

Note: On Unix / Linux you might need to use 'sudo' when executing 'npm'

Put a package in a package in a package?

Why we use three package managers

- **Maven**
 - for the Java guys, the backend stuff
 - nested dependency tree
- **npm**
 - for web developer tools like Yeoman, Grunt
 - nested dependency tree
- **bower**
 - for front-end packages
 - flat dependency tree

GitHub repository

A terminal window with a title bar containing three colored window control buttons (red, yellow, green) on the left, a folder icon, and the text "jhipsterhol — -bash — 80x20". The main area of the terminal displays a single line of text: "\$ git clone https://github.com/squeed-com/jhipsterhol.git".

```
jhipsterhol — -bash — 80x20  
$ git clone https://github.com/squeed-com/jhipsterhol.git
```

JHipster hands-on lab topics

- Installation
- **Generating a JHipster application**
- Creating entities and relations
- Customize the user interface
- Adding a Service
- Deploy to cloud
- Testing

Generating a JHipster application

- Simple application to visualise Hipster spots
 - Domain model (entities etc)
 - Map-ui
 - Geo coding integration
 - Cloud deployment

AngularJS

- The fastest growing web framework for Single Page Applications (SPA)
- AngularJS is 100% JavaScript, 100% client-side and compatible with both desktop and mobile browsers.

Spring Boot

Make Jar, not war

- Accelerates your Spring application development
- Default configurations determined by what's on the class path
- Gives you a runnable self containing jar
- Often used developing micro services
- Spring Data “magic”

yo lab> Generating a JHipster application

~20 minutes

Lab instructions

<https://github.com/squeed-com/jhipsterhol/wiki/Generating-a-JHipster-application>

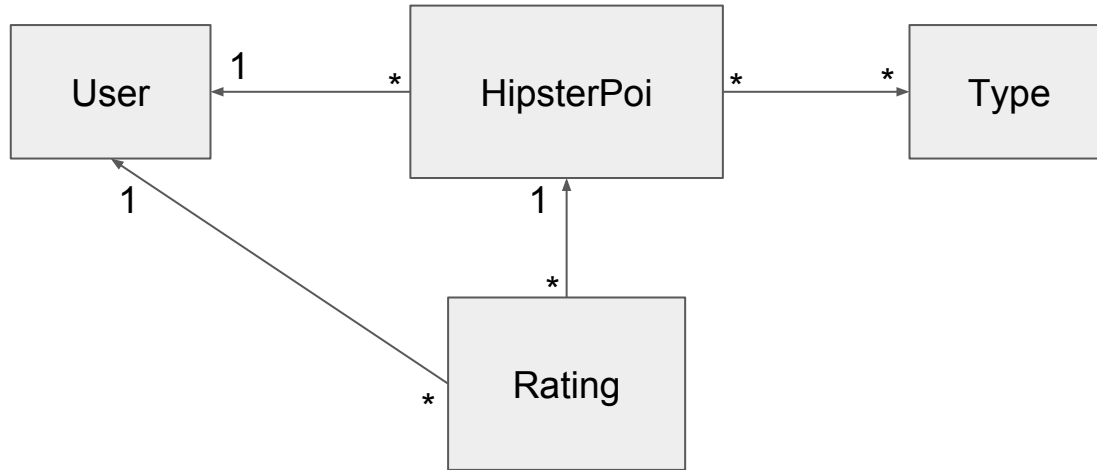
JHipster hands-on lab topics

- Installation
- Generating a JHipster application
- **Creating entities and relations**
- Customize the user interface
- Adding a Service
- Deploy to cloud
- Testing

Creating entities and relations

- “entity” sub-generator to generate a complete JPA Entity, from database to AngularJS CRUD-UI
 - A database table
 - A Liquibase change set
 - A JPA Entity
 - A Spring Data JPA Repository
 - A Spring MVC REST Controller, which has the basic CRUD operations
 - An AngularJS router, a controller and a service
 - An HTML view
 - Integration tests, to validate everything works as expected
 - Performance tests, to see if everything works smoothly

Creating entities and relations



Liquibase

- Must have tool if you work with RDBMS
- “Version control” for your database refactorings
 - Database refactorings are defined in changesets, typically XML
- Easy to use
- Easy to misuse

Key take away: If you use RDBMS you ~~should~~ must use a tool like Liquibase

yo lab> Creating entities and relations

~20 minutes

Lab instructions

<https://github.com/squeed-com/jhipsterhol/wiki/Creating-entities-and-relations>

JHipster hands-on lab topics

- Installation
- Generating a JHipster application
- Creating entities and relations
- **Customize the user interface**
- Adding a Service
- Deploy to cloud
- Testing

Customize the user interface

- Get a feel for the UI-development workflow
 - Browser Sync hot reload and interaction synchronization over several browsers
 - Bower - web package management
 - Grunt static resource management

yo lab> Customize the user interface

~20 minutes

Lab instructions

<https://github.com/squeed-com/jhipsterhol/wiki/Customize-the-user-interface>

JHipster hands-on lab topics

- Installation
- Generating a JHipster application
- Creating entities and relations
- Customize the user interface
- **Adding a Service**
- Deploy to cloud
- Testing

Adding a Service

- “service” sub-generator to generate a Spring business service

yo lab> Adding a Service

~20 minutes

Lab instructions

<https://github.com/squeed-com/jhipsterhol/wiki/Generating-a-JHipster-application>

JHipster hands-on lab topics

- Installation
- Generating a JHipster application
- Creating entities and relations
- Customize the user interface
- Adding a Service
- **Deploy to cloud**
- Testing

Deploy to cloud

You need a free Heroku account and the Heroku toolbelt installed

- One simple command to package your application in “production” mode and deploy it to Heroku

Just as easy to deploy to Cloud Foundry and AWS

yo lab> Deploy to cloud

~20 minutes

Lab instructions

<https://github.com/squeed-com/jhipsterhol/wiki/Deploy-to-cloud>

JHipster hands-on lab topics

- Installation
- Generating a JHipster application
- Creating entities and relations
- Customize the user interface
- Adding a Service
- Deploy to cloud
- **Testing**

Testing

JHipster generates an extensive set of tests for each generated application.

- Integration tests using the Spring Test Context framework
- UI tests ([Karma.js](#))
- Performance tests ([Gatling](#))

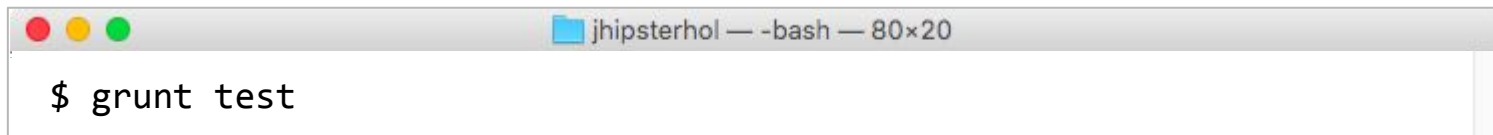
You need to write your own tests to test your business logic

It is a best practice to validate the generated code by running all tests

Karma - runs unit tests on your JavaScript code

Makes your test-driven development easy, fast, and fun. You should use it when:

- You want to test code in real browsers.
- You want to test code in multiple browsers (desktop, mobile, tablets, etc.).
- You want to execute your tests locally during development.
- You want to execute your tests on a continuous integration server.
- You want to execute your tests on every save.

A screenshot of a terminal window. The title bar shows a folder icon, the text 'jhipsterhol', and '- bash - 80x20'. The terminal content shows a prompt '\$' followed by the command 'grunt test' and a cursor.

```
jhipsterhol — -bash — 80x20  
$ grunt test
```

Gatling



Gatling is a highly capable load testing tool.

Testing workflow:

- Record scenario
- Edit scenario
- Execute
- Analyze

```
jhipsterhol — -bash — 80x20
$ mvn gatling:execute
```

yo lab> Testing

~20 minutes

Lab instructions

<https://github.com/squeed-com/jhipsterhol/wiki/Testing>

EOL (End Of Lab) - what we have done

In this hands-on-lab you ~~will~~ **have** learned **ed** how to leverage and love the JHipster stack, combining best of breed Java backend with a modern single-page application (SPA) JavaScript frontend and the tooling to get you and your team productive.

I have ~~We will~~ developed **ed** a full stack application, from scratch, and deployed it in the cloud.

And there is more...

- [Internationalization](#)
- Pagination
- Validation
- [DTOs](#)
- [Elasticsearch](#)
- [Websockets](#)
- Less & Sass
- [Modules](#)
- [JHipster-UML](#)
- Thymeleaf
- Caching
- [NoSQL](#)
- Social login

During the past hours you have hipped yourself with

GATLING
AKKA HEROKU
YEOMAN BOWER
NETTY NODE.JS
SPRING-BOOT NPM KARMA.JS
LIQUIBASE GRUNT
ANGULARJS SCALA
SPRING-SECURITY
PHANTOM.JS



This is the end



Q & A

References

1. [The JHipster Mini-book](#)

Thanks to Daniel Arenhage for writing a customized [presentation poll](#)

Image art from [jhipster.github.io](#) and [The JHipster Mini-book](#)