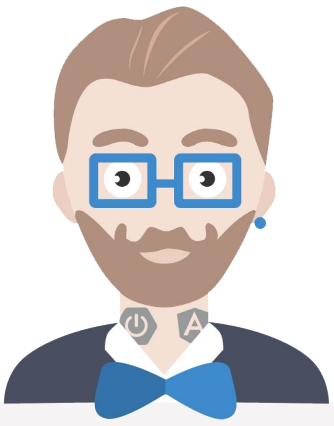
IPPON
Digital . Technologies . Hosting

WHITE PAPER



Jhipster Diary

A JOB BOARD MVP ON HEROKU IN ONE WEEK WITH JHIPSTER

SEPTEMBER
2015

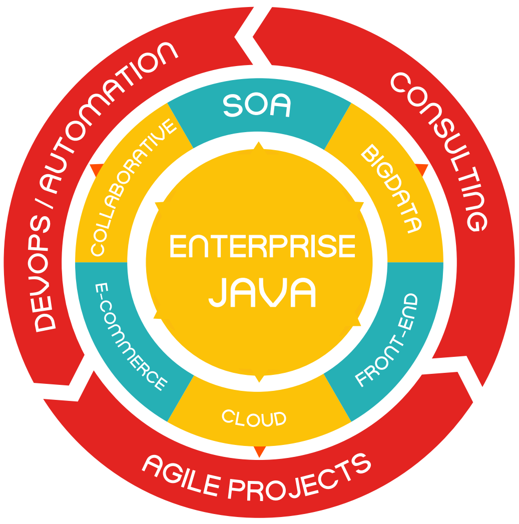
@IpponUSA - www.ipponusa.com - http://ipponusa.com/blog

1 ABOUT IPPON TECHNOLOGIES

Ippon Technologies is an IT consulting firm that specializes in Enterprise Java. Ippon engages in critical Java development projects that demand a high level of technical skills and a solid understanding of the client's business needs. Through its "End to End" service offer, Ippon is fully able to provide turnkey, cutting-edge, custom product development and consulting services, assisting companies at every stage of a complex project, including architecture, design, development, testing and release management.
Ippon's engagements span many industries such as telecommunications, banking, insurance and retail.

1.1 Our Solutions

Ippon's competencies are rooted in Enterprise Java. Along the years, Ippon has extended its functional and technical capabilities to meet the following enterprise challenges:



Modernize: overhaul strategic applications with new technologies.
Federate: implement collaborative and mobile-friendly enterprise portals.
Integrate: enable communication between systems through EAI, ETL and ESB technologies.
Broadcast: simplify access to information and data through rich and mobile user interfaces.

1.2 Contact us

FRANCE **www.ippon.fr** **accueil@ippon.fr**	Paris 43 avenue de la Grande Armée 75116 PARIS (01 46 12 48 48)	Nantes 1, Rue Du Guesclin 44000 Nantes (02 40 48 28 06)	Bordeaux 61 cours de l'Intendance 33000 Bordeaux (05 35 54 62 26)
UNITED STATES **www.ipponusa.com** **contact@ipponusa.com**	Richmond, VA 844 IPPON USA (844-477-6687)	New York, NY 844 IPPON USA (844-477-6687)	Washington 844 IPPON USA (844-477-6687)

2 LICENSE

3 INTRODUCTION

There is a lot of scaffolding work to do before a web application can be deployed. Reducing the time it takes to ship an application can have a big impact on getting feedback from your targeted user base and generating revenue faster.

JHipster aims at building the scaffolding necessary to build an enterprise ready application in just a few minutes. We want to take JHipster for a test drive and see if we can launch a prototype app in one week. To avoid the "works on my machine" syndrome we will use Heroku for our cloud deployment.

We decided to write a job board web application. In this "Minimum Viable Product", a company will be able to register and posts jobs, a recruiter will be able to login and search for applicants, and finally a user will be able to login and search for jobs.

For this project we will have 3 team members:

Preston will be the Product Owner to guide our features

Kile will be the primary (only) coder to build the app

Ben will be the Scrum Master to remove impediments and log the process.

3.1 What is JHipster?

JHipster is a Yeoman generator that aims at creating the scaffolding for a full stack java based webapp using Maven, Grunt, Bower, Spring, and AngularJS. JHipster provides an alternative to Ruby on Rails for prototyping enterprise ready applications. Let's take a look at the technologies that JHipster integrates.

Server side

The open source tech are chosen for their proven robustness, scalability and performance.

- Spring boot for easy application configuration
- Maven or Gradle for building, testing and running the application
- Spring security for hardening
- Spring MVC REST + Jackson for a RESTful web service
- Optional WebSocket support with Spring Websocket for push support
- Spring Data JPA + Bean Validation for persistence
- MySQL or Postgres for SQL based databases
- MongoDB for document oriented NoSQL databases
- Cassandra for column oriented NoSQL databases
- Liquibase for updating databases
- Elasticsearch support (currently only for SQL based databases)

Client side

The client side technologies are chosen to provide you with a fast and responsive application.

- HTML5 Boilerplate
- Twitter Bootstrap
- AngularJS
- Angular Translate for internationalization
- Optional Compass / Sass support for CSS design
- Bower for easy installation of JavaScript libraries
- Grunt or Gulp.js for building, optimization and live reloading
- Karma and PhantomJS for testing

Production / Operations

- Metrics for monitoring
- ehcache for local cache or hazelcast for distributed cache
- hazelcast for optional HTTP session clustering
- Logback for log management, configurable at runtime
- HikariCP for connection pooling

4 PROTOTYPE JOURNAL

We mentioned previously that JHipster aims at ramping up a web application from concept to prototype rapidly while using production ready technologies; let's put it to the test. The following is a developer journal that details our experience with a JHipster app.

DAY 1

Writing User Stories

We sat down with our product owner and had a storyboarding exercise to generate a backlog. He was overly ambitious and identified a set of features that would be impossible to develop in one week, so we prioritized to get the best prototype we could in that one week. Here's an overview of our backlog, broken down in epics:

- Candidate track: As a candidate I want to log in, build a profile and be able to search and apply to jobs easily.

- Recruiter track: As a recruiter I want to log in, build a personal and company profile and be able to posts jobs, and view the applicants for the job. I also want to be able to search the candidate pool to find a good match that has not applied.

- Sysadmin track: As a Sysadmin I want to be able to monitor the status of the server, the status of the application's services, be able to to interact with the REST API, and change the log levels from a browser.

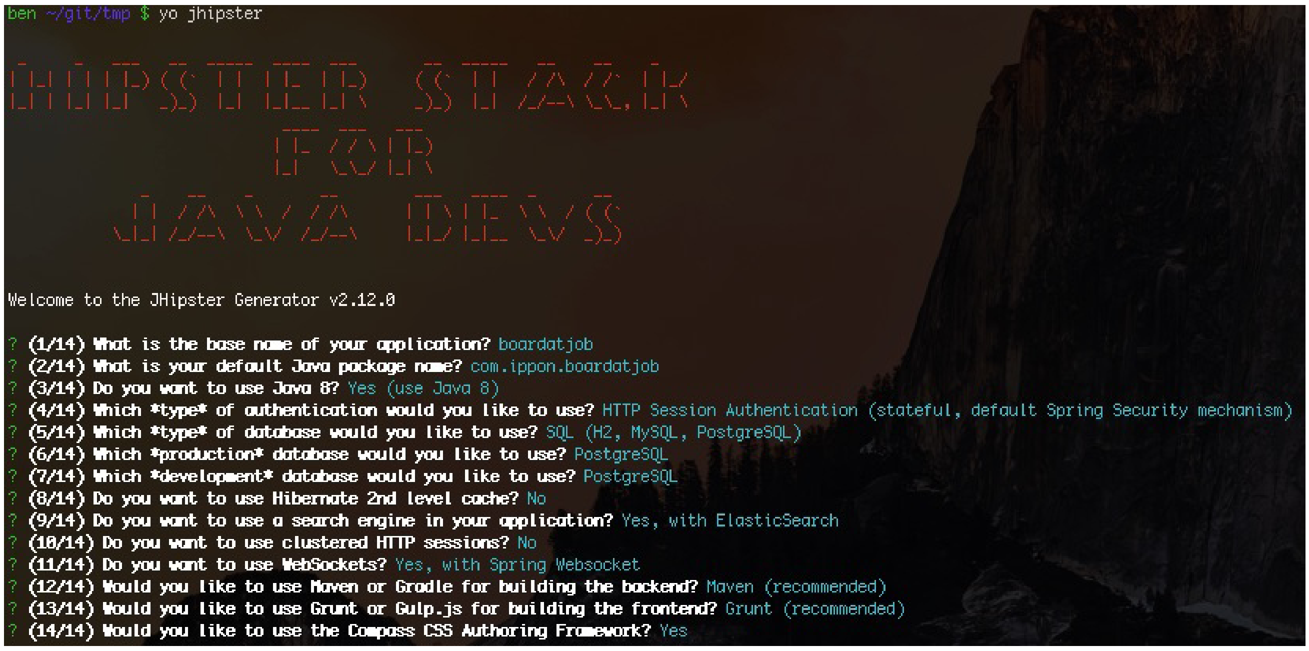
- Partner track: As a partner I want to be able to pull candidates, jobs, companies using a REST API.

- Scrum Master: As a Scrum Master I want to have code coverage tools and static analysis done on the project.

Any app ready for production has to be concerned about security. JHipster uses Spring security so we get support for authentication and authorization, protection from session fixation, clickjacking, cross site request forgery, etc.

Generating the base project

Let's generate the base project. JHipster will ask a set of questions that will customize what gets generated. Here's how we answered the questions:

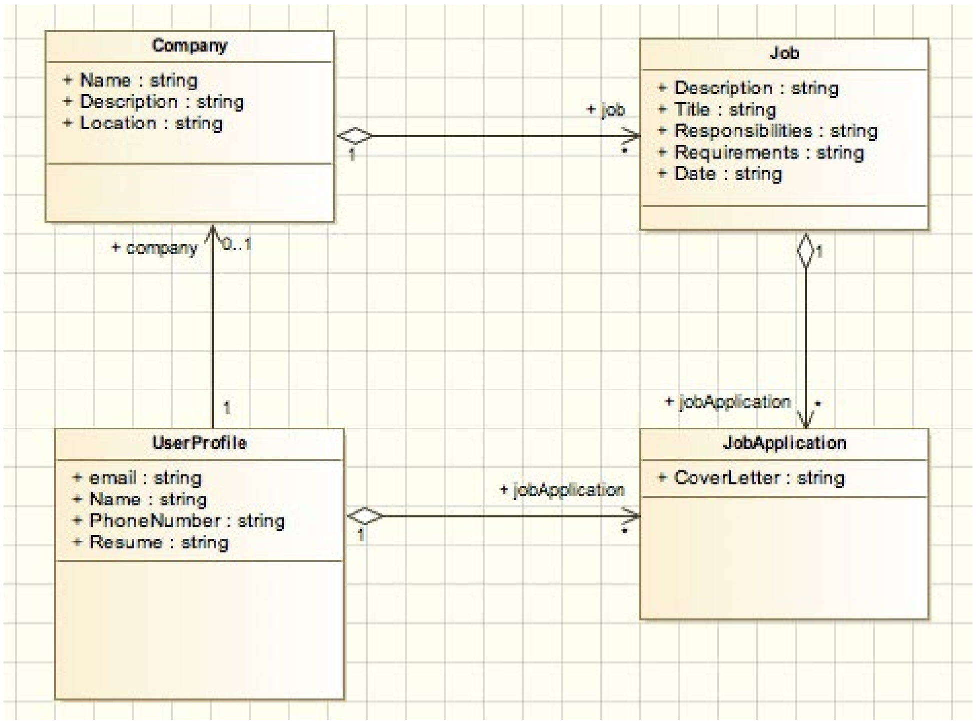


```
ben ~/git/tmp $ yo jhipster

Welcome to the JHipster Generator v2.12.0

? (1/14) What is the base name of your application? boardatjob
? (2/14) What is your default Java package name? com.ippon.boardatjob
? (3/14) Do you want to use Java 8? Yes (use Java 8)
? (4/14) Which *type* of authentication would you like to use? HTTP Session Authentication (stateful, default Spring Security mechanism)
? (5/14) Which *type* of database would you like to use? SQL (H2, MySQL, PostgreSQL)
? (6/14) Which *production* database would you like to use? PostgreSQL
? (7/14) Which *development* database would you like to use? PostgreSQL
? (8/14) Do you want to use Hibernate 2nd level cache? No
? (9/14) Do you want to use a search engine in your application? Yes, with ElasticSearch
? (10/14) Do you want to use clustered HTTP sessions? No
? (11/14) Do you want to use WebSockets? Yes, with Spring Websocket
? (12/14) Would you like to use Maven or Gradle for building the backend? Maven (recommended)
? (13/14) Would you like to use Grunt or Gulp.js for building the frontend? Grunt (recommended)
? (14/14) Would you like to use the Compass CSS Authoring Framework? Yes
```

As we selected a Postgres database, we had to install it on our development machine, add a user named boardatjob and a table named boardatjob. We also needed to grant permissions to the boardatjob user to create tables on the boardatjob database:

psql $ grant all on schema public to boardatjob;

We're now ready to generate the entities needed for our project. With JHipster you can add entities one by one and answer a set of questions for each entity, or run a tool, jhipster-uml, that based on a UML diagram will generate the entities for you. The following image shows the UML diagram we created using Modelio. JHipster requires you to export your model in an XMI format, than to feed it to the jhipster-uml tool from the project root.

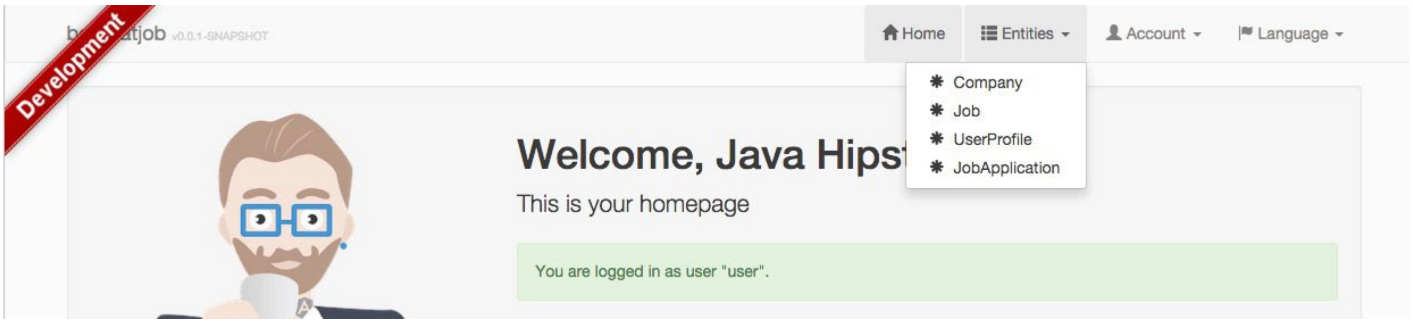


We now have a base project fully generated that we can run locally:

```
$ mvn clean install
$ mvn spring-boot:run
```

Once these commands are done running, we can see our webapp at: localhost:8080.

We have a generic JHipster landing page, authentication and two default users - an admin and regular user. JHipster also generated everything in our Partner and Sysadmin track. After logging in we can see the entities:

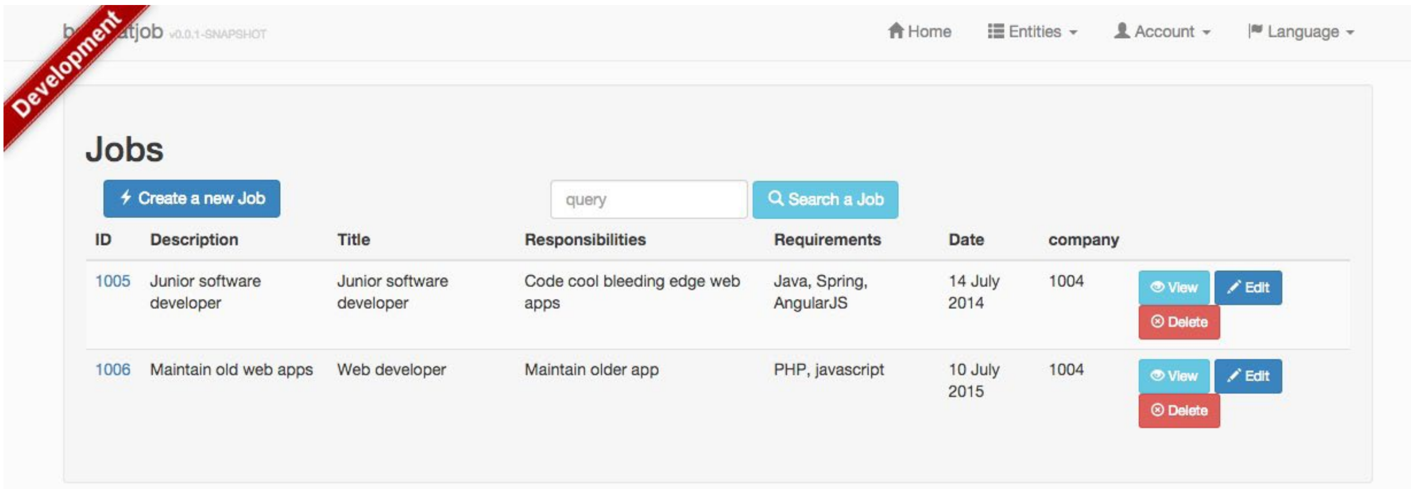


Welcome, Java Hips

This is your homepage

You are logged in as user "user".

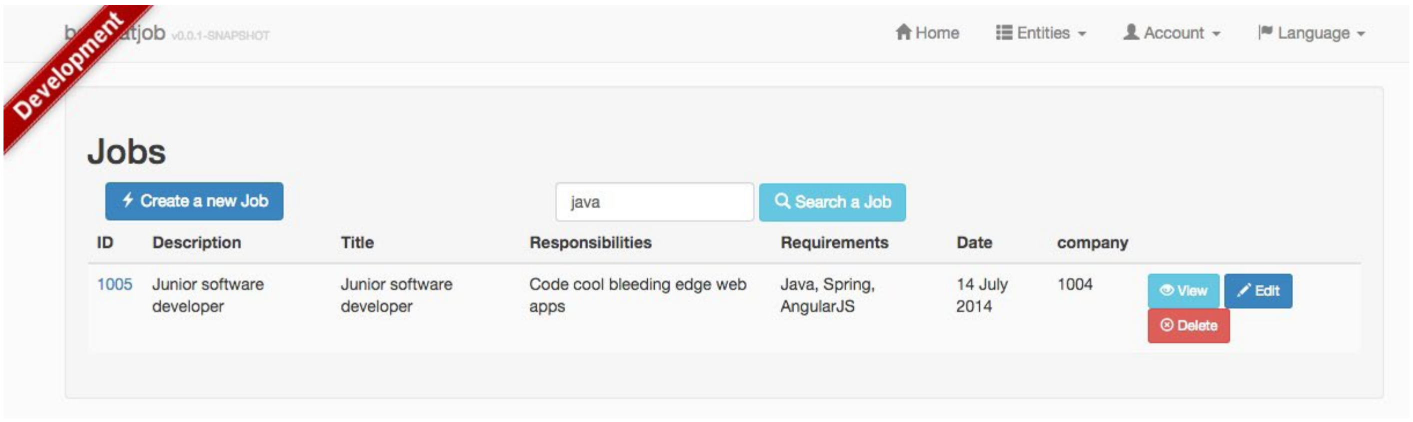
The UI is also completely connected to the back end. We can add an entity:



Jobs

ID	Description	Title	Responsibilities	Requirements	Date	company	
1005	Junior software developer	Junior software developer	Code cool bleeding edge web apps	Java, Spring, AngularJS	14 July 2014	1004	View Edit Delete
1006	Maintain old web apps	Web developer	Maintain older app	PHP, javascript	10 July 2015	1004	View Edit Delete

and search entities using elasticsearch:



Jobs

ID	Description	Title	Responsibilities	Requirements	Date	company	
1005	Junior software developer	Junior software developer	Code cool bleeding edge web apps	Java, Spring, AngularJS	14 July 2014	1004	View Edit Delete

We mentioned we wanted to be able to avoid "Works on my computer" syndrome, so we wanted to deploy to Heroku immediately to see how it ran:

We created a user account, added the free postgres database plan and went looking for an Elasticsearch addon. This is where we hit our first problem. JHipster uses Spring Data Elasticsearch which connects to an Elasticsearch server using Native Java API versus REST. The only compatible addon costs $67 a month, and we were not ready to pay this much for a demo app. We outsourced the Elasticsearch to one of our local servers. We were now ready to deploy.

First we needed to install the heroku CLI, then we logged in and deployed:

```
$heroku login
$yo jhipster:heroku
```

This command only needs to be run once, for subsequent deployment we used:

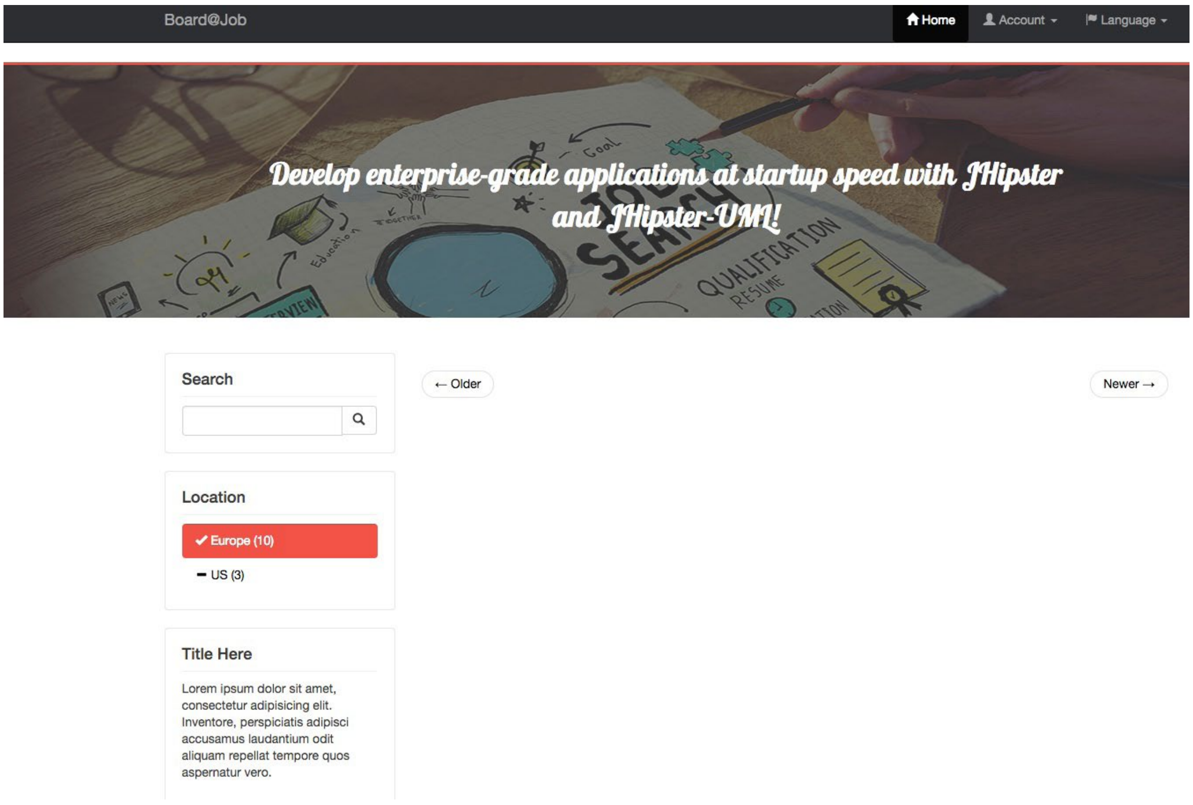
```
$mvn clean install -Pprod
$heroku deploy:jar --jar target/boardatjob.war
```

We were now ready to customize the generated app to actually make it do what we need. First order of business was to remove the default look and feel.

UI Style changes:

As our team consists of a scrum master and an architect with limited artistic capabilities, we recruited some help from one of our talented designers to give us a look and feel for the site. We asked her to build the layout and design on top of the Bootstrap CSS framework in order to easily integrate it with JHipster. Once we received the design, integrating it with the generated application was very simple.

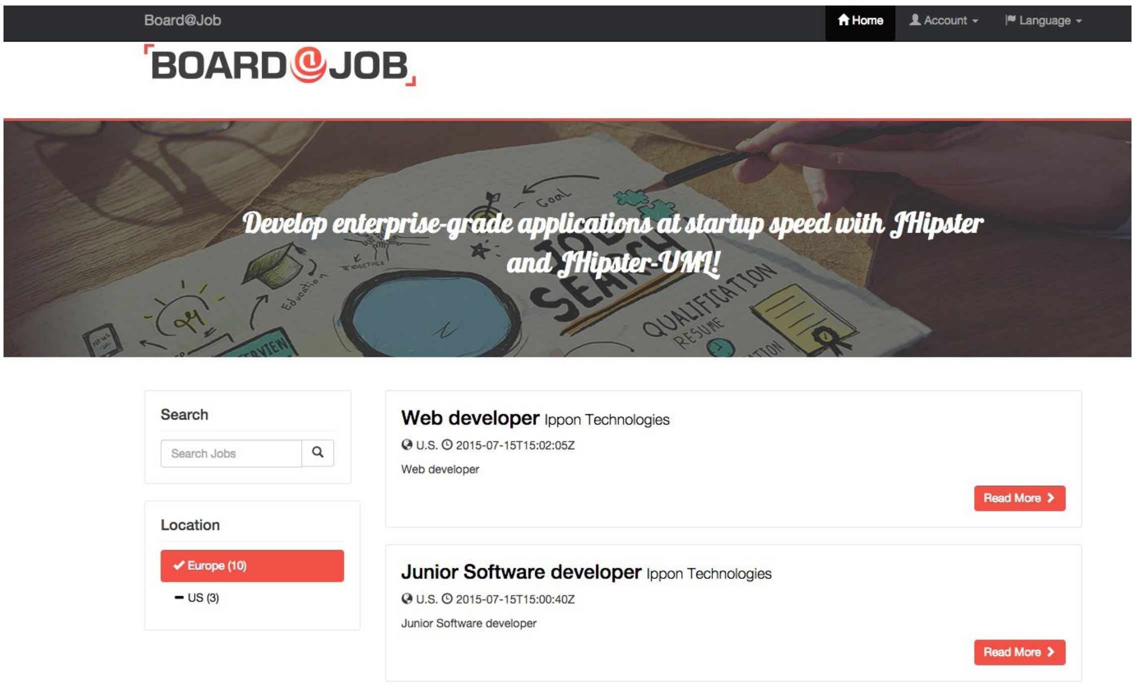
As we chose to use the Compass CSS Authoring Framework with our application, we simply replaced the contents of the main.scss file with the custom CSS provided by our designer, modified some paths to images, and added the images to the images folder. We copied the home page content into our main.html template and our site had its facelift. We then separated the common elements into the navbar template and a new header template for modularity.


Board@Job
Home
Account
Language
Develop enterprise-grade applications at startup speed with JHipster and JHipster-UML!
Search
← Older
Newer →
Location
✔ Europe (10)
US (3)
Title Here

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Inventore, perspiciatis adipisci accusamus laudantium odit aliquam repellat tempore quos aspernatur vero.

DAY 2

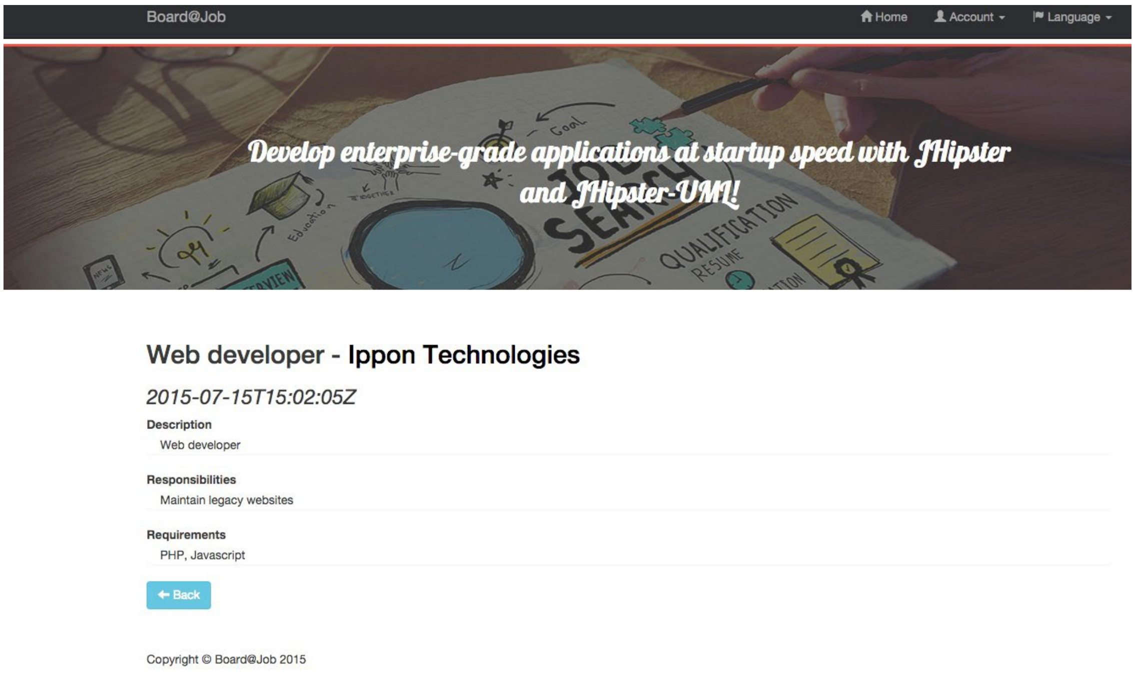
This day we tackled the landing page and started using the user roles that AngularJS provides. Now only an admin can add entities, and when a user logs in they are greeted with a landing page with a list of posted jobs, with a search box.



Web developer Ippon Technologies
U.S. 2015-07-15T15:02:05Z
Web developer

Junior Software developer Ippon Technologies
U.S. 2015-07-15T15:00:40Z
Junior Software developer

The listings are clickable to view more details:


Board@Job
Home
Account
Language
Develop enterprise-grade applications at startup speed with JHipster and JHipster-UML!
Back

Web developer - **Ippon Technologies**

2015-07-15T15:02:05Z

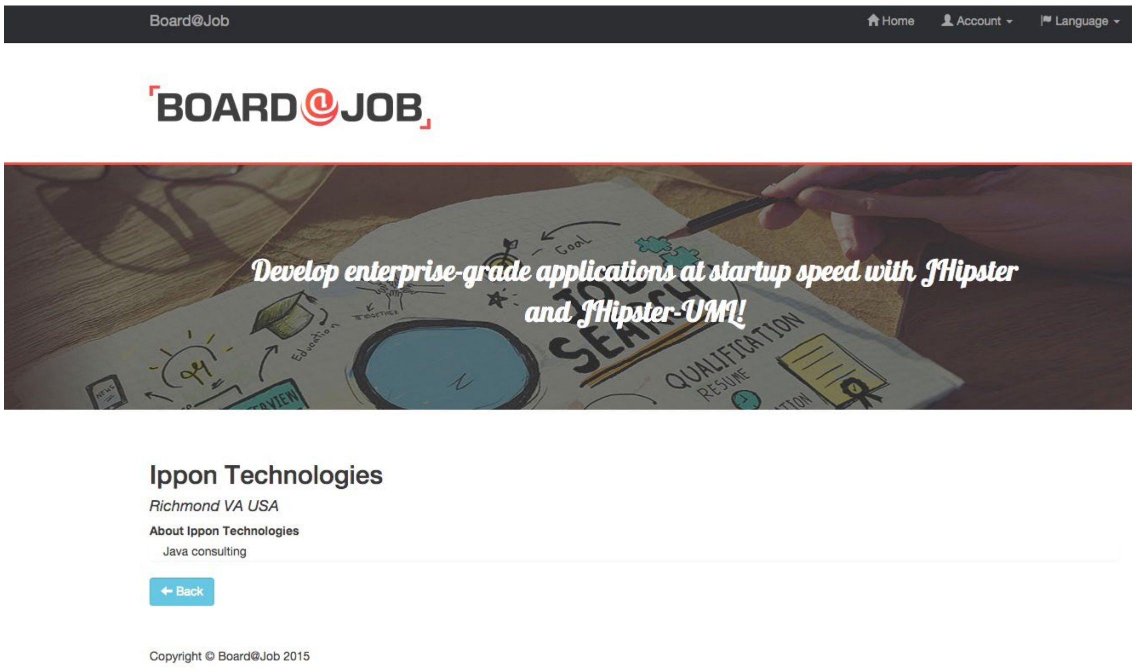
Description
Web developer

Responsibilities
Maintain legacy websites

Requirements
PHP, Javascript

Copyright © Board@Job 2015

and the company name also has a detail page now:


Board@Job
Home
Account
Language
BOARD@JOB
Develop enterprise-grade applications at startup speed with JHipster and JHipster-UML!
Back

Ippon Technologies
Richmond VA USA
About Ippon Technologies
Java consulting

Copyright © Board@Job 2015

On the second day, we were fortunate enough to get started on our first feature development tasks. This included adding a new listing template for the front page, tying in Elasticsearch, and modifying the job and company detail pages. This required little to no back-end development, as we were just moving publicly available information around in the application. Most of the code required was HTML and JavaScript with AngularJS.

We added some minor back-end code to sort the listings by post date by default. To do this, we added a date field to the job object and added the corresponding liquibase changeset. We then modified the REST controller to pass sort criteria to the generated Spring Data repository.

We added pagination (infinite scroll) to the job listing page, which is available from JHipster as an option, but we did not select it. Even though we did not select it by default, it was easy to implement by generating a temporary JHipster project with pagination and borrowing from the generated implementation.

DAY 3

Coverage and Static analysis:

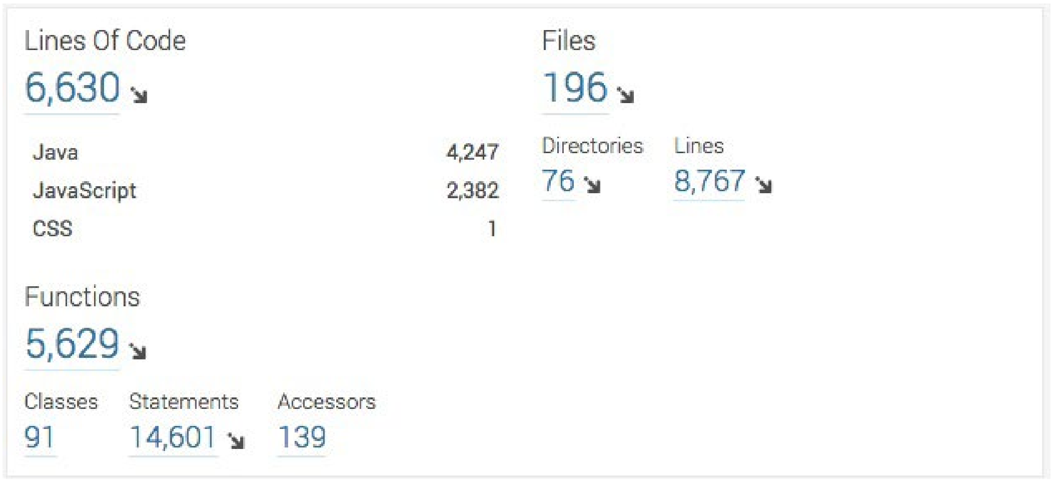
JHipster adds a basic Sonar configuration into the pom.xml, but the Sonar used was an outdated version. We updated the version and adding a few extra configuration such as ignoring bower_components files and consolidated tests reports in the same directory. We also added code coverage for Java using Jacoco and for AngularJS with Karma-coverage-reporter. These code coverage settings are standard and it would be nice to see JHipster automatically configure that for us.

You will need your own Sonar server, once setup you can run an analysis:

```
$ mvn clean install
$ grunt test
$ mvn sonar:sonar
```

With Sonar reports we can get an overall picture of what JHipster generated for us.

The project overview:



Lines Of Code
6,630 ↘

Java	4,247
JavaScript	2,382
CSS	1

Files
196 ↘

Directories
76 ↘

Lines
8,767 ↘

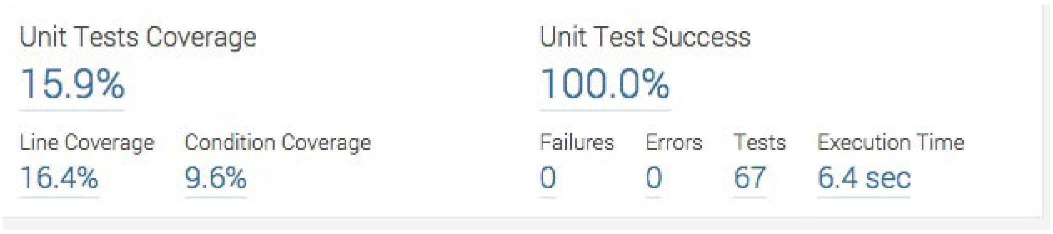
Functions
5,629 ↘

Classes
91

Statements
14,601 ↘

Accessors
139

JHipster even generates a few tests for us. As we were in prototype mode we did not add any extra tests.



Unit Tests Coverage
15.9%

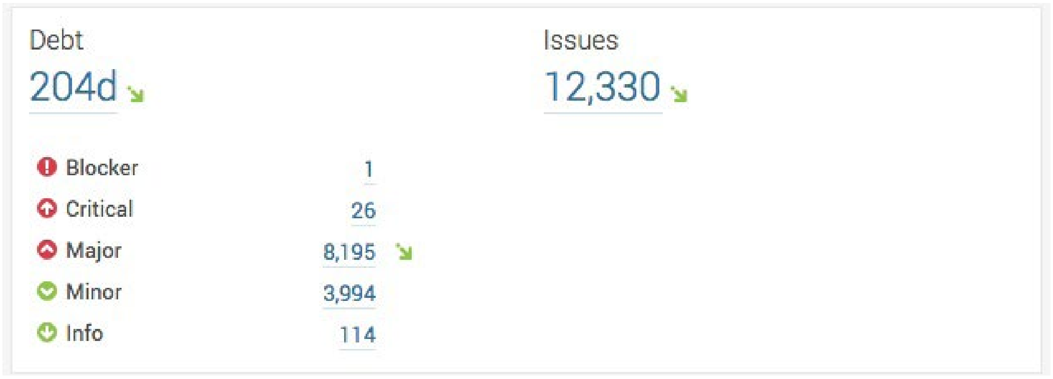
Line Coverage
16.4%

Condition Coverage
9.6%

Unit Test Success
100.0%

Failures	Errors	Tests	Execution Time
0	0	67	6.4 sec

And here is the issue overview that Sonar discovered. The Sonar quality settings were default.



Debt
204d ↘

Issues
12,330 ↘

Blocker	1	
Critical	26	
Major	8,195	↘
Minor	3,994	
Info	114	

Faceted search

Our developer hit another snag when trying to add faceted search using Elasticsearch. We currently have a UI mock for location based searches and we will revisit a solution to this problem another day.

The challenge with Elasticsearch was using aggregations to display counts by location. This is due to a combination of limited experience with Elasticsearch and little to no documentation for the Spring Data Elasticsearch library with facets/aggregations. Because of the time constraints, we decided to move on and address this feature in a later sprint.

User profile

In order to reuse the user settings component provided by JHipster, we added a user profile entity to support additional fields, only relating them by the unique login id. We modified the account registration process to create a user profile when an account is registered. We modified the settings page and Angular components to support additional fields for user profiles.

We also identified a minor JHipster bug on this day - Refreshing setting pages causes the user to lose his roles, until he leaves the page. We identified the issue with a developer on another team and he submitted a fix for the issue that day. The fix has already been accepted and implemented in JHipster.

We store the applicant resume in the user profile entity as a blob. We added a REST resource to support uploads and download requests for a given user profile id. We used an existing Angular library to support standard and drag and drop uploads.

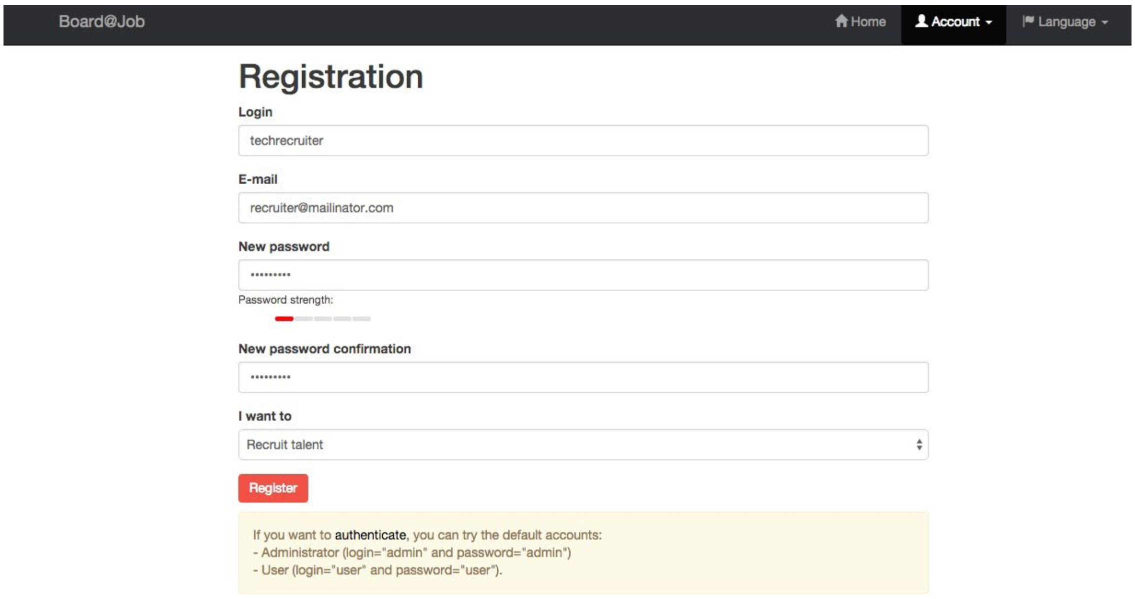
DAY 4

On day four, the final day of dedicated development, we first tackled the requirement to allow job seekers to apply for jobs.

Recruiter pages

The recruiter pages required the brunt of the work on the final day of feature development. We were able to create a company page for the recruiter. We also added the pages to allow a recruiter to submit and review job listings for their company. We then added the list of job applications for listings submitted by the recruiter to the home page. The recruiter was also given the ability to review information about the application such as the cover letter, applicant contact information, and download their resume.

Creating an account as a recruiter:



Registration

Login
techrecruiter

E-mail
recruiter@mailinator.com

New password
•••••••••

Password strength:

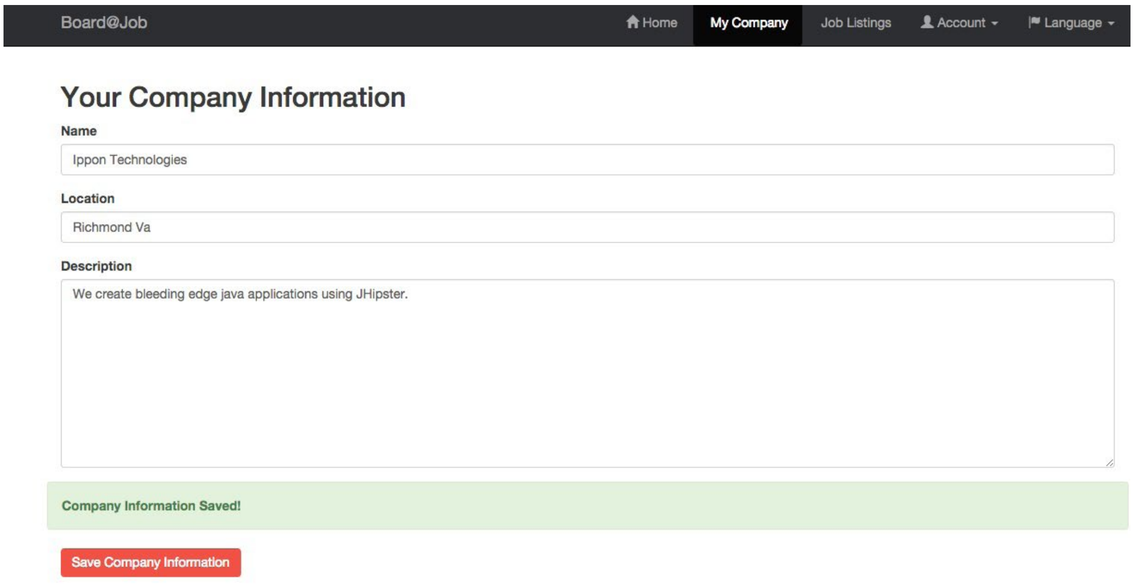
New password confirmation
•••••••••

I want to
Recruit talent

Register

If you want to **authenticate**, you can try the default accounts:
- Administrator (login="admin" and password="admin")
- User (login="user" and password="user").

Editing your company details:



Your Company Information

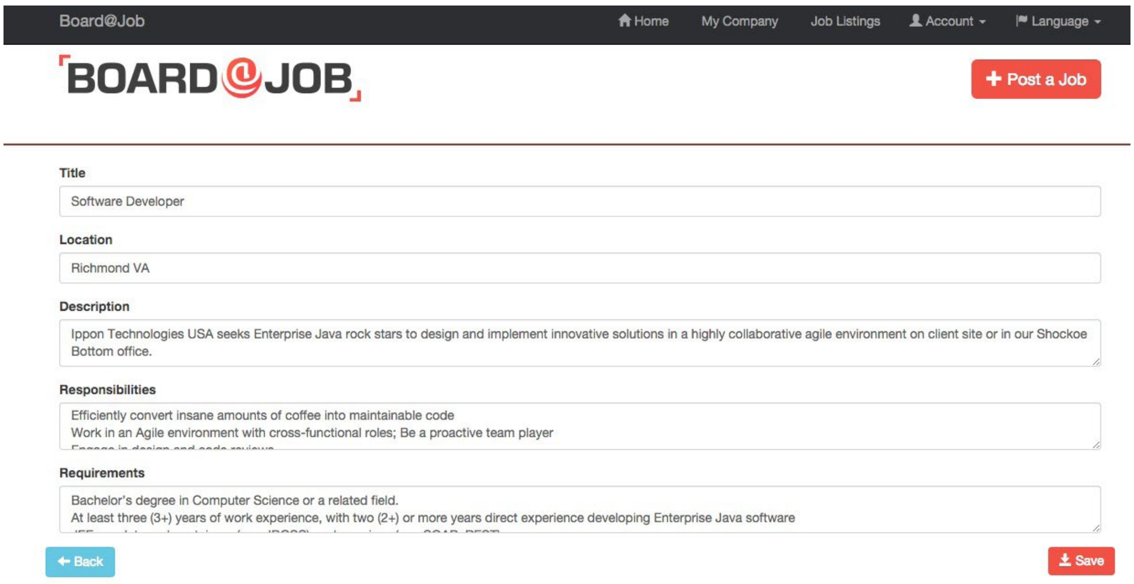
Name: Ippon Technologies

Location: Richmond Va

Description: We create bleeding edge java applications using JHipster.

Company Information Saved!

Creating a job:



Title: Software Developer

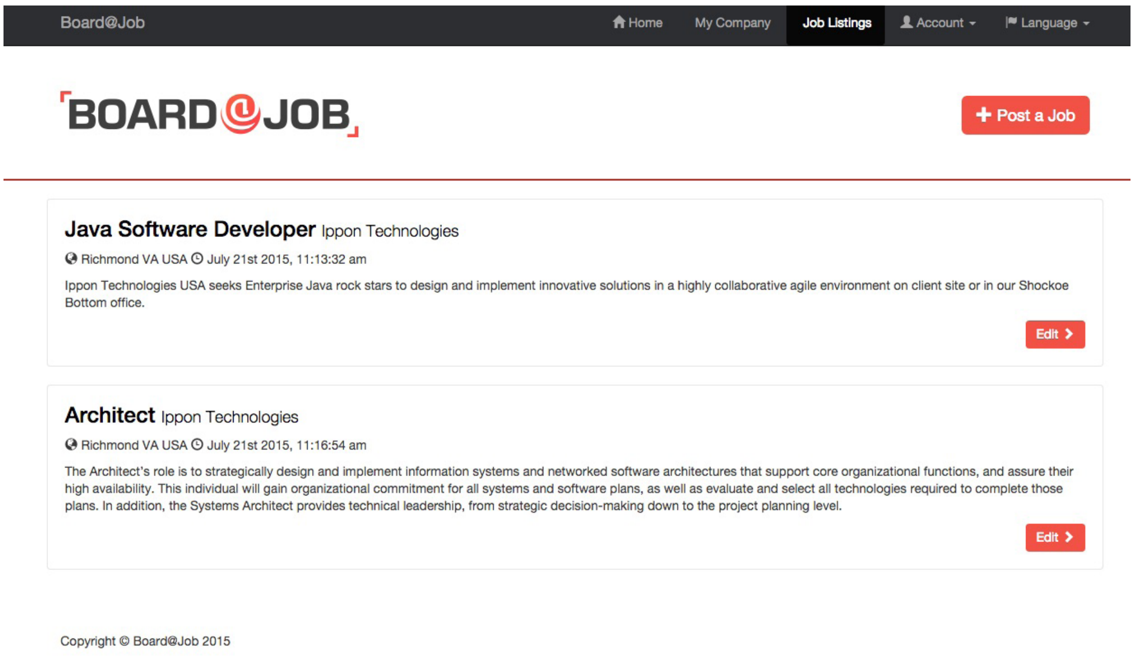
Location: Richmond VA

Description: Ippon Technologies USA seeks Enterprise Java rock stars to design and implement innovative solutions in a highly collaborative agile environment on client site or in our Shockoe Bottom office.

Responsibilities:
Efficiently convert insane amounts of coffee into maintainable code
Work in an Agile environment with cross-functional roles; Be a proactive team player

Requirements:
Bachelor's degree in Computer Science or a related field.
At least three (3+) years of work experience, with two (2+) or more years direct experience developing Enterprise Java software

As a recruiter you can see the jobs you've created:



Java Software Developer Ippon Technologies

Richmond VA USA July 21st 2015, 11:13:32 am

Ippon Technologies USA seeks Enterprise Java rock stars to design and implement innovative solutions in a highly collaborative agile environment on client site or in our Shockoe Bottom office.

Edit

Architect Ippon Technologies

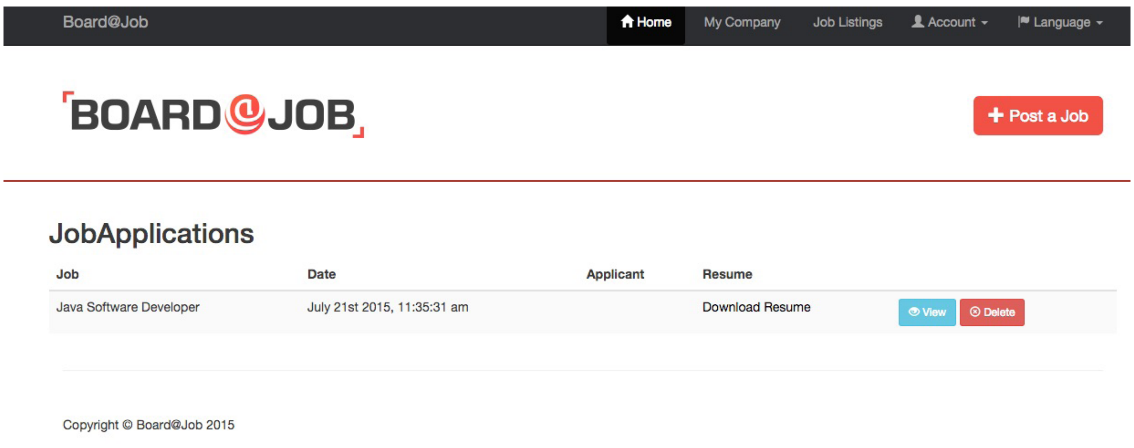
Richmond VA USA July 21st 2015, 11:16:54 am

The Architect's role is to strategically design and implement information systems and networked software architectures that support core organizational functions, and assure their high availability. This individual will gain organizational commitment for all systems and software plans, as well as evaluate and select all technologies required to complete those plans. In addition, the Systems Architect provides technical leadership, from strategic decision-making down to the project planning level.

Edit

Copyright © Board@Job 2015

And you can see the applicants:



JobApplications

Job	Date	Applicant	Resume	
Java Software Developer	July 21st 2015, 11:35:31 am		Download Resume	View Delete

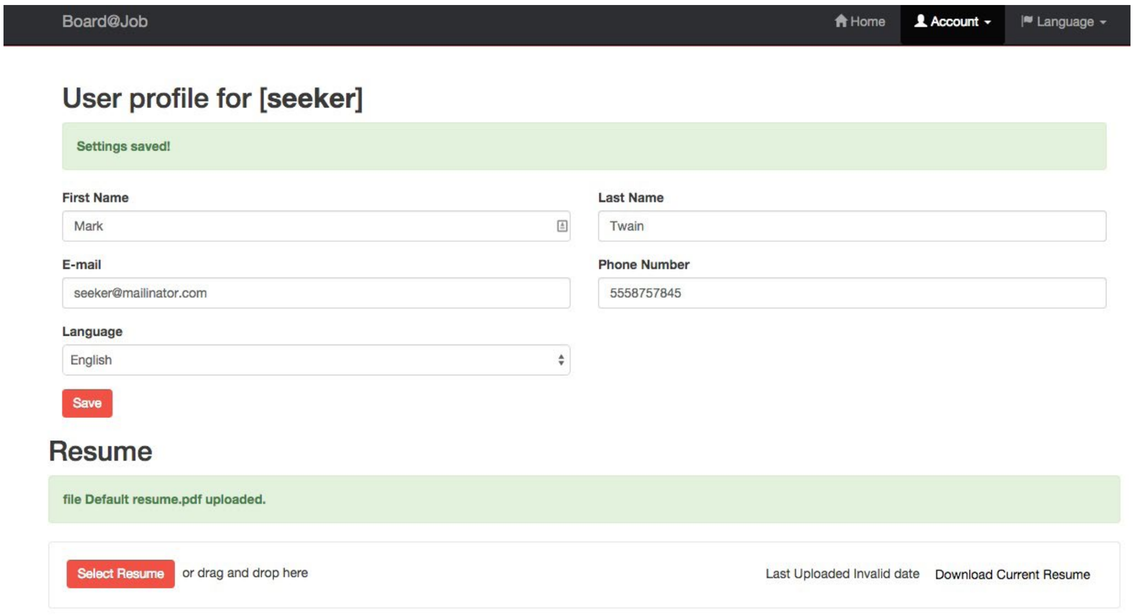
Copyright © Board@Job 2015

Unfortunately we have a bug we weren't able to fix in time where the name of the applicant doesn't get displayed.

Job Application

Using the supporting job application entity that links the user profile to a job, we added a page to allow a registered and authenticated user to fill out a cover letter and submit their application. This took very little effort, and most of the work was in the details - determining whether or not the applicant had applied for a job yet.

Creating a user profile and uploading a resume:



User profile for [seeker]

Settings saved!

First Name: Mark

Last Name: Twain

E-mail: seeker@mailinator.com

Phone Number: 5558757845

Language: English

Save

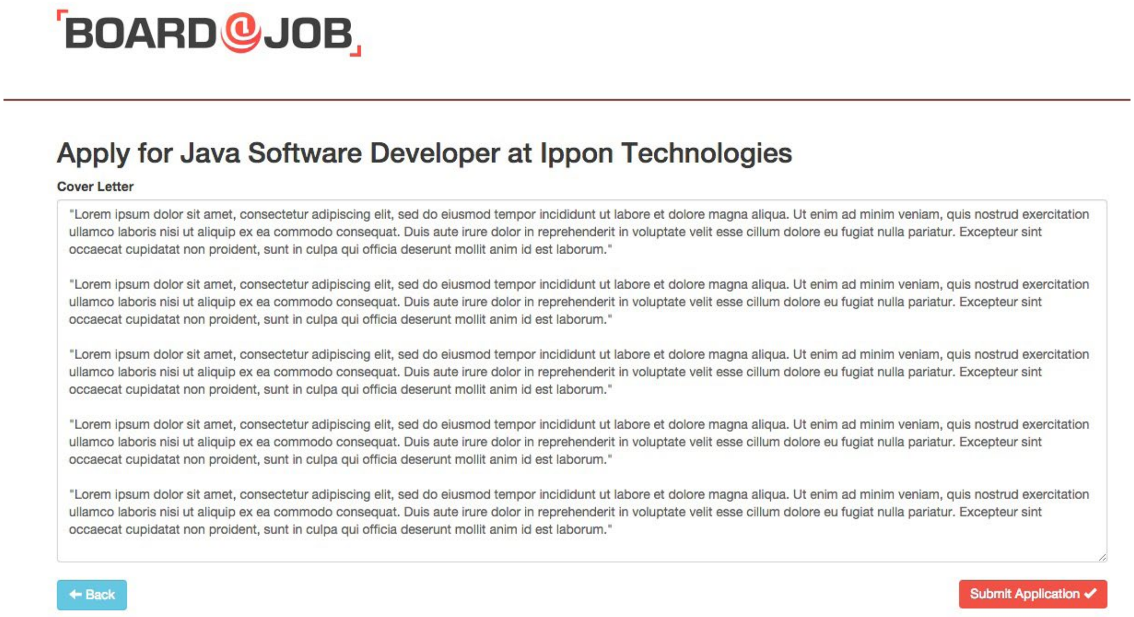
Resume

file Default resume.pdf uploaded.

Select Resume or drag and drop here

Last Uploaded Invalid date Download Current Resume

Once your profile is created, you can apply for a job:



Apply for Java Software Developer at Ippon Technologies

Cover Letter

"Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum."

"Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum."

"Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum."

"Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum."

"Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum."

DAY 5

We only had half a day to finish our sprint so we decided against doing any new features and focused on fixing bugs as we found them. A few bugs were found and fixed, and a few others remain.

Email Support

For email support, we signed up for a free SendGrid account and configured the sendmail settings in the application.yml file.

5 CONCLUSION

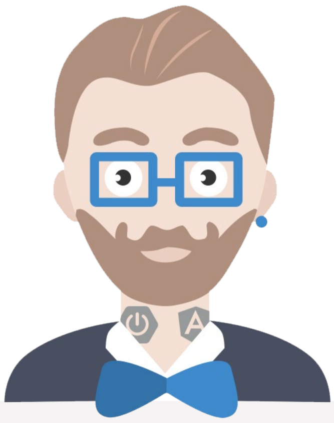
After the first day of running JHipster we haven't had to touch it again, though it can still be used to add entities at any time. The greatest part of JHipster is that it allowed us to focus immediately on what our app's purpose was. All the scaffolding was done and the app was production ready as seen with our deployment to Heroku. The generated code is well organized and so it was easy to find your way through the codebase. We easily integrated third party libraries using bower or maven.

JHipster really shines at creating a usable, scalable prototype that allows you to get immediate business feedback within the first week. You can then decide if it is worth it to spend a few months developing the full product, and since JHipster uses a stable, secure and fast technology stack you don't need to start over, you can just build on top of your prototype.

Further reading

https://jhipster.github.io/
http://jhipster.github.io/jhipster_uml.html
https://www.heroku.com/
http://www.hcltech.com/white-papers/engineering-and-rd-services/cloud-ready-web-apps-jhipster
eb-apps-jhipster

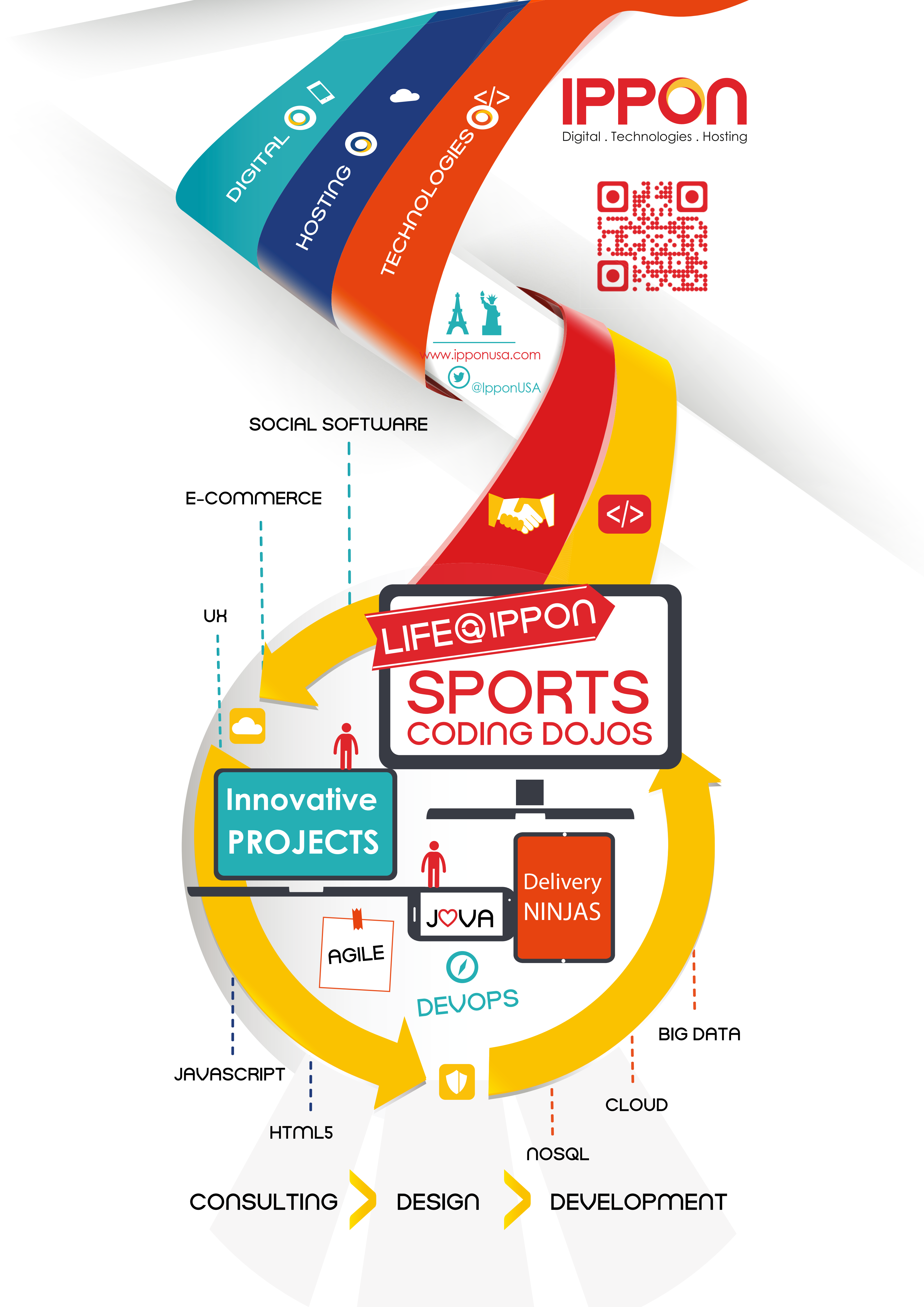
IPPON
Digital . Technologies . Hosting



Jhipster Diary

A JOB BOARD MVP ON HEROKU IN ONE WEEK WITH JHIPSTER

FRANCE **www.ippon.fr** **accueil@ippon.fr**	Paris 43 avenue de la Grande Armée 75116 PARIS (01 46 12 48 48)	Nantes 1, Rue Du Guesclin 44000 Nantes (02 40 48 28 06)	Bordeaux 61 cours de l'Intendance 33000 Bordeaux (05 35 54 62 26)
UNITED STATES **www.ipponusa.com** **contact@ipponusa.com**	Richmond, VA 844 IPPON USA (844-477-6687)	New York, NY 844 IPPON USA (844-477-6687)	Washington 844 IPPON USA (844-477-6687)


IPPON
Digital . Technologies . Hosting
DIGITAL
HOSTING
TECHNOLOGIES
www.ipponusa.com
@IpponUSA
SOCIAL SOFTWARE
E-COMMERCE
UX
LIFE@IPPON
SPORTS
CODING DOJOS
Innovative
PROJECTS
Delivery
NINJAS
JAVA
AGILE
DEVOPS
JAVASCRIPT
HTML5
BIG DATA
CLOUD
NOSQL
CONSULTING
DESIGN
DEVELOPMENT