

## Modern Java web development

Thomas Kruse

---

- Thomas Kruse
  - Consultant
  - Leader JUG Münster
  - @everflux on twitter

## Social App (not only) for JUGs

- Twitter
- Facebook
- ...
- Keeping credentials centralized
- Timed publications
- ...
- Showcase Netbeans ;-)
- And modern Java

Illustrate architecture decision making

## Environment

- More complex systems
- Mobile devices
- New major players / trend setters

## Architecture requirements

- Decoupled front-end/back-end
- Low latency
- Manage front-end complexity
- Leverage the cloud

## User expectations

- Low latency
- Cross-device
- Ease of use
- Time to market
- Long lasting systems

# FRONTEND OPTIONS



**Java Swing /  
JavaFX**

- No production quality cross device
- Requires Java on user device
- Valid for special use cases



**JSF**

- Tight backend coupling
- High resource consumption
- R.I.P.!
- ... same for Wicket, Vaadin, ...



**HTML5+  
REST**

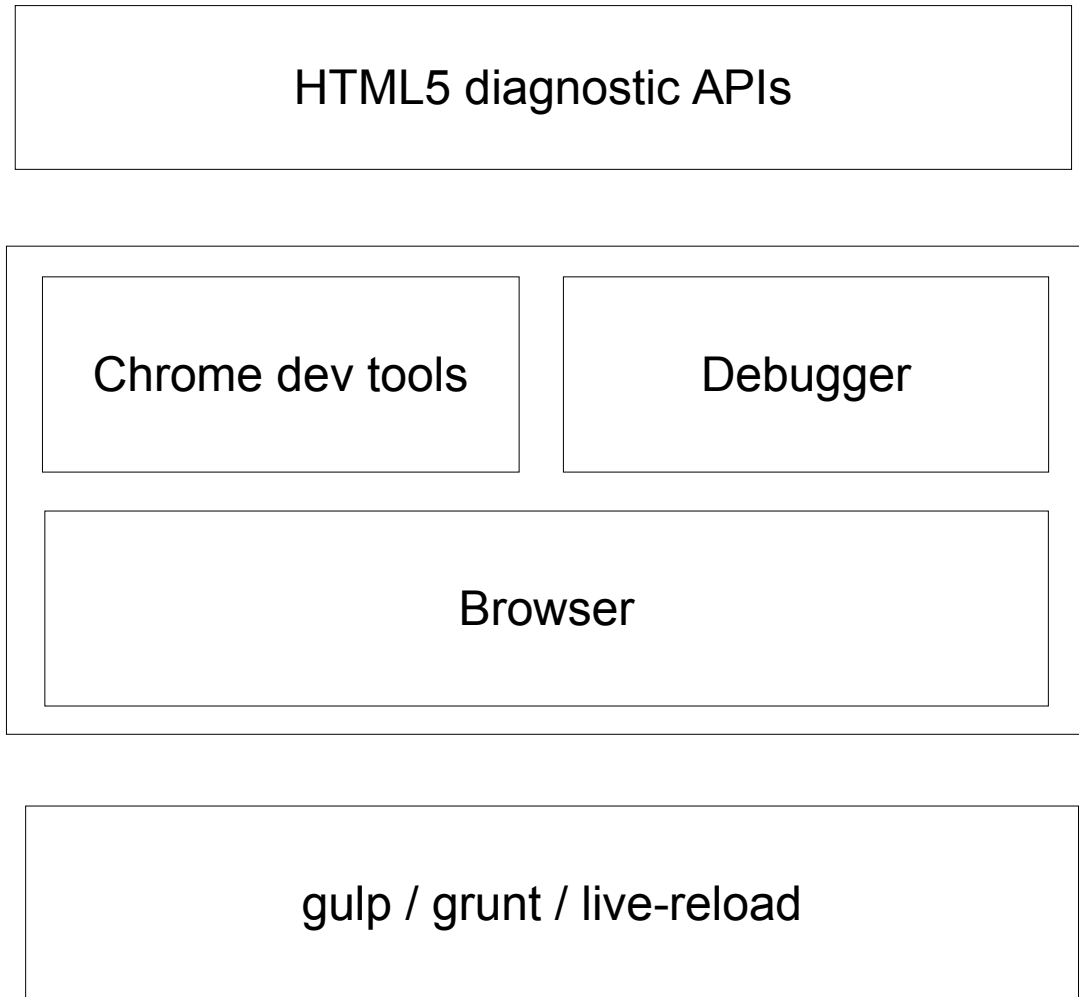
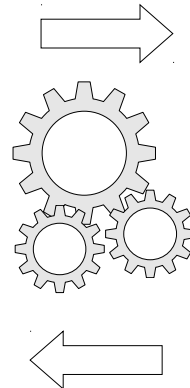
- Server side rendering or ...
- ... HTTP API driven
- Ajax, Websockets, SSE, ...
- Vanilla or Framework?
- jQuery: Already legacy





IDE

- Code-completion
- Navigation
- Hints
- Round-trip integration
- Instant Live-reload
- Debugging / breakpoints
- gulp / grunt / npm



- Netbeans round-trip editing

## Front-End

- Finer grained
- Parallel
- Out-of-order
- Composition of multiple APIs
- Same API for M2M

## Back-End

- Coarse grained (JSF, MVC)
- Sequential
- Single-Threaded
- Container services
  - Security
  - Transactions
- Separate API for M2M



# PROJECT STRUCTURE



## Single project

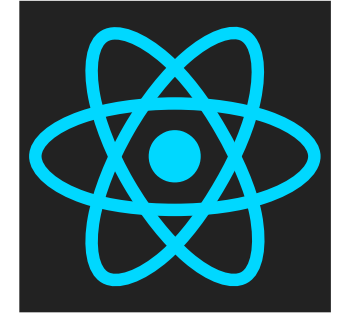
## Separate front end project

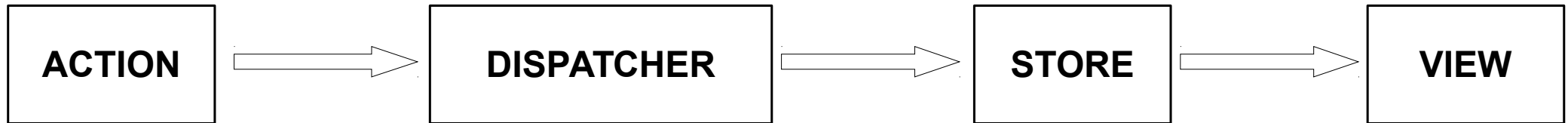
Dependency	<ul style="list-style-type: none"><li>• WebJars</li><li>• Manual – add to SCM</li></ul>	<ul style="list-style-type: none"><li>• npm</li><li>• bower</li></ul>
Build system	<ul style="list-style-type: none"><li>• Maven</li><li>• Gradle</li></ul>	<ul style="list-style-type: none"><li>• grunt</li><li>• gulp</li></ul>
Coupling	<ul style="list-style-type: none"><li>• Highly coupled</li><li>• Pace of evolution mismatch</li></ul>	<ul style="list-style-type: none"><li>• Loosely coupled</li><li>• Enables separate evolution</li><li>• Encourages testability</li><li>• Easy replacement of one part</li></ul>
Deployment	<ul style="list-style-type: none"><li>• Part of application</li><li>• Simultaneous release</li></ul>	<ul style="list-style-type: none"><li>• Separate</li><li>• Low requirements for plain HTML</li></ul>

## Why a front-end framework?

- jQuery is dead
- ... and was never a framework
- Testability
- Ease of development
- Team scaling
- Even outsourcing

- Facebook
  - ... and others
- React.js
  - Library
  - (virtual) DOM manipulation
  - Unidirectional data-flow concept
  - Server-side rendering possible
- No framework!
  - HTTP Service - superagent

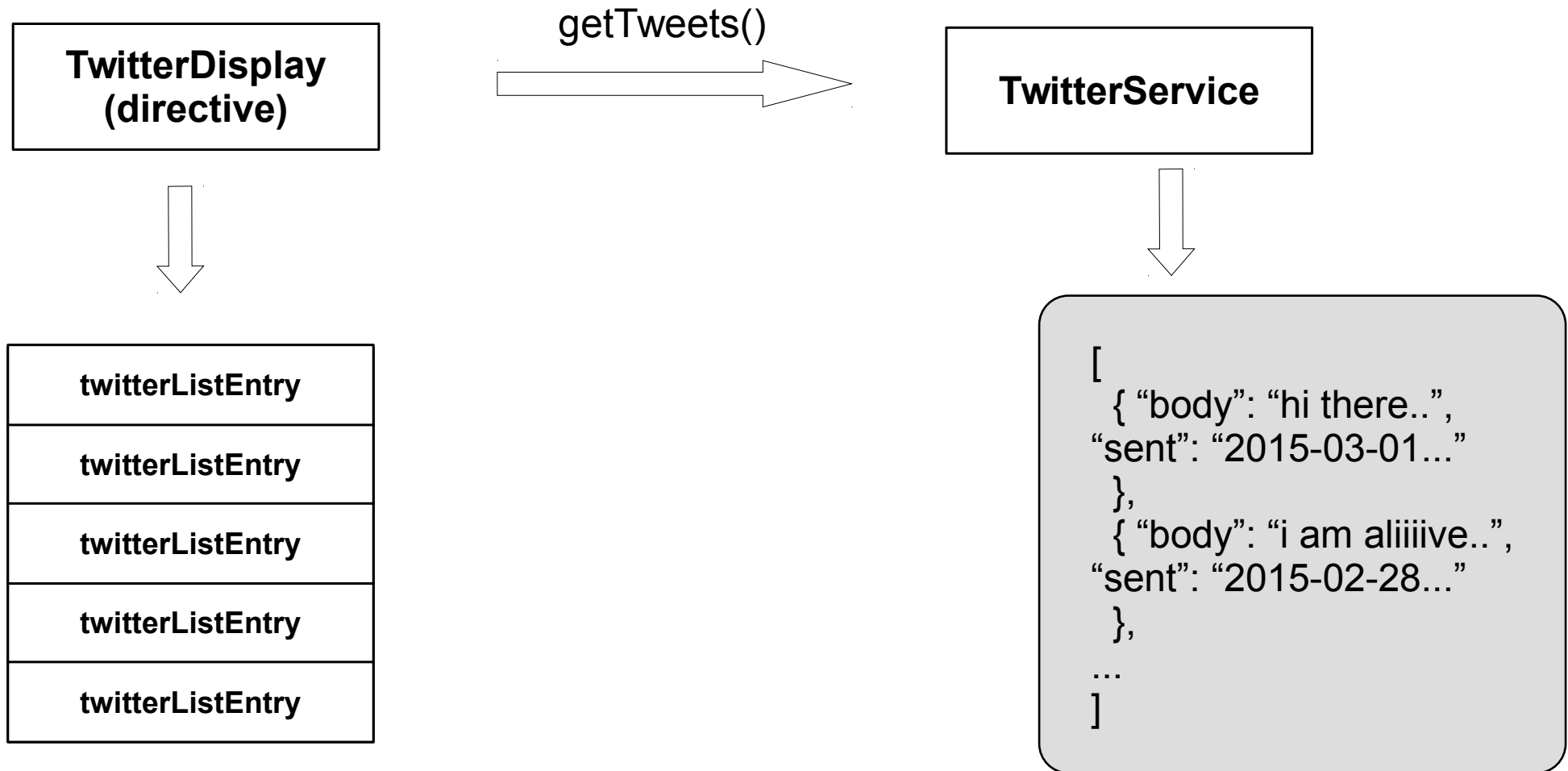




- View components
  - JSX (optional, no IDE support)
  - Properties (immut.) input
- Dispatcher
  - Receives actions, notifies callbacks
- Store
  - Container for state and callbacks
- Action
  - Passed to dispatcher (payload)



- Front-end framework
- Opinionated
- Dependency injection
- Model-View-ViewModel
- Two-way data binding
- Routing
- Directives, Services
- Java developers feel at home




- Angular message service – fake or real

- Mostly implemented across relevant platforms

- Elements

- Templates
- HTML import
- Custom elements
- Shadow DOM



	CHROME	OPERA	FIREFOX	SAFARI	IE
	■	■	■	■	■
	■	■	■	■	■
	■	■	■	■	■
	■	■	■	■	■

- Foundation for libraries like x-tag / Polymer

- Polyfill for IE / stable spec: Bosonic



# DEMO

- Netbeans web-component editing

# BACKEND LANGUAGE

---

TRION

Java

- Java EE
- Spring Boot
- dropwizard
- Fancy other options like vert.x



PHP / Ruby /  
Python

- Sometimes a good choice
- Consider tooling and operations



node / io.js

- All developers use JavaScript anyway
- Same language on all layers
- Consider evolution speed
- Avatar: Uncertain future



go

- Sweet spot for JSON micro services



# OUR STACK



## Backend

## Frontend

Tooling	<ul style="list-style-type: none"><li>• Netbeans IDE</li><li>• curl</li></ul>	<ul style="list-style-type: none"><li>• Netbeans IDE</li><li>• Chrome / Firefox</li></ul>
Build system/ Dependency mgmt	<ul style="list-style-type: none"><li>• Maven</li></ul>	<ul style="list-style-type: none"><li>• npm / bower</li><li>• grunt</li></ul>
Frameworks	<ul style="list-style-type: none"><li>• Spring Boot</li><li>• Spring Integration</li></ul>	<ul style="list-style-type: none"><li>• Foundation CSS</li><li>• Angular</li><li>• Angular Foundation</li></ul>
Runtime	<ul style="list-style-type: none"><li>• Tomcat</li></ul>	<ul style="list-style-type: none"><li>• Browser</li></ul>

**jhipster**

- Single project
- Maven
- Spring Boot
- Angular



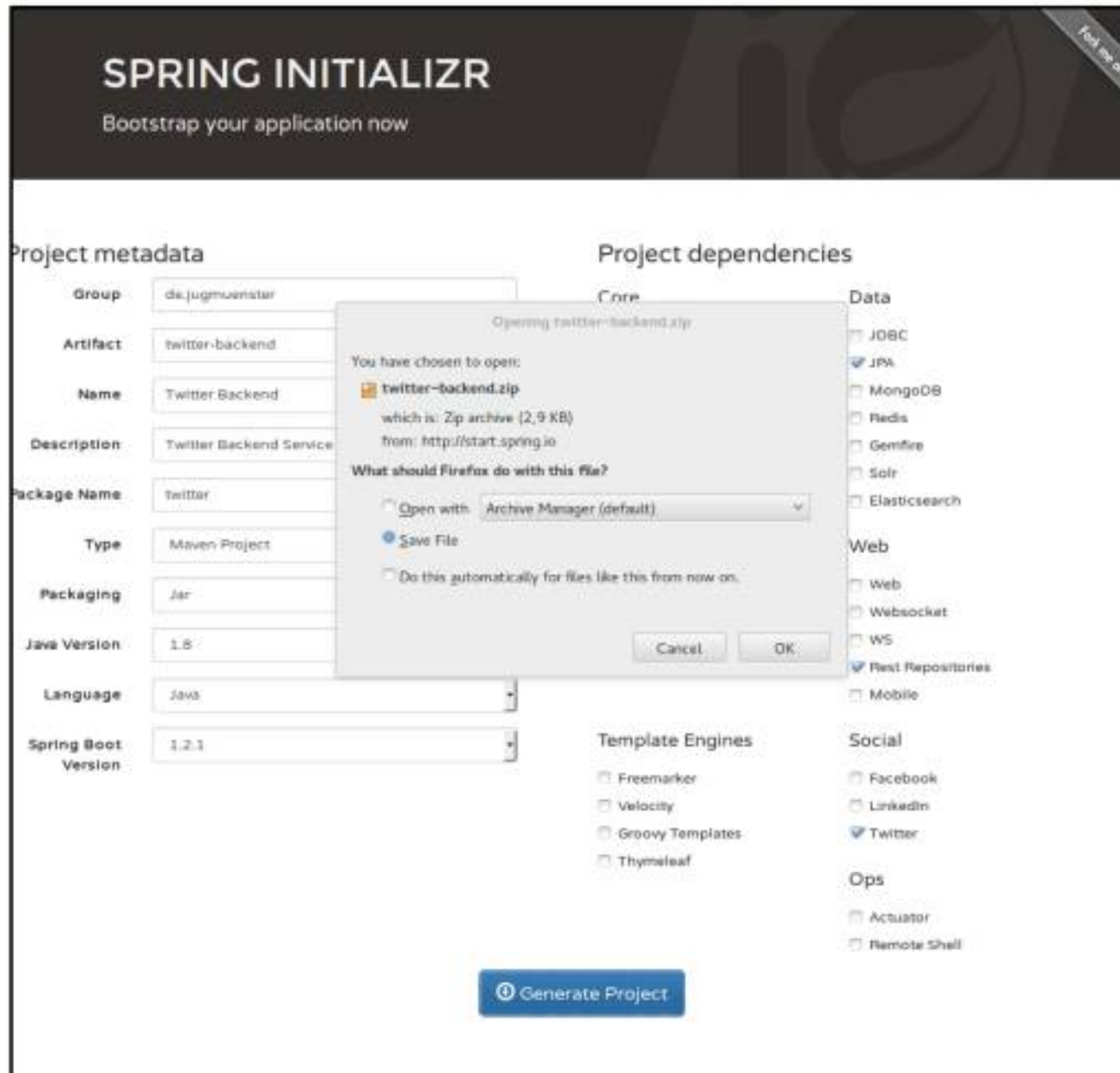
**spring initializr**

- Code generation maintained by spring
- Maven + Gradle
- Cloud service



**maven  
archetype**

- No broad support
- Hard to implement
- Often not well maintained
- One time scaffolding



**SPRING INITIALIZR**  
Bootstrap your application now

**Project metadata**

- Group: de.jugmuenster
- Artifact: twitter-backend
- Name: Twitter Backend
- Description: Twitter Backend Service
- Package Name: twitter
- Type: Maven Project
- Packaging: Jar
- Java Version: 1.8
- Language: Java
- Spring Boot Version: 1.2.1

**Project dependencies**

- Core
  - Core
- Data
  - JDBC
  - JPA
  - MongoDB
  - Redis
  - Gemfire
  - Solr
  - Elasticsearch
- Web
  - Web
  - Websocket
  - WS
  - Rest Repositories
  - Mobile
- Social
  - Facebook
  - LinkedIn
  - Twitter
- Ops
  - Actuator
  - Remote Shell
- Template Engines
  - Freemarker
  - Velocity
  - Groovy Templates
  - Thymeleaf

**File Download Dialog:**

Opening twitter-backend.zip

You have chosen to open:  
twitter-backend.zip  
which is: Zip archive (2.9 KB)  
from: http://start.spring.io

What should Firefox do with this file?

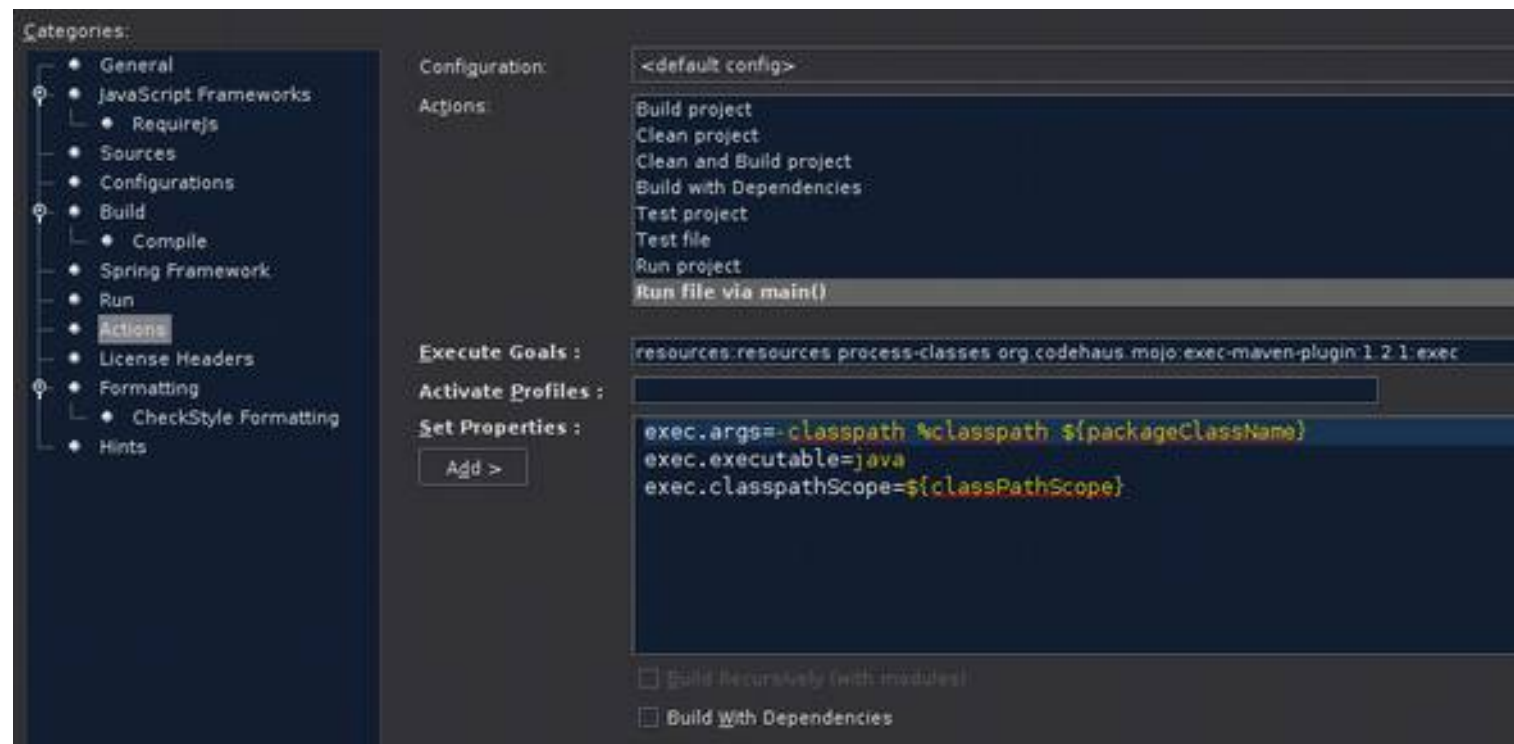
- Open with Archive Manager (default)
- Save File
- Do this automatically for files like this from now on.

Buttons: Cancel, OK

**Generate Project**

## Run spring-boot in Netbeans

### Configuration hint: resources:resources



# FRONTEND BOOTSTRAP



yeoman

- Code generator



YEOMAN

angular seed

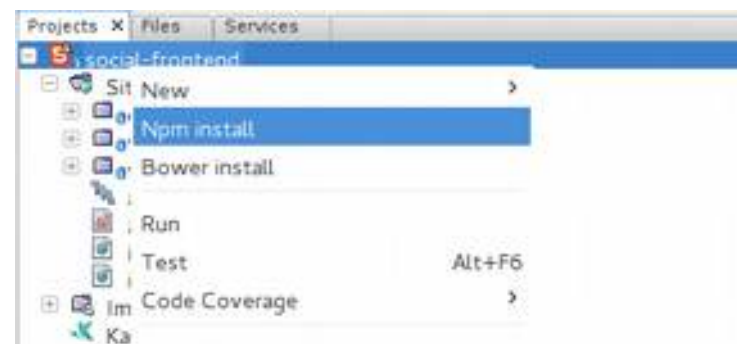
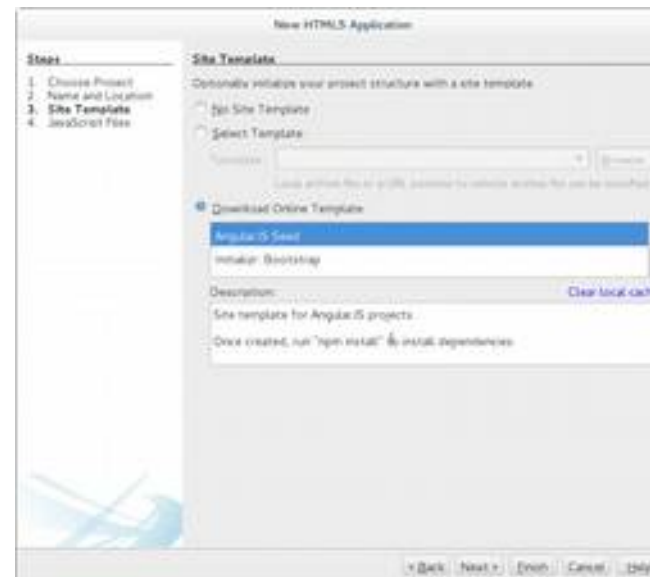
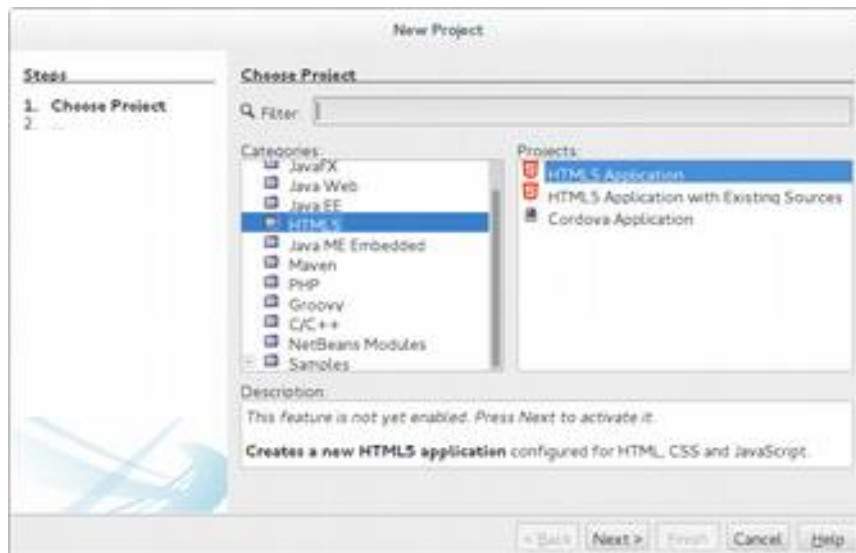
- Static template project



GRUNT



## ■ Netbeans angular seed bootstrap





# IMPLICATIONS

---

TRION

## Teams / Roles

- Develop stateless back-end services
- Develop front-end components
- Different skill set

## Development

- Testability
- Flexibility
- API management
- Different level of re-use
- Speed of change

## Operations

- Different infrastructure
- Distributed logging and diagnosis
- Scaling can leverage cloud infrastructure

Paradigm  
Change

# LESSONS LEARNED



## **Mindshift**

### **Architecture is different**

New architecture patterns

API not only for m2m

## **Effort**

### **Transition involves many aspects**

Technology, Organization, Development

Shift in complexity

## **Evolution**

### **High evolution speed**

Prepare for (even more) constant change

## Your Questions?

Thomas Kruse @everflux

# STOP BACKUP STOP

---



- HTML Round-trip-editing
- Angular message service
- Webcomponent editing
- Spring-boot-init opening, running on tomcat
- Angular-seed opening