

# Spring Boot



Eberhard Wolff

Freelancer

Head Technology Advisory Board  
adesso AG

<http://ewolff.com>

# Prep demo

- Web Browser localhost:8080
- IDE
- Terminal in Project Directory

What does it take  
to implement a  
Java Web  
Application?

Any non-Java  
guys in the room?

Java

Libraries

Build

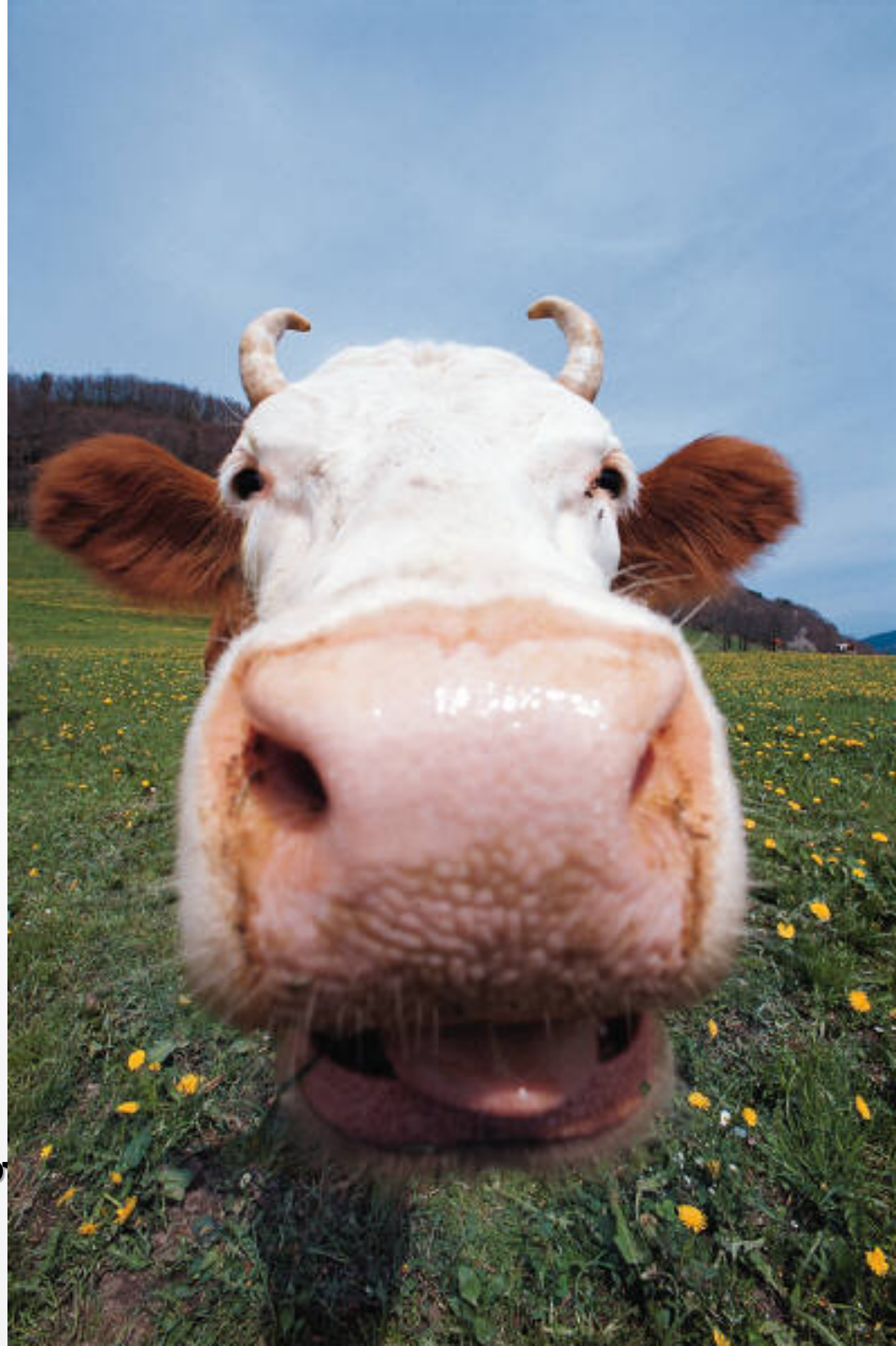
WAR

AppServer

Java

Build

AppSer



raries

WAR

Enter Spring  
Boot

# Spring Boot Demo



# Demo

- Project simplest-spring-boot
- Launch ControllerAndMain
- Open <http://localhost:8080>
- Show main
- Explain `@EnableAutoConfiguration`
- Explain `@RestController` and `@RequestMapping`
- Show pom.xml (parent, starter, maven Plugin)
- Show test

# Spring Boot

- Build much easier
- Supports Gradle, too
- Not limited to simple application
- Can add any other library
- Application much easier to test, debug etc
- Simple code

Let's add  
Spring Data  
JPA!

# Spring Data JPA

- Makes implementation of repositories trivial
- Just provide an interface
- Needs JPA infrastructure of course

# Spring Boot + Spring Data JPA Demo

# Demo: Spring Data JPA

- Show pom.xml: Just added spring-boot-starter-data-jpa and spring-boot-starter-thymeleaf
- Show CustomerRepository
- That is it – no more configuration
- No DataSource, no nothing
- pom.xml: Show hsql.jar dependency
- Note: transactional tests

# Behind the Scenes

- DataSource for HSQL created
- JPA infrastructure created
- Thymeleaf infrastructure created
- How does it work?
- Convention over configuration

# Spring Boot

- Creates sensible infrastructure defaults
- i.e. DataSource if HSQL is on the classpath
- Done by \*AutoConfiguration classes



# Auto Configuration

# Demo Auto Configuration

- Show DataSourceAutoConfiguration
- Show EmbeddedConfiguration
- Show \*AutoConfiguration
- Explain: HibernateJpaAutoConfigure depends on Hibernate classes. Those are included in the dependencies.

# Spring Boot Starter

- Support for specific frameworks

security                  test                  actuator                  web  
                                aop                  remote-shell                  websocket  
  
log4j logback                                  jetty  
                                data-jpa    jdbc    redis                  tomcat  
                                data-mongodb    data-rest                  mobile  
  
                                batch  
                                integration  
                                amqp

# Spring Boot for Dev

# Dev

- Turn Around
- Thymeleaf Templates can be changed on the fly
- ...same with class files
- Spring Loaded makes reloading more powerful

# Spring Boot for Dev Demo

# Demo Dev Turn Around

- Change template
- Reload page
- Show application.properties
- Change code
- Reload page
- Show Spring Loaded
- Run configuration: `-javaagent:/Users/wolff/springloaded-1.1.5.RELEASE.jar -noverify`

# Spring Boot for Ops



# CRaSH Shell

- Ever ssh'd into your application?
- Enter CRaSH
- External library integrated into Spring Boot
- Alternative approach to operations

# Spring Boot CRaSH Demo

# Demo CRaSH

- `ssh -p 2000 user@localhost`
- Password in log output
- `help`
- Mention `jpa` / `jdbc` / problems
- `thread top CTRL-C`
- `metrics`
- `dashboard`

# Actuator

- Provide information about the application
- Via http / JSON
- Can be evaluated by monitoring tools etc.
- Another alternative approach to monitoring

# Spring Boot Actuator Demo

# Actuator Demo

- Show pom.xml – spring-boot-starter-actuator
- Open <http://localhost:8080/monitor.html>
- Show metrics / trace
- Show info and application.properties

# Deploy

- Just package everything in an executable JAR
- ...or a WAR
- Based on Maven or Gradle
- Build in configuration (YAML, properties etc.)

# Spring Boot Deploy Demo



# Deploy Demo

- `mvn package`
- Change packaging to war
- `mvn package` again
- Show `java -jar` with WAR
- `mvn spring-boot:run`

# Conclusion

- Easy to start
- Can add libraries and features as needed
- Dev: Quick turn around
- Ops: Integrated monitoring
- Ops: Easy deployment
- Didn't show Groovy support

# Links

<http://start.spring.io/>

<http://projects.spring.io/spring-boot/>

<https://github.com/spring-projects/spring-boot>

<https://github.com/ewolff/spring-boot-demos>

<https://spring.io/guides/gs/spring-boot/>

<https://spring.io/guides/gs/actuator-service/>

<https://spring.io/guides/gs/spring-boot-cli-and-js/>

# Thank You!!