Why do you trust an open source project?

Open source from

the trenches

Jongo

- First commit 3 years ago
- v1.2 release few days ago
- 15k+ downloads on Maven Central Repo
- jongo.org: 2000 visitors per month

The very beginning

- Have a problem

  just build something for yourself

- Have a big idea

  and opinions « Query in Java as in Mongo Shell »

- Have an enemy

  « Because DBObject sucks! »

- Have a friend

  and then make it real

What do you need?

- A name, short and memorable
- A domain name
- An open-source licence
- A version control system
- A logo

Let’s do this

- **Build something you can manage**
- **Ignore details early on**
- **Get it working ASAP**
  even if the implementation isn’t perfect,
  the early collaboration leads to features that do exactly what user need
- **Epicentre Design**
  focused on the true essence of the project

A small API

- **Make it public, make it work, make it fast**

  this not the same as our day-to-day job
  limit dependencies, fight against code

- **Limit the surface**

  « Most of time must be spent here »

- **Keep it small**

  « Each time you increase the amount of code,
  your software becomes more complicated »

  Getting Real by 37signals

An open API

- « **Public APIs**, like diamonds, are forever »

  so let’s buy zirconium

  Joshua Bloch

- **Favor features over backward compatibility**

- **Don’t be paranoiac**

- **Allow user to implement unexpected things**

  ~~« Design and document for inheritance or else prohibit it. »~~

  Joshua Bloch

Tests

- Test your Public API
- A bug report is free use case
- Create a compatibility tests suite
- Test your code against different dependency versions

Tools

- IntelliJ, free but…
- Github
- Yourkit Profiler
- That’s all folks

Performances

- Release a feature first then optimize it
- Micro benchmarks with Caliper
  - results can be published online
- IntelliJ Chrono Debugger
- Time consuming
  - worth the price?

Continuous Integration

- Cloudbees with FOSS free licence
- Dev@cloud (Jenkins, Git & Maven repos)
- Be prepared for sleepless night
- Running tests on your machine can be enough

Documentation

- **Favor nice public API over javadoc**

  « APIs should be easy to use, hard to misuse […] and self documented »

  Joshua Bloch

- **Github README is not enough**

- **Build a nice promo website**

Website

- Host it at Github and use your domain name
- Add an overview
- Add a manifesto
- And a lot of codes samples in a single page

Release it

- Sonatype OSS Repository Hosting Service
- Well documented in Wiki
- Invest time in a push button release
- Maven tip

  split release:prepare and release:perform

Spread the word

- **Talk about it as often as you can**

  user groups, Paris JUG, BBL, stickers

- **Be easily reachable**

  twitter

- **Respond to critiques thoughtfully**

- **Help people to solve their problems…**

  …and suggest them to use your project

Track it

- **Watch Twitter**
  - use TweetDeck with search words
- **Add Google analytics into your website**
- **Add a tag on stackoveflow**
  - need a reputation greater than 1500
- **Use Sonatype statistics**

Support

- **Provide a bug tracking tool**
  - mailling list, Github issue
- **Answer quick**
- **Make your fixes available ASAP**
  - early releases
- **Don’t panic this is just software**

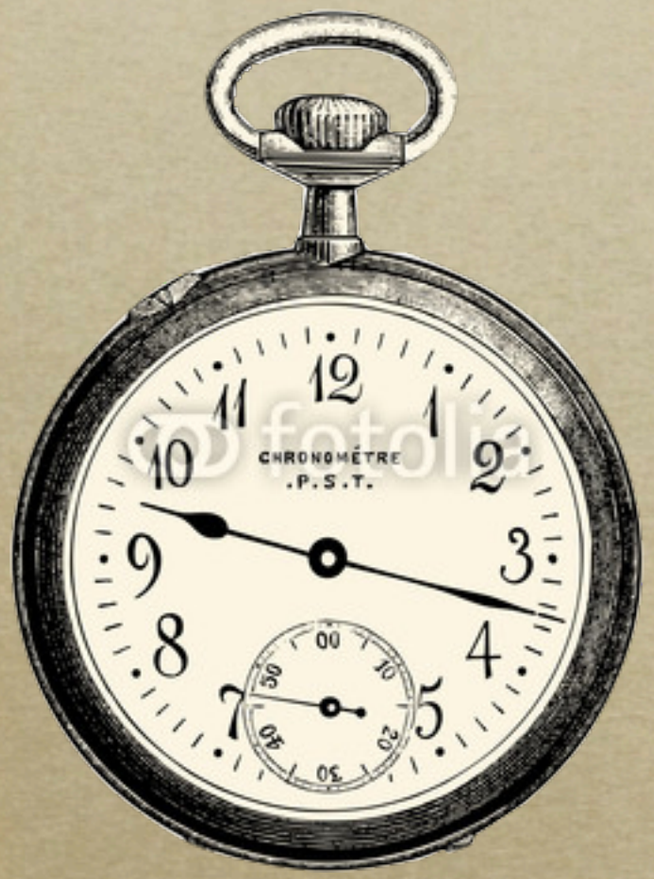
Feature request

- Start with no, don't be a yes-man
- Let users remind you what is important
- Is it compliant with your philosophy?
- Github can help

Pull request

- Its' awesome and incremental
  - and not only for others
- Avoid selfish pull request
- Remember that it's a free contribution
- Try to find a tradeoff
  - you can still refactor

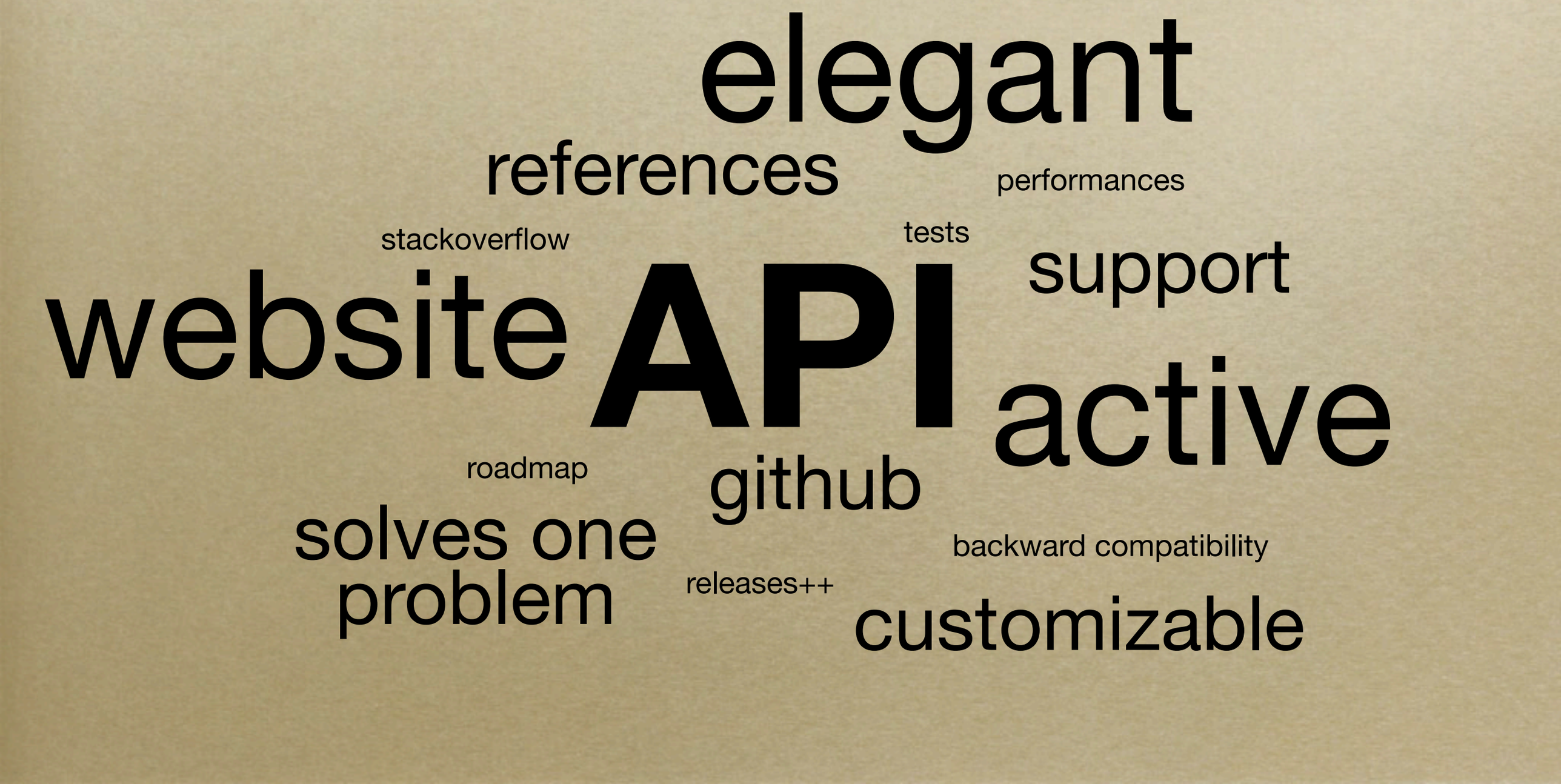
Your enemy



Why?

- Because you had a problem
- Because you can help others *anonymously*
- Lot of time, no money…
- … BUT this is an awesome adventure

Why do **we** trust an open source project?



Reading

Getting Real

The smarter, faster, easier way to build a successful web application by 37 signals
https://gettingreal.37signals.com/

Effective Java

by Joschua Blosch
http://tinyurl.com/pnlhxfn

Clean Code

by Robert C. Martin
http://tinyurl.com/7jawt5u

Practical API Design

Confessions of a Java Framework Architect by Jaroslav Tulach
http://tinyurl.com/pvqw226

Blogs

API Design vs. API Usability *http://tinyurl.com/p6gokr6*
Dependency Injection with constructors? *http://tinyurl.com/pvagke8*
Evolving Java-based APIs 2 *http://tinyurl.com/qevu6qn*
Bumper-Sticker API Design http://tinyurl.com/3t5znr

Hall of Fails

Please use Google Translate

**« I try and I mention, Not update pojo java and not display error.
In MongoDB I did but did not know how to Jongo.
thank you very much for the example »**

The student

**« Mongo jongo grabbed my penis in PE today!!
Did the teacher listen to me? Not a single bit »**

The epic fail

- « I'm trying to query by Enum but I get this stack trace: »

Caused by: java.lang.IllegalArgumentException: can't serialize class com.foo.model.WorkflowTaskState

at *org.bson.BasicBSONEncoder._putObjectField(BasicBSONEncoder.java:234)*
at com.mongodb.DBCollection.insert(DBCollection.java:132)
at com.foo.WorkflowTaskDao.saveTasks(WorkflowTaskDao.java:190)
... 33 more

- ***« The stack trace you provide does not contain a method invocation from org.jongo…., are you using Jongo? »***
- **« Not yet »**

Jongo

v1.2

http://www.jongo.org

@bguerout ~ @amsellemyves