

JavaScript&QA工程师

yuanzhijia@yidengxuetang.com

测试核心概念

- 单元测试
- 性能测试
- 安全测试
- 功能测试

单元测试

why?

- 正确性：测试可以验证代码的正确性，在上线前做到心里有底
- 自动化：当然手工也可以测试，通过console可以打印出内部信息，但是这是一次性的事情，下次测试还需要从头来过，效率不能得到保证。通过编写测试用例，可以做到一次编写，多次运行
- 解释性：测试用例用于测试接口、模块的重要性，那么在测试用例中就会涉及如何使用这些API。其他开发人员如果要使用这些API，那阅读测试用例是一种很好地途径，有时比文档说明更清晰
- 驱动开发，指导设计：代码被测试的前提是代码本身的可测试性，那么要保证代码的可测试性，就需要在开发中注意API的设计，TDD将测试前移就是起到这么一个作用
- 保证重构：互联网行业产品迭代速度很快，迭代后必然存在代码重构的过程，那怎样才能保证重构后代码的质量呢？有测试用例做后盾，就可以大胆的进行重构

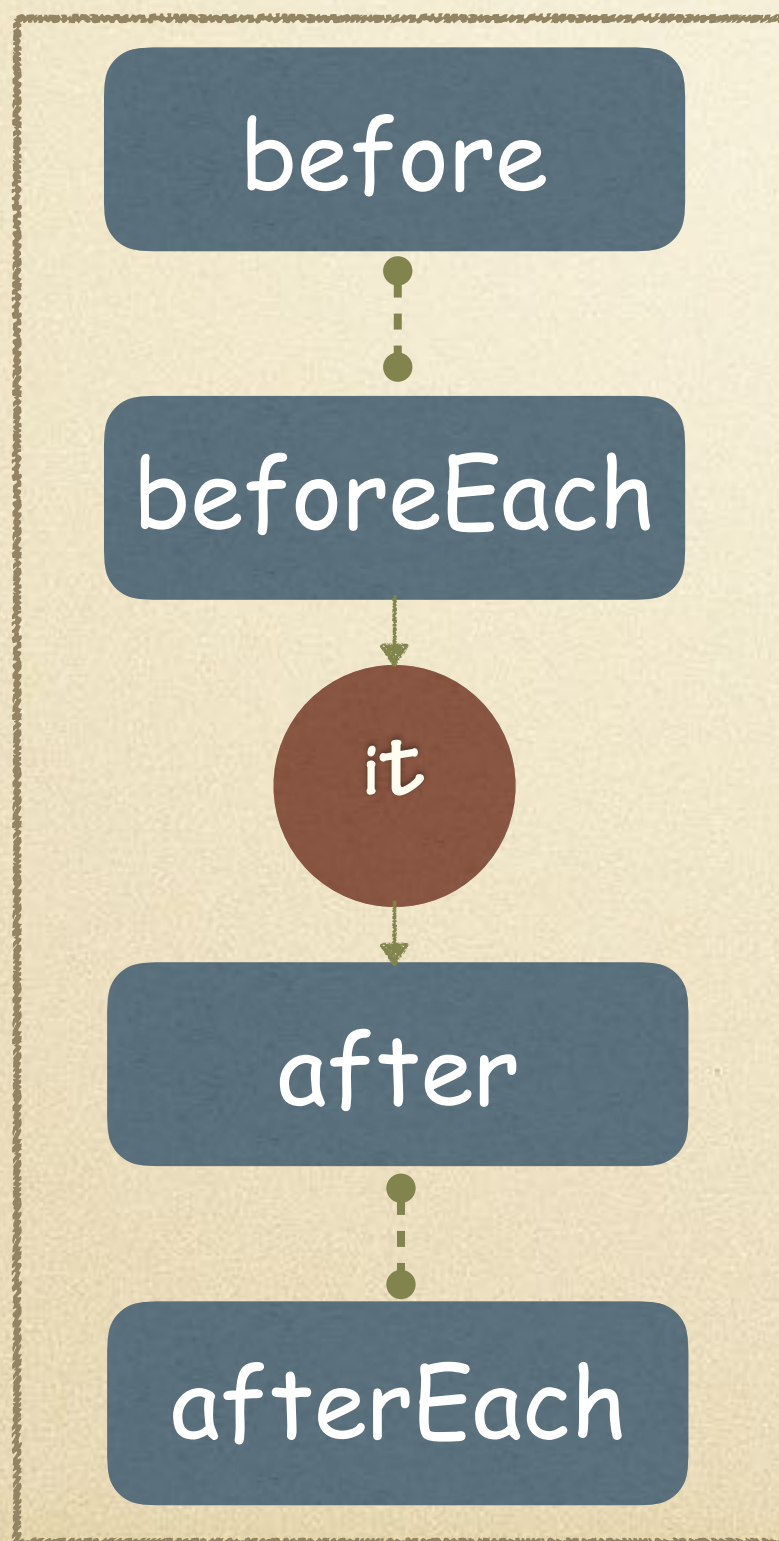
单元测试

- 目的：单元测试能够让开发者明确知道代码结果
- 原则：单一职责、接口抽象、层次分离
- 断言库：保证最小单元是否正常运行检测方法
- 测试风格：测试驱动开发(Test-Driven Development,TDD)、(Behavior Driven Development,BDD)行为驱动开发均是敏捷开发方法论。
TDD关注所有的功能是否被实现(每一个功能都必须有对应的测试用例), suite配合test利用assert('tobi' == user.name);
BDD关注整体行为是否符合整体预期,编写的每一行代码都有目的提供一个全面的测试用例集。expect/should,describe配合it利用自然语言expect(1).toEqual(fn())执行结果。

单元测试框架

- better-assert(TDD断言库Github 190star 19fork)
- should.js(BDD断言库Github 2295star 194fork)
- expect.js(BDD断言库Github 1391star 162fork)
- chai.js(TDD BDD双模Github 2823star 271fork)
- Jasmine.js(BDD Github10723star1680fork)
- Node.js本身集成 `require("assert");`
- Intern 更是一个大而全的单元测试框架
- QUnit 一个游离在jQuery左右的测试框架
- Macaca 一套完整的自动化测试解决方案 国产神器来自阿里巴巴

单元测试运行流程



每一个测试用例组通过
describe进行设置

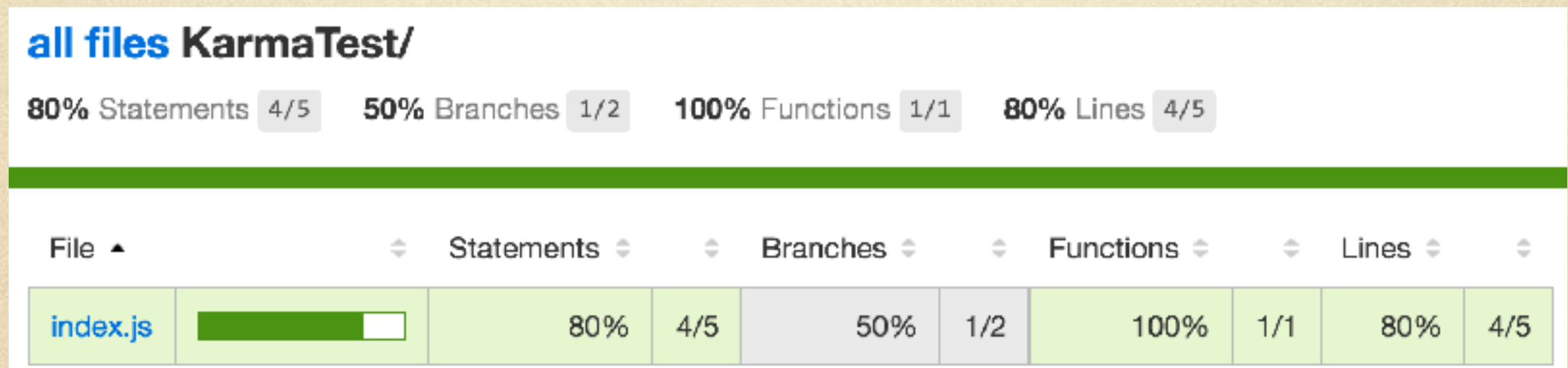
- 1.before单个测试用例(it)开始前
- 2.beforeEach每一个测试用例开始前
- 3.it定义测试用例 并利用断言库进行设置chai如：`expect(x).to.equal(true);`
异步mocha
- 4.以上专业术语叫mock

自动化单元测试

- karma 自动化runner集成PhantomJS无刷新
- `npm install -g karma`
- `npm install karma-cli —save-dev`
- `npm install karma-chrome-launcher —save-dev`
- `npm install karma-phantomjs-launcher —save-dev`
- `npm install karma-mocha —save-dev`
- `npm install karma-chai —save-dev`

报告和单测覆盖率检查

- `npm install karma-coverage --save-dev`
- `coverageReporter: { type: 'html', dir: 'coverage/' }` // 配制代码覆盖测试率生成结果



性能测试

基准测试

- 面向切面编程AOP无侵入式统计
- Benchmark基准测试方法，它并不是简单地统计执行多少次测试代码后对比时间，它对测试有着严密的抽样过程。执行多少次取决于采样到的数据能否完成统计。根据统计次数计算方差。

压力测试

- 对网络接口做压力测试需要检查的几个常用指标有吞吐率、响应时间和并发数，这些指标反映了服务器并发处理能力。
- PV网站当日访问人数 UV独立访问人数。PV每天几十万甚至上百万就需要考虑压力测试。换算公式 $QPS = PV / t$ ps : $1000000 / 10 * 60 * 60 = 27.7$ (100万请求集中在10个小时，服务器每秒处理27.7个业务请求)
- 常用的压力测试工具是ab、siege、http_load。
- `ab -c 100 -n 100 http://localhost:8001` 每秒持续发出28个请求
Request per second 表示服务器每秒处理请求数 即为QPS
Failed requests 表示此次请求失败的请求数 理论上压测值越大增加
Connection Times 连接时间，它包括客户端向服务器端建立连接、服务器端处理请求、等待报文响应的过程

JavaScript测试与检测

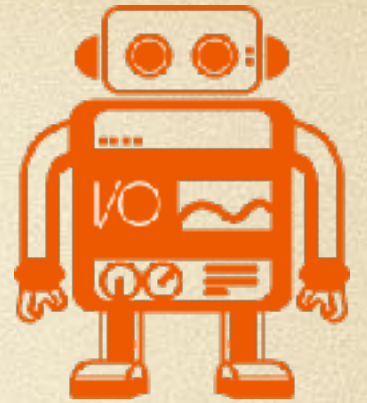
安全测试

安全漏洞检查

- XSS
- SQL
- CSRF

功能测试

用户真实性检查



- selenium-webdriver
- protractor selenium-standalone
- <http://webdriver.io/> WEBDRIVERI/O
- 冒烟测试 SmokeTest 自由测试的一种，找到一个BUG开发修复，然后专门针对此BUG,优点节省生煎防止build失败，缺点是覆盖率极低。
- 回归测试 修改一处对整体功能全部测试，一般配合自动化测试。

JavaScript测试与检测

JsLint&JsHint

JavaScript Lint&Hint

- 目的：检测JavaScript代码标准
- 原因：JavaScript代码诡异，保证团队代码规范
- lint：<http://www.jshint.com/>
- hint：<http://www.jshint.com/>
- 搭配自动化任务管理工具完善自动化测试grunt-jshint、grunt-jshint

Q&A