

个人资料



jackiej

访问: 117119次

积分: 1700

等级: BLOG > 4

排名: 第13728名

原创: 37篇 转载: 85篇

译文: 0篇 评论: 16条

文章搜索

文章分类

.Net Framework (27)

BizTalk (11)

C++ (9)

Economics (3)

Management (13)

Other (29)

Science (8)

Silverlight (1)

Technology (35)

Testing (6)

Git (1)

Objective-C (1)

文章存档

2013年03月 (1)

2013年02月 (2)

2013年01月 (1)

2012年06月 (1)

2012年05月 (2)

展开

阅读排行

Cucumber 入门一

(17701)

学院APP首次下载，可得50C币！ 欢迎来帮助开源“进步” 当讲师？爱学习？投票攒课吧 CSDN 2015博客之星评选 感恩分享

Cucumber 入门一

2013-02-27 16:15 17738人阅读 评论(3) 收藏 举报

分类: Testing (5)

目录(?) [+]

(转自: <http://www.cnblogs.com/jarodzz/archive/2012/07/02/2573014.html>)

第一次看到Cucumber和BDD（Behavior Driven Development, 行为驱动开发），是在四年前。那时才开始工作，对软件测试工具相当着迷。只要是开源的、免费的，我就一定要下载，安装，试用。有的工具用途单一、好懂（如Jmeter，Watir）；有的工具，则涉及到一些软件领域的独有概念，不好懂，（如STAF，Cucumber）。好懂的，我上手、试用、推广，不亦乐乎；不好懂的，就只能丢在一边，这里面就包括Cucumber。

再次看到Cucumber，已是两年前。我对软件开发的理解也深了些，这一看，可真是个好东西。之后我与Cucumber间发生的故事，稍后慢慢向大家交代。这开篇的第一章，我想献给如当年的我一样，偶然见到了Cucumber和BDD，却不明确所以将之丢在一边的家伙们。

初闻Cucumber的人，第一件事一定是来到Cucumber的[首页](#)，第一眼看到的一定是

1: Describe behaviour in plain text

```
Feature: Addition
  In order to avoid silly mistakes
  As a math idiot
  I want to be told the sum of two numbers

Scenario: Add two numbers
  Given I have entered 50 into the calculator
  And I have entered 70 into the calculator
  When I press add
  Then the result should be 120 on the screen
```

2: Write a step definition in Ruby

```
Given /^I have entered (.*) into the calculator/ do |n|
  calculator = Calculator.new
  calculator.push(n.to_i)
end
```

3: Run and watch it fail

```
$ cucumber features/addition.feature
Feature: Addition # features/addition.feature
  In order to avoid silly mistakes
  As a math idiot
  I want to be told the sum of two numbers
  Scenario: Add two numbers # features/addit
    Given I have entered 50 into the calculator # features/step_d
      uninitialized constant Calculator (NameError)
    //features/step_definitions/calculator_steps.rb:2:in `Given /
    //features/addition.feature:7:in `Given I have entered 50 into
    And I have entered 70 into the calculator # features/step_d
    When I press add # features/addit
    Then the result should be 120 on the screen # features/addit
```

4. Write code to make the

```
class Calculator
  def push(n)
    @args ||= []
    @args << n
  end
end
```

5. Run again and see the step pass

```
$ cucumber features/addition.feature
Feature: Addition # features/addition.feature
  In order to avoid silly mistakes
  As a math idiot
  I want to be told the sum of two numbers
  Scenario: Add two numbers # features/addit
    Given I have entered 50 into the calculator # features/step_d
    And I have entered 70 into the calculator # features/step_d
    When I press add # features/addit
    Then the result should be 120 on the screen # features/addit
```

6. Repeat 2-5 until green like a cucumber

```
$ cucumber features/addition.feature
Feature: Addition # features/addition.feature
  In order to avoid silly mistakes
  As a math idiot
  I want to be told the sum of two numbers
  Scenario: Add two numbers # features/addit
    Given I have entered 50 into the calculator # features/step_d
    And I have entered 70 into the calculator # features/step_d
    When I press add # features/step_d
    Then the result should be 120 on the screen # features/step_d
```

很不幸的是，这六张图不太好懂。因为它们按照BDD的流程来编写的。为了让它们好懂些，我们抛开BDD，采用传统的软件开发模型（设计->编码->测试）来看它。传统流程如下：

图1，这是一个被测系统——用ruby编写的计算器。

```
class Calculator
  def push(n)
    @args ||= []
    @args << n
  end
end
```

为了便于大家理解，我试着修改了一些。

```
class Calculator
  def push(n)
    #记数
    @args ||= 0
    [ ] #初始化空数组
    @args <<
```

一个图论中的概念: Betv	(9628)
VS 2010 TFS 指南	(3919)
Storyboard和属性自动绑	(3423)
150楼盘调查: 北大教授	(2388)
搭建git服务器及利用git h	(2283)
一个JS动画框架-基于HT	(2013)
在MSMQ中使用事务处理	(1861)
使用 TightVNC 从 Windo	(1650)
Microsoft Project 2010 简	(1638)

评论排行	
Cucumber 入门一	(3)
用 Lisp 语言计算一个集合	(2)
一个图论中的概念: Betv	(2)
在C#中接收系统屏幕锁定	(2)
VS 2010 TFS 指南	(2)
通过程序自动向 TFS 201	(1)
在MSMQ中使用事务处理	(1)
Linux文件系统发展迅速	(1)
Chrome线程模型 (编译)	(1)
GIT和SVN之间的五个基	(1)

推荐文章	
*Hadoop节点"慢磁盘"监控	
*假如你想成为全栈工程师...	
*没有躲过的坑--正则表达式截取字符串	
*CardView完全解析与RecyclerView结合使用(三十二)	
*And roid 高仿微信发朋友圈浏览图片效果	
*通过Ajax的方式执行GP服务	

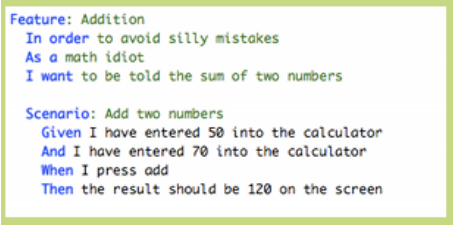
最新评论	
Cucumber 入门一	
Grvan: 你好, 我用英文运行都没有问题, 但是用中文进行描述时, 运行结果显示乱码, 请问这个问题怎么解决	
Cucumber 入门一	
judy2008ilufam: 检查一下你的文件名是不是相对应的。	
Cucumber 入门一	
kakaka2011: 你好按照你的例子跑下来。cucumber addition.feature总是1 scenario ...	
用 Lisp 语言计算一个集合的所有	
GGeneral2: sicp上有比较短的实现。	
用 Lisp 语言计算一个集合的所有	
GGeneral2: 幂集	
通过程序自动向 TFS 2010 中添加	
xh91900: 为什么我的连不上, 报错"未将对象引用设置到对象的实例"一开始就错""projectCollectio...	
在C#中接收系统屏幕锁定和解锁	
xunmi_love: 帮到我了, 感谢	
在C#中接收系统屏幕锁定和解锁	
星00恋: 啥意思啊?	
一个图论中的概念: Betweennes	
superyangtze: 终于明白了。	
GIT和SVN之间的五个基本区别	
stamina88: 第一次听说, 长见识了	

技术

```
n
end
def sum()
#返回所有数字和
sum
= 0
@args.each do |i|
sum
+= i
end
@result =
sum
end
def result
@result
end
end
end
```

计算器Calculator提供两个功能: 记数push; 加和sum。push将数字一一记录在@args数组中; sum则将所有@args数组中的数字累加得和, 存入@result中。写完了被测系统, 我们来编写测试用例。

图2, 这是为了测试上述计算器, 使用Cucumber描述的测试用例。



原图中的英文描述, 被我翻译成了中文。:

Feature:
计算器
Scenario:
两数相加
Given
我有一个计算器
And
我向计算器输入50
And
我向计算器输入70
When
我点击累加
Then
我应该看到结果120

支持中、英等自然语言, 是Cucumber的特点之一。在Cucumber的帮助文档里, 声明它支持包括简体中文、繁体中文、日文、韩文和英文在内的45种语言。
注意: 我并未将所有英文都翻译成中文, 而是留下了几个关键字:

- Feature (功能)
- Scenario (情景)
- Given (给定)
- And (和)
- When (当)
- Then (则)

它们的含义与原有自动化测试工具中的概念相同, 类比如下:

Cucumber	Unit Test
----------	-----------

Hadoop

经济

谢国忠的博客
经济观察网

科学

环球科学
集智俱乐部

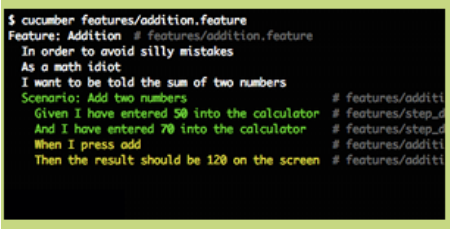
其它

爱稀奇
时光网
新摄影

Feature (功能)	test suite （测试用例集）
Scenario （情景）	test case （测试用例）
Given （给定）	setup （创建测试所需环境）
When （当）	test （触发被测事件）
Then （则）	assert(断言， 验证结果)

Cucumber放弃了原有的关键字，而选择了左边五种，只是为了更加流畅地支持自然语言。使用Cucumber的关键字，创建了测试用例，接下来，要如何使用Cucumber来运行它呢？

图3，这是运行Cucumber时的画面。



在一台安装好Cucumber的机器上，运行上述测试用例，便可以看到下列输出：

```
Feature:
  计算器

  Scenario:
    两数相加 #
    features/calculator.feature:3

    Given
    我有一个计算器 #
    features/calculator.feature:4

    And
    我向计算器输入50 #
    features/calculator.feature:5

    And
    我向计算器输入70 #
    features/calculator.feature:6

    When
    我点击累加 #
    features/calculator.feature:7

    Then
    我应该看到结果120 #
    features/calculator.feature:8

1 scenario
  (1 undefined)
5 steps
  (5 undefined)
0m0.005s

You
  can implement step definitions for undefined
  steps with these snippets:

Given
  /^我有一个计算器$/ do
    pending
  #
  express the regexp above with the code you wish you had
end

Given
  /^我向计算器输入(\d+)/ do |arg1|
```

```
pending

#
  express the regexp above with the code you wish you had
end

When
  /^我点击累加$/ do
    pending
  #
    express the regexp above with the code you wish you had
  end

Then
  /^我应该看到结果(\d+)$/ do |arg1|
    pending
  #
    express the regexp above with the code you wish you had
  end
```

Cucumber首先输出的是测试用例的描述，然后3行总结性地输出：本功能（Feature）有1个情景（1 scenario）；5个步骤（5 steps），全部5个步骤均未定义（undefined）；运行耗时0.005秒。这里出现了两个新名词：步骤(steps)和步骤定义(step definitions)。在Cucumber中，以关键字Given, And, When, Then开头的每一行，为一个步骤。在两数相加的情景中，一共有5行。因此，结果显示:5个步骤。

如何定义一个步骤，在Cucumber的运行结果中也给出了详细的办法。在3行总结性输出后，紧接着便是：[You can implement...](#)即：你可以使用下面的代码段实现步骤定义，然后是4个小的代码段。这些代码段，便是Cucumber依照情境中我们使用的5个步骤，帮助我们生成的步骤定义框架。每个框架都将内容部分空白出来，等待填充。下面，我们来进行步骤定义。

图4, 这是一个步骤定义的代码示范。

```
Given /I have entered (.*) into the calculator/ do |n|
  calculator = Calculator.new
  calculator.push(n.to_i)
end
```

我们依照图2的样子，向中文步骤中填入代码，如下：

```
Given
  /^我有一个计算器$/ do
    @c =
      Calculator.new
  end

Given
  /^我向计算器输入(\d+)$/ do |num|
    @c.push(num.to_i)
  end

When
  /^我点击累加$/ do
    @c.sum
  end

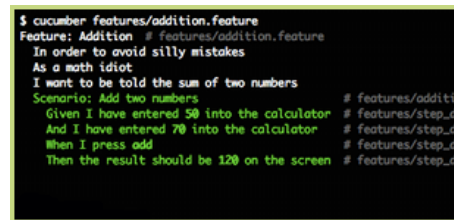
Then
  /^我应该看到结果(\d+)$/ do |result|
    @c.result.should
      == result.to_i
  end
```

步骤定义的过程，就是向代码段——步骤定义框架——中填入代码的过程，即：用代码来描述你期望的，该步骤应该执行的动作。[完整的步骤定义](#)是一个函数，它：

- 以正则表达式作为函数名
- 匹配值作为参数
- 以测试人员输入的代码作为内容

因为有了正则表达式的匹配，5个步骤只需要4个步骤定义。“我向计算器输入50、70”两个步骤，都可以用“我向计算器输入(\d+)”一个正则表达式来描述。匹配值被自动提取出来作为参数，传入代码。注意：所有匹配值，即参数，都是以字符串的形式传递，因此，我加入了num.to_i 与 result.to_i，将得到的字符串转为整形。步骤定义完成，再次执行Cucumber。屏幕将会显示一片绿色。

图5，它是一个执行Cucumber测试用例，并成功通过的画面。



步骤定义完成后，再次运行Cucumber。Cucumber会找到步骤定义，并按照其代码去执行。结果如下：

```
Feature:
  加法

  Scenario:
    两数相加 #
    features/calculator.feature:3

    Given
    我有一个计算器 #
    features/step_definitions/a.rb:2

    And
    我向计算器输入50 #
    features/step_definitions/a.rb:6

    And
    我向计算器输入70 #
    features/step_definitions/a.rb:6

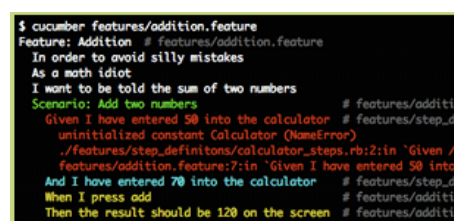
    When
    我点击累加 #
    features/step_definitions/a.rb:10

    Then
    我应该看到结果120 #
    features/step_definitions/a.rb:14

1 scenario
  (1 passed)
5 steps
  (5 passed)
0m0.003s
```

步骤定义被我保存在文件夹step_definitions下的a.rb当中。步骤定义所在文件与起始行数，被打印在每个步骤结尾，以方便查找和修改。最后，Cucumber总结性地输出运行结果：1个情景，5个步骤，全部通过。

图6，这是一个执行Cucumber测试用例，但失败的画面。



为了让这个已经十分简单的计算器产生bug，我只好将它改错为：

```
class Calculator
  def sum()
    sum
  = 0
    @args.each do |n|
      sum
    = n    #此处原为: sum
    += n
    end
    @result =
    sum
  end
end
```

再次运行Cucumber,结果为：

```
Feature:
  加法

  Scenario:
    两数相加    #
    features/calculator.feature:3

    Given
    我有一个计算器    #
    features/step_definitions/a.rb:2

    And
    我向计算器输入50    #
    features/step_definitions/a.rb:6

    And
    我向计算器输入70    #
    features/step_definitions/a.rb:6

    When
    我点击累加    #
    features/step_definitions/a.rb:10

    Then
    我应该看到结果120 #
    features/step_definitions/a.rb:14

    expected:
    120

    got:
    70 (using
    ==) (RSpec::Expectations::ExpectationNotMetError)
    ./features/step_definitions/a.rb:15:in `^我应该看到结果(\d+)\$/'
    features/calculator.feature:8:in `Then
    我应该看到结果120'
```

Failing

Scenarios:

cucumber

features/calculator.feature:3 #

Scenario: 两数相加

1 scenario

(1 failed)

5 steps

(1 failed,

4 passed)

0m0.004s

失败的步骤是用红色标示出来的。在最后一个步骤中，Cucumber期待的结果为120，但得到的是70。注意：失败的情景列表（Failing Scenarios）里列出的是：“[两数相加](#)”这个情景所在的文件与起始行数。这是因为一个功能文件内，可能含有多个情景，这种输出可以便于找到出错的情景。
接下来的总结性结果为：1个情景失败（1 failed），5个步骤中，4个通过，1个失败。

作为自动化测试工具的Cucumber，就介绍到这里。

在继续之前，我们先回顾一下本章内容。

回顾：

- Cucumber是一个自动化测试工具
- 它提供了自然语言的支持，我们可以用自然语言描述、并执行测试用例
- 它提供了自然语言与代码的衔接，通过步骤与步骤定义
- 它提供了自然语言对代码的调用，当步骤定义结束后，执行Cucumber，它会自动调用步骤定义内的代码运行
- 它提供了良好的断言（assert）机制。当执行失败时，我们可以看到完成的测试用例，以及明确的失败原因。

[上一篇](#) Storyboard和属性自动绑定

[下一篇](#) 使用 SSH Tunnel 加密 VNC 遠端桌面連線

顶 0 踩 0

我的同类文章

Testing (5)	
• 简单快速的可用性测试	• 网络数据的背后——网络日志的分析指标
• 超简单的变化测试	• JUnit2.0详细使用方法
• JUnit4 详解	

主题推荐

[color](#) [cucumber](#) [软件开发](#) [驱动开发](#) [软件测试](#) [jmeter](#)

猜你在找

- [iOS自动化测试第一季](#)
- [Python自动化测试MonkeyRunner](#)
- [Android自动化测试第二季（提高篇）](#)
- [Android自动化测试第三季](#)
- [Android自动化测试第一季（基础篇）](#)







查看评论

3楼 [Gryan](#) 2015-01-29 18:55发表



你好，我用英文运行都没有问题，但是用中文进行描述时，运行结果显示乱码，请问这个问题怎么解决

2楼 [judy2008ilufam](#) 2014-11-20 11:07发表



检查一下你的文件名是不是相对应的。

1楼 [kakaka2011](#) 2014-08-28 00:20发表



你好按照你的例子跑下来。cucumber addition.feature
总是

1 scenario (1 undefined)

5 steps (5 undefined)

提示是

WARNING: cannot load such file -- 2.0/gherkin_lexer_en

Couldn't load 2.0/gherkin_lexer_en这个地方出错?

请问你知道是什么原因吗?

您还没有登录,请[\[登录\]](#)或[\[注册\]](#)

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

核心技术类目

全部主题 Hadoop AWS 移动游戏 Java Android iOS Swift 智能硬件 Docker OpenStack
VPN Spark ERP IE10 Eclipse CRM JavaScript 数据库 Ubuntu NFC WAP jQuery
BI HTML5 Spring Apache .NET API HTML SDK IIS Fedora XML LBS Unity
Splashtop UML components Windows Mobile Rails QEMU KDE Cassandra CloudStack
FTC coremail OPhone CouchBase 云计算 iOS6 Rackspace Web App SpringSide Maemo
Compuware 大数据 aptech Perl Tornado Ruby Hibernate ThinkPHP HBase Pure Solr
Angular Cloud Foundry Redis Scala Django Bootstrap

公司简介 | 招贤纳士 | 广告服务 | 银行汇款帐号 | 联系方式 | 版权声明 | 法律顾问 | 问题报告 | 合作伙伴 | 论坛反馈

网站客服 杂志客服 微博客服 webmaster@csdn.net 400-600-2320 | 北京创新乐知信息技术有限公司 版权所有 | 江苏乐知网络技术有限公司 提供商务支持

京 ICP 证 070598 号 | Copyright © 1999-2014, CSDN.NET, All Rights Reserved

