# Integration with Travis CI

# OWdmateyou'll learn

- Prerequisites
- Implementingwetstantegrate with Travis CI
- IntegratingRuiththeaTravis CI pipeline push the test report to Xray
- • SMalidateshat the test results are available in Jira
- Triggering automation from Xray side
- References

Source-code for this tutorial

• code is available in GitHub

### **Overview**

Travis CI is a continuous integration service used to build and test projects hosted in GitHub, Bitbucket, GitLab and Assembla.

The tool will automatically detect when a commit has been made and pushed to a repository that is configured with Travis CI, and each time this happens, it will try to execute what is defined in the Travis CI yaml file (build, pack, tests, etc).

### Prerequisites

For this example we will use a simple calculator application that is part of the code available and tests defined in JUnit. We will use that repository to start a pipeline in Travis CI, where we will execute the compilation and tests, and report back to Xray.

What you'll need:

- Java and Maven installed
- ٠ Travis CI account linked to GitHub (in this case, you can also use Assembla, Bitbucket or GitLab)

### Implementing tests

In the repository you will find a calculator application and also tests that are validating those abilities.

The test consists of validating the supported operations (addition, subtraction, division and multiplication) of calculator application.

```
./test/java/com/xpand/java
```

```
package com.xpand.java;
import org.junit.After;
import org.junit.Before;
import org.junit.Test;
import static org.hamcrest.CoreMatchers.is;
import static org.junit.Assert.assertThat;
public class CalcTest {
    @Before
   public void setUp() throws Exception {
    }
    @After
   public void tearDown() throws Exception {
    }
        @Test
    public void CanAddNumbers()
    ł
        assertThat(Calculator.Add(1, 1), is(2));
        assertThat(Calculator.Add(-1, 1), is(0));
    }
    @Test
   public void CanSubtract()
    {
        assertThat(Calculator.Subtract(1, 1), is(0));
        assertThat(Calculator.Subtract(-1, -1), is(0));
        assertThat(Calculator.Subtract(100, 5), is(95));
    }
    @Test
    public void CanMultiply()
    {
        assertThat(Calculator.Multiply(1, 1), is(1));
        assertThat(Calculator.Multiply(-1, -1), is(1));
       assertThat(Calculator.Multiply(100, 5), is(500));
    }
    public void CanDivide()
    {
        assertThat(Calculator.Divide(1, 1), is(1));
        assertThat(Calculator.Divide(-1, -1), is(1));
        assertThat(Calculator.Divide(100, 5), is(20));
    }
   @Test
    public void CanDoStuff()
    {
        assertThat(true, is(true));
    }
}
```

Once the code is implemented, we can compile and execute the tests with the following command:

mvn clean compile test --file pom.xml

#### The results are immediately available in the terminal.



In this example, all test have succeed, the output generated in the terminal is the above one and the corresponding JUnit report is below:

#### Junit Report <?xml version="1.0" encoding="UTF-8"?> <testsuite xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi: schemaLocation="https://maven.apache.org/surefire/maven-surefire-plugin/xsd /surefire-test-report.xsd" name="com.xpand.java.CalcTest" time="0.047" tests="4" errors="0" skipped="0" failures="0"> <properties> <property name="java.runtime.name" value="Java(TM) SE Runtime</pre> Environment"/> <property name="java.vm.version" value="16.0.1+9-24"/> <property name="sun.boot.library.path" value="/Library/Java</pre> /JavaVirtualMachines/jdk-16.0.1.jdk/Contents/Home/lib"/> <property name="maven.multiModuleProjectDirectory" value="/Users</pre> /cristianocunha/Documents/Projects/tutorials/tutorial-java-junit-Travis CI" /> <property name="java.vm.vendor" value="Oracle Corporation"/> <property name="java.vendor.url" value="https://java.oracle.com/"/> <property name="guice.disable.misplaced.annotation.check" value="true"</pre> /> <property name="path.separator" value=":"/> <property name="java.vm.name" value="Java HotSpot(TM) 64-Bit Server VM"</pre> /> <property name="user.country" value="PT"/> <property name="sun.java.launcher" value="SUN\_STANDARD"/> <property name="java.vm.specification.name" value="Java Virtual" Machine Specification"/> <property name="user.dir" value="/Users/cristianocunha/Documents</pre> /Projects/tutorials/tutorial-java-junit-Travis CI"/> <property name="java.vm.compressedOopsMode" value="Zero based"/> <property name="java.runtime.version" value="16.0.1+9-24"/> <property name="os.arch" value="x86\_64"/> <property name="java.io.tmpdir" value="/var/folders/s1</pre> /z2z7bgns1\_qd21w\_t5j33w200000gn/T/"/> <property name="line.separator" value=" "/> <property name="java.vm.specification.vendor" value="Oracle" Corporation"/> <property name="os.name" value="Mac OS X"/> <property name="classworlds.conf" value="/opt/apache-maven-3.8.1/bin</pre> /m2.conf"/> <property name="sun.jnu.encoding" value="UTF-8"/> <property name="java.library.path" value="/Users/cristianocunha/Library</pre> /Java/Extensions:/Library/Java/Extensions:/Network/Library/Java/Extensions: /System/Library/Java/Extensions:/usr/lib/java:."/> <property name="maven.conf" value="/opt/apache-maven-3.8.1/conf"/> <property name="jdk.debug" value="release"/>

```
<property name="java.class.version" value="60.0"/>
    <property name="java.specification.name" value="Java Platform API</pre>
Specification"/>
    <property name="sun.management.compiler" value="HotSpot 64-Bit Tiered"
Compilers"/>
    <property name="os.version" value="10.15.7"/>
    <property name="library.jansi.path" value="/opt/apache-maven-3.8.1/lib</pre>
/jansi-native"/>
    <property name="http.nonProxyHosts" value="local|*.local|169.254/16|*.</pre>
169.254/16"/>
    <property name="user.home" value="/Users/cristianocunha"/>
    <property name="user.timezone" value="Europe/Lisbon"/>
    <property name="file.encoding" value="UTF-8"/>
    <property name="java.specification.version" value="16"/>
    <property name="user.name" value="cristianocunha"/>
    <property name="java.class.path" value="/opt/apache-maven-3.8.1/boot</pre>
/plexus-classworlds-2.6.0.jar"/>
    <property name="java.vm.specification.version" value="16"/>
    <property name="sun.arch.data.model" value="64"/>
    <property name="sun.java.command" value="org.codehaus.plexus.</pre>
classworlds.launcher.Launcher clean compile test --file pom.xml"/>
    <property name="java.home" value="/Library/Java/JavaVirtualMachines</pre>
/jdk-16.0.1.jdk/Contents/Home"/>
    <property name="user.language" value="en"/>
    <property name="java.specification.vendor" value="Oracle Corporation"/>
    cproperty name="java.vm.info" value="mixed mode, sharing"/>
    <property name="java.version" value="16.0.1"/>
    <property name="java.vendor" value="Oracle Corporation"/>
    <property name="maven.home" value="/opt/apache-maven-3.8.1"/>
    <property name="file.separator" value="/"/>
    <property name="java.version.date" value="2021-04-20"/>
    <property name="java.vendor.url.bug" value="https://bugreport.java.com"
/bugreport/"/>
    <property name="sun.io.unicode.encoding" value="UnicodeBig"/>
    <property name="sun.cpu.endian" value="little"/>
    <property name="socksNonProxyHosts" value="local|*.local|169.254/16|*.</pre>
169.254/16"/>
    <property name="ftp.nonProxyHosts" value="local|*.local|169.254/16|*.</pre>
169.254/16"/>
  </properties>
  <testcase name="CanDoStuff" classname="com.xpand.java.CalcTest" time="
0.003"/>
  <testcase name="CanAddNumbers" classname="com.xpand.java.CalcTest" time="
0 " />
  <testcase name="CanSubtract" classname="com.xpand.java.CalcTest" time="0"
/>
  <testcase name="CanMultiply" classname="com.xpand.java.CalcTest" time="0"
/>
</testsuite>
```

#### Notes:

 For more information regarding the execution and integration of JUnit tests with Xray please check this article.

### Integrating with Travis CI

As we saw in the previous example, we produced a JUnit report locally with the test results; it is now a matter of executing those tests in Travis CI and importing their results to your Jira instance. You can do this by simply submitting automation results to Xray through the REST API or using the Jira interface to do so. In this case, we will show you how to integrate with Travis CI using API calls.

In order to integrate Travis CI with Xray, we define the *CLIENT\_ID* and *CLIENT\_SECRET* as environment variables so they can later be used in the Xray API calls mentioned in the *.travis.yml* file.

For that please go to the configuration of your repo in Travis CI and proceed as follows.

🛛 Xray-App / tutorial-java-junit-travisci 🔘 📼						
Carrent Branches BuildHistory PullRequests	Settings		Nore options =			
General						
Build pushed branches (?)		Caller Limit concurrent jobs ??				
C Build pushed pull requests (?)		User management				
Auto Cancellation						
Auto Cancellation allows you to only run builds for the latest finish.	commits in the queue. This setting can be applied to b	ulids for Branch builds and Pull Request builds separately, Builds will only be	canceled if they are waiting to run, allowing for any running jobs to			
Auto cancel branch builds		Auto cancel pull request builds				
Environment Variables						
Customize your build using environment variables. For secur	re tips on generating private keys read our documentat	on				
CUBVT_ID	@	Available to all branches	0			
CUENT_SECRET	۵	Available to all branches	Û			
S Types secret variable has special characters like <b>B</b> .	escape them by adding $\underline{\mathcal{K}}$ in frant of each special cha	actes. For example, maker doo, would be entered as matked 1440.				

We have also added the necessary .travis.yml file to our project in GitHub in order to configure our pipeline.

On the file we can see one job definition that is compiling the code (like we did locally):

mvn clean compile test --file pom.xml

We can also see one API call to submit the results.

#### Submit results

```
curl -H "Content-Type: multipart/form-data" -X POST -u USERNAME:PASSWORD -
F "file=@target/surefire-reports/TEST-com.xpand.java.CalcTest.xml"
"https://YOUR_JIRA_SERVER/rest/raven/2.0/import/execution/junit?
projectKey=$PROJECTKEY_SERVER&testPlanKey=$TESTPLAN_SERVER"
```

With this command, you can create a new Test Execution for your tests, linked to the referred Test Plan " *COM-109.*" Tests will be auto-provisioned at the moment of the first import of results and will be reused afterward.

Once everything is configured you can start the pipeline by committing to your code repository (in our case pushing code in GitHub).

On the side of Travis CI, it will look like this:

ment Branches Build History Pull Requests	More options	
🗸 main. Adding test plan key to Xray call	~ #2 passed	🕒 Restart bolid
<ul> <li>Commit esetocela. Il</li> </ul>	ीए Ran for 56 sec	
1) Compare SizeFeldddode 0	🔅 2 days ago	
D Barcheate U		

The log that will appear in the bottom of the page in Travis CI will output all of the steps we are taking:

135	(INFO) Wolfong to campile - all classes are up to date
131	
135	[1860] ··· mivee-restrices-plugin:2.4:testHissources (default-testHissources) @ spind-test ···
1.3.2	[semented] buing platform encoding (arm-m actually) to capy filtered resources, i.e. build is platform dependent:
1.85	[1890] skip non-existing resourcedirectory memorizabilitaring-approtectual-pana-junit-transit/sectrestoressories
132	[146]
135	[1893] maves-complet-plagatin.litestcomplete (establish # spano-test
135	[TNO] Charges astected - reconstraining the monater
1.65	(MARING) File enceing has not deep set, sang platters economy one e, i.e. salls is platters oppropriate
1.32	[INVO] COMPLIED I BOUTON TILE TO /ROMA/THAVILADUILD/ATMY-ADD/CHTOTIAL-JUNE-JUNET-THAVILED/CHTOTIAL-JUNE-JUNET-THAVILED/CHTOTIAL-JUNE-JUNET-THAVILED/CHTOTIAL-JUNE-JUNET-THAVILED/CHTOTIAL-JUNE-JUNET-THAVILED/CHTOTIAL-JUNE-JUNET-THAVILED/CHTOTIAL-JUNE-JUNET-THAVILED/CHTOTIAL-JUNE-JUNET-THAVILED/CHTOTIAL-JUNET-THAVILID/CHTOTIAL-JUNET-THAVILID/CHTOTIAL-JUNET-THAVILID/CHTOTIAL-JUNET-THAVILID/CHTOTIAL-JUNET-THAVILID/CHTOTIAL-JUNET-THAVILID/CHTOTIAL-JUNET-THAVILID/CHTOTIAL-JUNET-THAVILID/CHTOTIAL-JUNET-THAVILID/CHTOTIAL-JUNET-THAVILID/CHTOTIAL-JUNET
112	Larry Then anne-scatte-charter in their Josten's a visit for the
1111	(and a second se
112	- consisting in the graphic matrix which is a support matrix devices and a support matrix of a particular support in the support of the suppo
1.1.1	mentionation from constant and a second and a
132	Distingent from monte saves central. HTML//Waves central.strong-mentional complexits.com/man/s/ann/man/man/s/ann/man/s/ann/man/s/ann/man/s/ann/man/s/ann/man/s/ann/man/s/ann/man/s/ann/man/s/ann/man/s/ann/man/s/ann/man/s/
1.00	
1.9.2	Downloading from copale-maxes-central: https://maxes-central.storage-download.gooolegois.com/maxes/ors/apache/maxes/aparefire-iunit4/3.19.1/marefire-iunit4/3.19
132	continued from continuing to the state of th
133	
1.12	
132	
133	
1.53	Bunning com.xpand.java.GalcTest
1.3.2	Tests run: 4, Failures: 0, Errers: 0, Skipped: 0, Time mlapsed: 0.042 sec - in com.xpand.java.falclast
133	
1.33	
1.5.3	
1.3.2	Tests run: 4, Failures: 0, Arrora: 0, Skipped: 0
1.3.3	
133	[190]
134	piso) kita sooass
114	1000
134	[DNO] 7024[ [344] 4.555 5
1.04	Trand i Durande et : 2005-01-07.121.141.005
134	
1.0.5	Transmission and the control of the second s In second second second second se Second second seco
	The second
	and plan resolution from writing to an internal plantation and plantation in the standard of t
120	The comment Such Shull Have come .
134	non of pay compile fast (The non-and
114	[107] B. "Onited Ton: million//frm.data", J. 2001. a sinistable. 4. "filedarmet/arefire.resorts/IEELow start ins. falciant.ml". Peters / Add. 40, 455, 114.
	Th. North. In/rest/Tawar/2.0/Import/Rescaltor/Impires/Second/Phone:Ther Second Figure Second
13.4	ects "date" exited with 0.
► 135	stere bullé cache
135	
1.35	
1.35	

This is how it looks in Xray once the test report is ingested back in, namely on the TestPlan "COM-109":

🎴 🔤	avisCI-inte	gration														
/ tot	Q Connert	Assign More	* 10 Do	In Progress	Moldow Y	$Admin ~ \forall$								<	Ouc. Deservator	å boot v
× Ootalla											× Xporter					
Type:		Test Plan				Stetus:	22	E3 (New Workflew)			Template		looue Details wit	N QR Code	· • 0	
Compose	-124 N					resources		(BOOME)			Output format		POF		~	
Labels:	N												A.most.			
× Description																
											× People					
× Tests	_										Assigneer		Drassigned			
Add Tex	Diale	fest Execution +								Test Plan Board	Reporters	- 1	Animate			
Overall De	ecution Status										Votes:		0			
											Watchers:		Stop watchin	g this issue	•	
4											V Dates					
											Created		hours and			
Total Tests	e.4										Updated.	5	hours ago			
Ŧ	Fibral										il Anto					
											Ven or Board					
10 <b>-</b>								Show 12 w entries	All Environments	<ul> <li>Columns -</li> </ul>						
	Summary	Requirements	#Test Execution	in Innue Anni	ipen Gerpe	menta Begin Bab	End Ba	in TestPlan DeVen	iosh Latest Status							
•	CanDedis.#		1	Administra	160°			None	Ph33							
0	CanAddNumber		2	Administra	107			None	Ph33							
	Cardulation		2	Advision	adar .			None	Pass							
•	Centerior		2	Approx	ator			Nove	1455							
Showing 1	to 4 of 4 antries								First Previous	Not Let						

As we can see the Test Plan has 4 Tests, all currently marked as being passed.

If we drill down to the latest Test Run, we can see the details of that specific execution:

and the Name 2005		Animan Antoneous Manimum
Started Ox: 00,Feb;22.3:50 PM () Finited Ox: 02,Feb;22.3:50 PM		Encoded by: Advisory Revision Trans.
Comment	↑ Execution Defects (0) ⊕	^ Execution Exidence (3) ⊙
Tek to add comment		
Test Details genres		
Custon Fields		
here are no Test Run Custom Pields defined.		
Test Description		
Test Type: Generic		
Definition: com apand jana CalcTest, CanSubtract		
Results		
	0.001	Durden Robe
Carried.		

## Triggering automation from Xray side

Please have a look at Integration with Automation for Jira to see some examples of how automation can be triggered from Xray side.

# References

- https://github.com/Xray-App/tutorial-java-junit-Travis CI
  https://docs.travis-ci.com/user/tutorial/
  Taking advantage of Jira Cloud built-in automation capabilities

- Overview
   Prerequisites
   Implementing tests
   Integrating with Travis CI

   Submit results

   Triggering automation from Xray side
   References