



Jenkins user plugin

This time it's  *Groovy*

Shiran Rubin
@ShiranRU

JFrog
<http://jfrog.com>

July 16, 2014

#jenkinsconf

About me

- Groovy developer in JFrog. The home of



We work with:



Jenkins

But support many others.

It's time to



There's something you should know...

- Jenkins supports Groovy:
 - Groovy Script for troubleshooting
 - Groovy post build plugin to run a groovy script when your build is done.
 - Groovy plugin to run a groovy script on selected steps of the build.



So... What's the difference?

- That's not what we're trying to achieve!



Our real goal


- Our plugin will be written in Groovy but not necessarily used for Groovy.



There's a plugin for anything

- Always wanted to be a mechanic and have models calender?
- There's a plugin for that!
- So why do we need another one?



 **Build #1 (May 21, 2010 5:18:49 PM)**

[ENABLE AUTO REFRESH](#)

[Delete this build](#)

Started 7,8 sec ago
Took [94 ms](#)

[add description](#)



No changes.



Started by user [iyoshiek021510](#)



"Put your hand on a hot stove for a minute, and it seems like an hour. Sit with a pretty girl for an hour, and it seems like a minute. That's relativity." Albert Einstein



It's mostly about needs

- Language support
- Tools
- Environment
- Fantastic communities need contributors



Your build may vary – we're all different

- Some setups can be very unique
- Some issues may be solved by modifying the cycle
- Some issues may be solved with tools or infrastructure



Don't overstep the line

- Jenkins should make life easier:
 - Prepare the environment
 - Monitor and interface with tools
 - Automate processes
 - Show you ridiculous pictures of 1970's Chuck Norris or super models
- It's complimentary, not a replacement



What can I extend?

- User management and authentication

	User Id	Namn	Senast aktiv ↑	På
	erik	Erik Ramfelt	N/A	

Ikon: [S](#) [M](#) [L](#)



What can I extend?

- UI



Build History		(trend)
● #37	May 19, 2014 9:41:03 AM	○ 🌟
● #36	Jan 30, 2014 7:21:22 PM	○ 🌟 📄
● #35	Jan 14, 2014 2:16:46 PM	○ 🌟 📄
● #34	Nov 3, 2013 4:17:29 PM	○ 🌟 📄
● #33	Nov 3, 2013 3:55:58 PM	○
● #32	Jul 25, 2013 11:05:07 AM	○ 🌟
● #31	Jul 25, 2013 11:04:01 AM	○
● #30	Jul 24, 2013 1:53:38 PM	○ 🌟
● #29	Jul 24, 2013 1:10:33 PM	○
● #28	Jul 24, 2013 1:07:08 PM	○
● #27	Jul 24, 2013 12:53:00 PM	○
● #26	Jul 23, 2012 3:33:28 PM	○ 🌟
● #25	Jun 24, 2012 9:16:49 PM	○ 🌟
● #24	Jun 13, 2012 2:15:32 PM	○
● #23	Jun 13, 2012 2:14:50 PM	○
● #22	Jun 12, 2012 11:42:29 PM	○
● #21	May 23, 2012 9:26:54 PM	○ 🌟
● #20	May 23, 2012 9:22:12 PM	○
● #19	May 3, 2012 2:19:46 PM	○

What can I extend?

- Build Processes – deploy and resolve



```
Deploying artifact: https://oss.jfrog.org/oss-release-local/org/jfrog/buildinfo/build-info-extractor/2.2.7/build-info-extractor-2.2.7-javadoc.jar
Deploying artifact: https://oss.jfrog.org/oss-release-local/org/jfrog/buildinfo/build-info-client/2.2.7/build-info-client-2.2.7.jar
Deploying artifact: https://oss.jfrog.org/oss-release-local/org/jfrog/buildinfo/build-info-vcs/2.2.7/build-info-vcs-2.2.7.pom
Deploying artifact: https://oss.jfrog.org/oss-release-local/org/jfrog/buildinfo/build-info-extractor/2.2.7/build-info-extractor-2.2.7.jar
Deploying artifact: https://oss.jfrog.org/oss-release-local/org/jfrog/buildinfo/build-info-extractor/2.2.7/build-info-extractor-2.2.7-sources.jar
Deploying artifact: https://oss.jfrog.org/oss-release-local/org/jfrog/buildinfo/build-info-vcs/2.2.7/build-info-vcs-2.2.7-javadoc.jar
Deploying artifact: https://oss.jfrog.org/oss-release-local/org/jfrog/buildinfo/build-info-client/2.2.7/build-info-client-2.2.7-sources.jar
Deploying artifact: https://oss.jfrog.org/oss-release-local/org/jfrog/buildinfo/build-info-client/2.2.7/build-info-client-2.2.7-javadoc.jar
Deploying artifact: https://oss.jfrog.org/oss-release-local/org/jfrog/buildinfo/build-info-extractor/2.2.7/build-info-extractor-2.2.7.pom
Deploying artifact: https://oss.jfrog.org/oss-release-local/org/jfrog/buildinfo/build-info-api/2.2.7/build-info-api-2.2.7-javadoc.jar
Deploying artifact: https://oss.jfrog.org/oss-release-local/org/jfrog/buildinfo/build-info-api/2.2.7/build-info-api-2.2.7.pom
Deploying artifact: https://oss.jfrog.org/oss-release-local/org/jfrog/buildinfo/build-info-vcs/2.2.7/build-info-vcs-2.2.7.jar
Deploying artifact: https://oss.jfrog.org/oss-release-local/org/jfrog/buildinfo/build-info-vcs/2.2.7/build-info-vcs-2.2.7-sources.jar
Deploying artifact: https://oss.jfrog.org/oss-release-local/org/jfrog/buildinfo/build-info-api/2.2.7/build-info-api-2.2.7-sources.jar
Deploying artifact: https://oss.jfrog.org/oss-release-local/org/jfrog/buildinfo/build-info-client/2.2.7/build-info-client-2.2.7.pom
Deploying artifact: https://oss.jfrog.org/oss-release-local/org/jfrog/buildinfo/build-info-api/2.2.7/build-info-api-2.2.7.jar
Deploying build info to: https://oss.jfrog.org/api/build

BUILD SUCCESSFUL
```

Resolve artifacts from Artifactory ?

Artifactory server

Resolution releases repository

Resolution snapshots repository

Override default resolution credentials ?

What can I extend?

- Tooling

Artifactory client tool:



Project build-info-extractor

Build only the build info project with Gradle



[Artifactory Build Info](#)



[Workspace](#)



[Recent Changes](#)

Permalinks

- [Last build \(#37\), 1 mo 26 days ago](#)
- [Last stable build \(#37\), 1 mo 26 days ago](#)
- [Last successful build \(#37\), 1 mo 26 days ago](#)
- [Last failed build \(#29\), 11 mo ago](#)
- [Last unsuccessful build \(#31\), 11 mo ago](#)

What can I extend?

And many many more...

You can even create new extension points!



Jenkins thrives on plugins

- 1113 and growing (13.7.2014)
(Jenkins - two years ago 600+)



I'm convinced



My first instinct – RTFM



Problem

- Most documentations and talks are about:
 - Write in Java
 - Build with Maven
 - Use Jelly for UI – explanation soon...





BUT



BUT WHY

There's a better way just wait and see



Tools we'll need

- IDE - I use IntelliJ IDEA:
Because it kicks ass and can totally makes life easier.
- Build tool
- Programming language
- UI Framework



Tools we'll need

Why Groovy?

Build on top of Java

Functional language

Dynamic language

Runs on JVM

- Groovy 1.8.9



Groovy tease

- No ;
- No return
- def something – encapsulation on type
- Map → [:]
- Smart collections → each, collect, find, any, findAll ...
- “Hello world this is $\${something}$ ”
- def myFunction = {param -> 'hello' } (Java 7)

And so much more!



Tools we'll need

- Gradle – our building tool



Why Gradle?

- Build on top of groovy
- Customize jobs and tasks to your need.
- Maven users will feel at home:
 - Declarative as Maven
 - Same dependencies
 - Same project structure
 - Similar life cycle



JUST MORE FLEXIBLE

Question – This will be in the Trivia

- Does anyone know why Jenkins balls are blue and red?



Jenkins [ENABLE AUTO REFRESH](#) [add description](#)

[New Item](#) [People](#) [Build History](#) [Project Relationship](#) [Check File Fingerprint](#) [Manage Jenkins](#) [Credentials](#)

Build Queue
No builds in the queue.

Build Executor Status

#	Status
1	Idle
2	Idle

All +

S	W	Name ↓	Last Success	Last Failure	Last Duration	
		Balls	18 hr - #4	N/A	4 min 46 sec	
		fail	22 hr - #2	1 hr 55 min - #24	0.58 sec	
		Frog Balls	1 day 20 hr - #24	1 day 20 hr - #22	13 sec	
		test	35 min - #35	1 hr 35 min - #31	3.5 sec	

Icon: [S](#) [M](#) [L](#)

[Legend](#) [RSS for all](#) [RSS for failures](#) [RSS for just latest builds](#)

[Help us localize this page](#) Page generated: Jul 3, 2014 12:01:09 PM [REST API](#) [Jenkins ver. 1.570](#)

Traffic lights



Especially for us - Green balls plugin



- Java + Maven
- Availability – Jenkins plugin management
- All Jenkins plugins in one place
- Install + Restart

Jenkins Plugin Manager

Filter:

Updates Available Installed Advanced

Install	Name	Version
<input checked="" type="checkbox"/>	Green Balls Changes Hudson to use green balls instead of blue for successful builds	1.14

[Install without restart](#) [Download now and install after restart](#)

Jenkins

ENABLE AUTO REFRESH [add description](#)

[New Item](#) [People](#) [Build History](#) [Project Relationship](#) [Check File Fingerprint](#) [Manage Jenkins](#) [Credentials](#)

Build Queue
No builds in the queue.

Build Executor Status

#	Status
1	Idle
2	Idle

Icon: [S](#) [M](#) [L](#)

All	S	W	Name	Last Success	Last Failure	Last Duration
			Ball	2 days 0 hr - #5	N/A	42 sec
			fail	6 days 10 hr - #2	2 days 0 hr - #76	0.58 sec
			Frog Balls	7 days 7 hr - #24	7 days 7 hr - #22	13 sec
			test	6 hr 57 min - #41	5 days 12 hr - #31	3.6 sec

[Legend](#) [RSS for all](#) [RSS for failures](#) [RSS for just latest builds](#)

First plugin - Frog balls

- Change weather balls
TO
Frog mood balls



Jenkins

search

ENABLE AUTO REFRESH

add description

New Item

People

Build History

Project Relationship

Check File Fingerprint

Manage Jenkins

Credentials

Build Queue

No builds in the queue.

Build Executor Status

#	Status
1	Idle
2	Idle

All	S	W	Name	Last Success	Last Failure	Last Duration
	🟢	☀️	Balls	2 days 0 hr - #5	N/A	42 sec
	🔴	☁️	fail	6 days 10 hr - #2	2 days 0 hr - #76	0.58 sec
	🟢	☁️	Frog Balls	7 days 7 hr - #24	7 days 7 hr - #22	13 sec
	🟢	☀️	test	6 hr 57 min - #41	5 days 12 hr - #31	3.6 sec

Icon: S M L

Legend

RSS for all

RSS for failures

RSS for just latest builds

Help us localize this page

Page generated: Jul 8, 2014 11:22:49 PM

REST API

Jenkins ver. 1.570

My First Plugin – add filter

- Extension point:
 - Filter - man in the middle.
- Servlet hack
- Basic plugin – no UI
- Change exiting objects



Our goal



Java

Groovy

Maven

Gradle (Based on Groovy)

Where do I start?



Setting up the environment

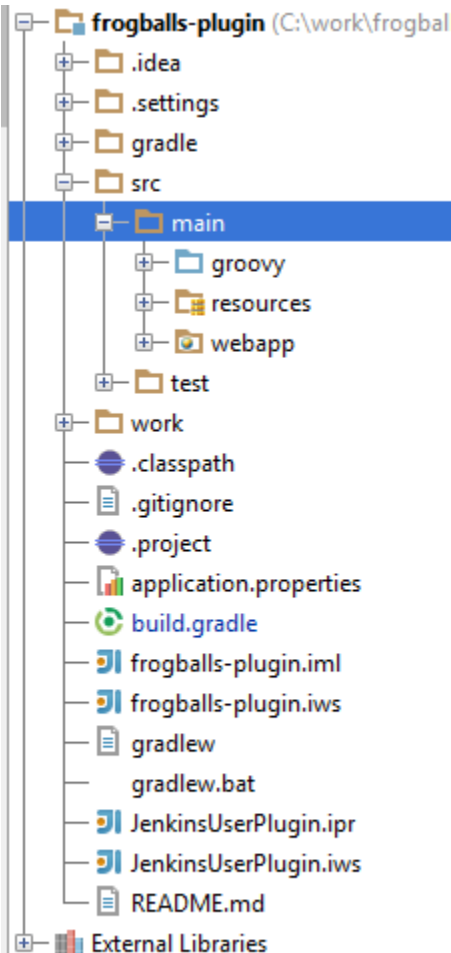
- Plugin Skeleton Generator:
<http://plugin-generator.jenkins-ci.org/>
- Note: We will need to replace the pom file with build.gradle file
- GroupId – hudson.plugins.MY_COOL_PI



Basic project structure



- build.gradle (descriptor)
- src/main
 - groovy (plugin classes)
 - resources (UI files)
 - webapp (static resources – images etc.)



Comparing – Problem!

Jenkins supports jpi/hpi packaging formats.

How do I package my project with Gradle??



```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <parent>
    <groupId>org.jenkins-ci.plugins</groupId>
    <artifactId>plugin</artifactId>
    <version>1.440</version>
  </parent>
  <artifactId>greenballs</artifactId>
  <packaging>hpi</packaging>
  <version>1.15-SNAPSHOT</version>
  <name>Green Balls</name>
  <description>Because green balls are better than blue</description>
  <url>http://wiki.jenkins-ci.org/display/JENKINS/Green+Balls</url>
  <inceptionYear>2009</inceptionYear>

  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
  </properties>

  <developers>
    <developer>
      <id>asgeir</id>
      <email>asgeir@twingine.no</email>
      <name>Asgeir Storesund Nilsen</name>
      <timezone>1</timezone>
    </developer>
  </developers>
  <licenses>
    <license>
      <distribution>repo</distribution>
      <name>The MIT License</name>
      <url>http://www.opensource.org/licenses/mit-license.php</url>
    </license>
  </licenses>
</project>
```



What's JPI/HPI?

- Stands for Jenkins/Hudson plugin
- Jenkins plugin packaging
- Zip of the entire project with the dependencies



Name	Size	Packed Size	Modified	Created	Accessed	Attributes	Encrypted	Comment
META-INF	183	141	2014-07-11 15:54			D	-	
WEB-INF	2 868 877	2 535 039	2014-07-11 15:54			D	-	
help-globalConfig-color.html	127	109	2014-07-06 14:04				-	
help-globalConfig-room.html	78	75	2014-07-06 14:00				-	
help-globalConfig-token.html	57	54	2014-07-06 13:58				-	
help-globalConfig-username.html	57	51	2014-07-06 21:24				-	
help.html	123	99	2014-07-06 16:12				-	

Gradle JPI plugin

Someone already did it for me!

- <https://github.com/erdi/gradle-jpi-plugin>
- <https://wiki.jenkins-ci.org/display/JENKINS/Gradle+JPI+Plugin>



Code!



First plugin conclusions

- Shorter code
- More understandable
- Cleaner
- Easy to maintain and extend

- What about deployment?? We'll see it later on.



Let's kick it up a notch – Second plugin

- Who doesn't know HipChat?



The screenshot shows the HipChat web interface. At the top, there are tabs for 'Lobby', 'Scranton sch...', 'Pam H.', and 'Mich...'. The main chat area displays a conversation between Jim H. and Pam H. on October 25th. A blue callout bubble points to a red message in the chat history, stating: "Break out into a one-on-one chat. New messages will show up in red so you don't miss them." On the right side, there is a profile card for Pam Halpert, including her photo, name, status (Available), role (Sales), and email (pam@dundermifflin.com). A blue callout bubble points to this profile card, stating: "Each team member has a photo and contact info." At the bottom of the chat window, there is a dark grey notification bubble for Michael S. with the message "What's for lunch today?". A blue callout bubble points to this notification, stating: "New messages will trigger these handy desktop notifications. Click it and go right to that chat." The bottom of the interface shows a text input field and a 'Send' button.

HipChat

- Acquired by Atlassian.
- Free messenger.
- REST API
- Already has a Java plugin for Jenkins – So we need one in Groovy!



Our goal

- Send a HipChat notification with:
 - Project name
 - Build number
 - Build status
- Extension point:
 - UI – configuration
 - Build Processes



<http://www.hipchat.com/grGF8OAZ1>



Lets start with the UI

What is it good for? Configuration!

- API Token – auth
- Room Id – send the message to
- Color – just for fun



UI - Jenkins configuration

- Global configuration
- Build configuration
- Saved as XML – No DB



Jelly sounds nice right?



© Getty Images

Lets examine Jelly

- Templates framework
- Executable XML
- Jelly → Java → XML (our configuration)



Jelly disadvantages

- No debugging
- Jelly → Java → XML
- Very hard to extend



Is there really always room for jelly?



Replace Jelly with Groovy!

- You can debug!
- Groovy → Groovy → XML
- Easy to extend



Our second plugin will use

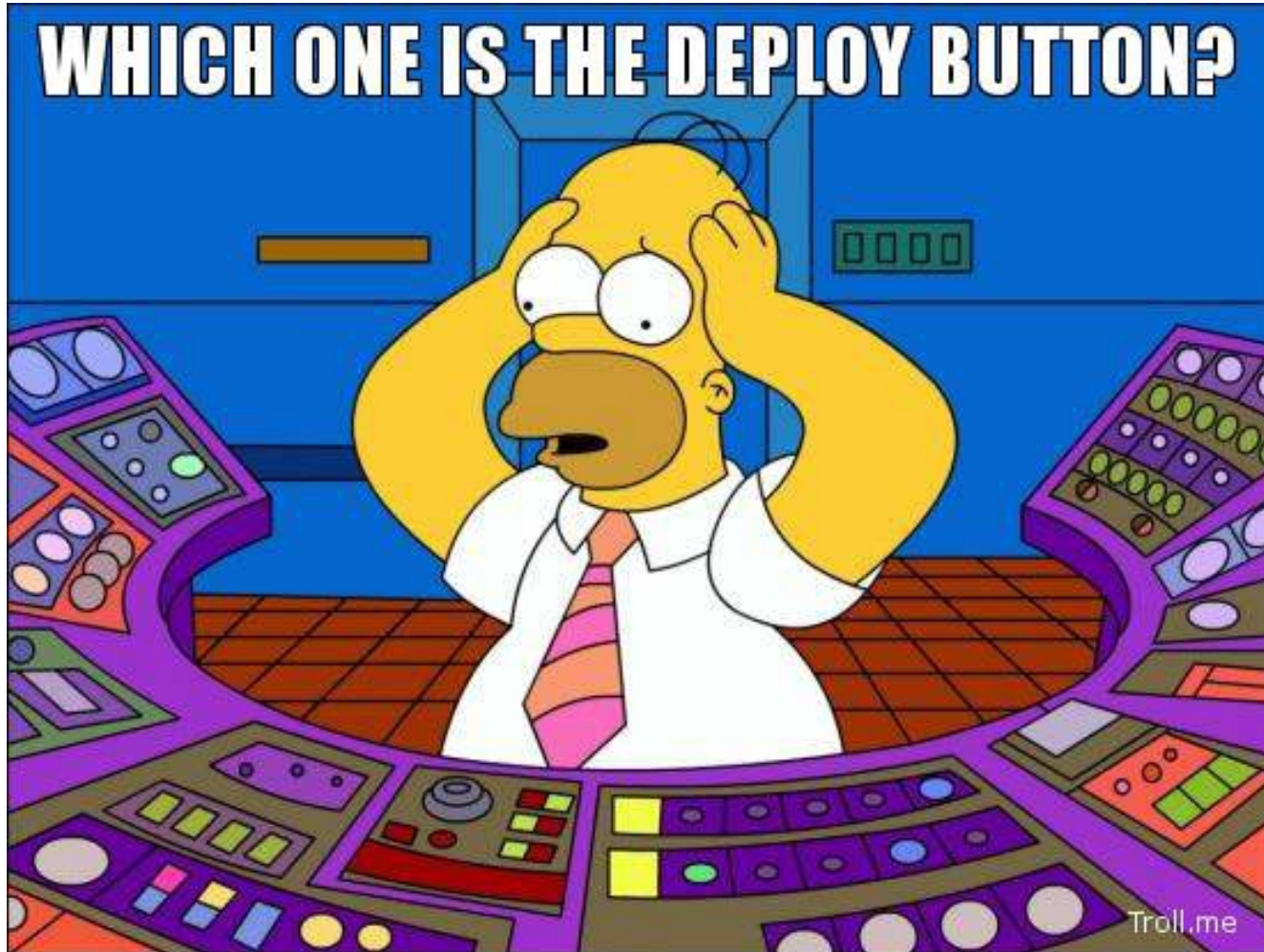


Java	Groovy
Jelly	Groovy
Maven	Gradle (Based on Groovy)

Once again...



We're ready! DEPLOY!



Development

- Locally
- `gradlew clean build/jpi`
- Add jpi file to plugins folder



That's good but...

- No one will use an Android app which is not in the Google play!



Ready? Let's go!

- Deploy to Jenkins-ci.org:
- <https://wiki.jenkins-ci.org/display/JENKINS/Hosting+Plugins>



References

- <http://thingsyou didnt know about jenkins. tumblr. com/ post/ 26585787635/ built- in- groovy- scrip- ting>
- <https://github.com/erdi/gradle-jpi-plugin>
- <https://wiki.jenkins-ci.org/display/JENKINS/Gradle+JPI+Plugin>
- <https://github.com/jenkinsci/greenballs-plugin>
- <http://groovy.codehaus.org/>
- All examples can be found in here:
<https://github.com/shiranr>
- Baruch Sadogursky: @jbaruch





ANY
QUESTIONS
?

Thank You To Our Sponsors

Platinum



Gold



Silver

