

GROOVY DSL ДЛЯ PLUGIN-ОВ:



Расширения человеческим языком

Joker



Конференция по Java-технологиям

Анти-Шипилёвский слайд, а-ля «дисклеЙмер»

Анти-Шипилёвский слайд, а-ля «дисклеЙмер»

- Доклад про удовольствие, удовлетворение и отсутствие проблем (почти)

Анти-Шипилёвский слайд, а-ля «дисклеймер»

- Доклад про удовольствие, удовлетворение и отсутствие проблем (почти)
- Вы наверняка всё и так знаете, так что чисто поржать

Анти-Шипилёвский слайд, а-ля «дисклеЙмер»

- Доклад про удовольствие, удовлетворение и отсутствие проблем (почти)
- Вы наверняка всё и так знаете, так что чисто поржать
- Докладчик тоже чисто поржать

Анти-Шипилёвский слайд, а-ля «дисклеЙмер»

- Доклад про удовольствие, удовлетворение и отсутствие проблем (почти)
- Вы наверняка всё и так знаете, так что чисто поржать
- Докладчик тоже чисто поржать
- JVM перформанса, GC, heap и прочих ужасов не будет

Анти-Шипилёвский слайд, а-ля «дисклеймер»

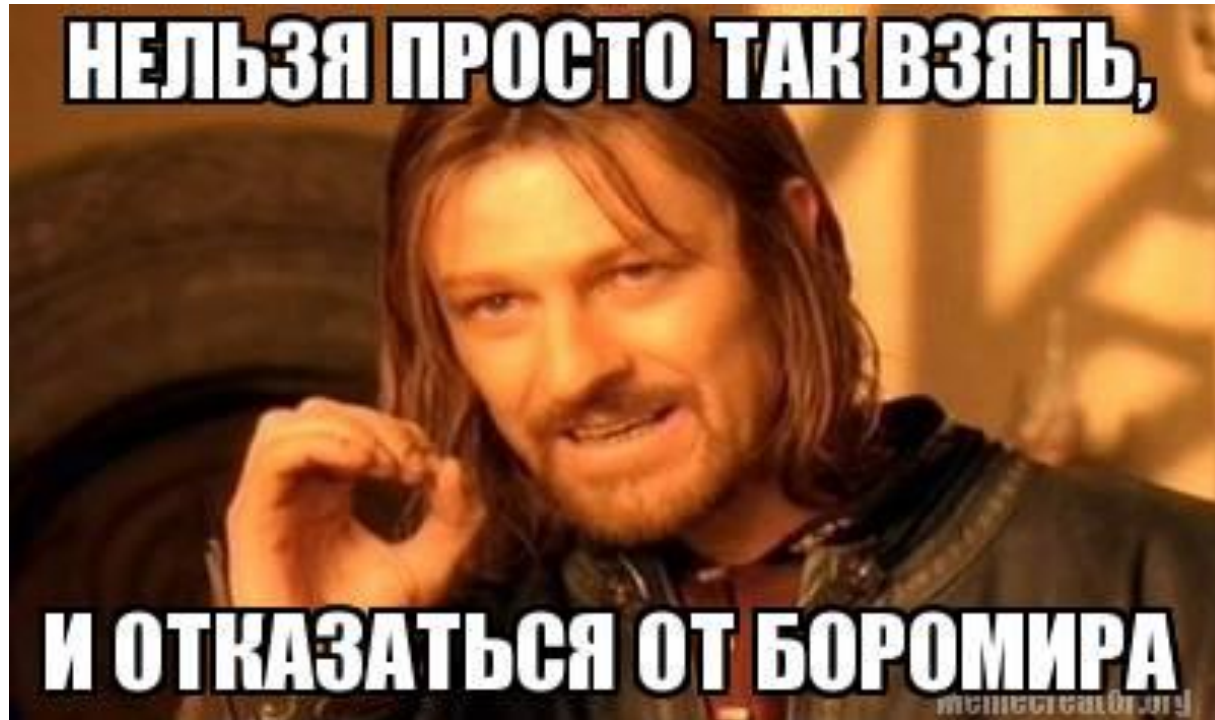
- Доклад про удовольствие, удовлетворение и отсутствие проблем (почти)
- Вы наверняка всё и так знаете, так что чисто поржать
- Докладчик тоже чисто поржать
- JVM перформанса, GC, heap и прочих ужасов не будет

***Да! Пушистые котики и розовые пони –
это к нам!***

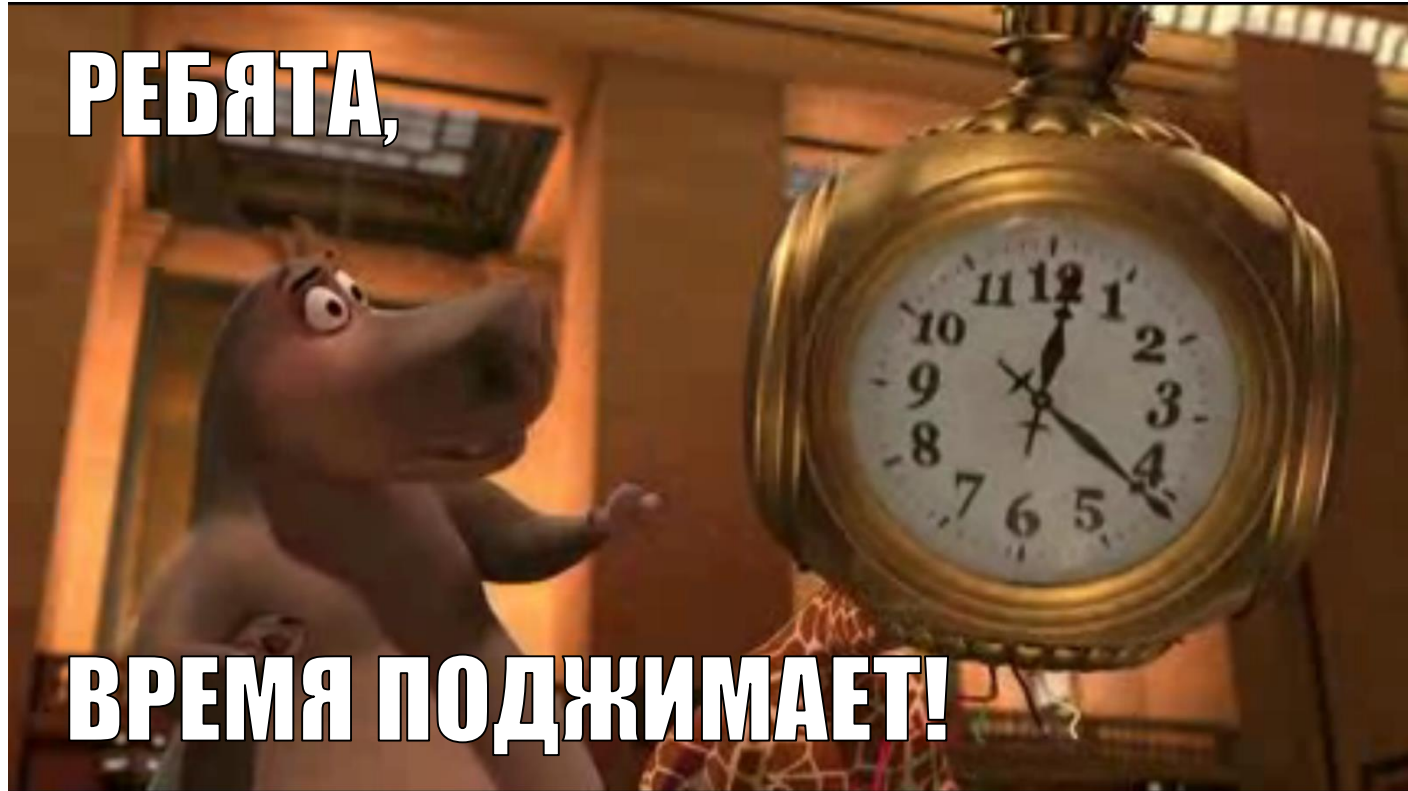




Ищо дисклеймер



И последний дисклеймер



РЕБЯТА,

ВРЕМЯ ПОДЖИМАЕТ!

Джей-Бáрух

github.com/jbaruch

@jbaruch



Developer кто?

ТРУДОВОЙ ДОГОВОР (КОНТРАКТ)

много скучного текста...

7. Работник должен выполнять следующие обязанности:

1. Зависать с девелоперами и сисадминами

2. Ездить по конференциям и рассказывать об этом

Developer кто?



Какая-какая лягушка?



Какая-какая лягушка?



Какая-какая лягушка?



Какая-какая лягушка?



© Caters News Agency

artifactory



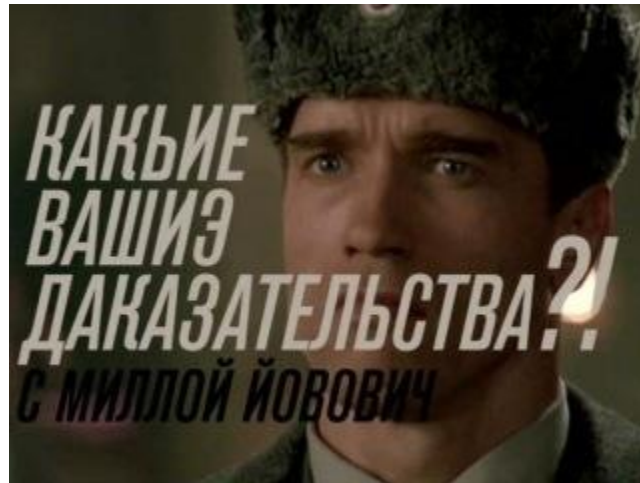
Artifactory щито?

Лучишкѣ

РЕНОВАТОРѣ

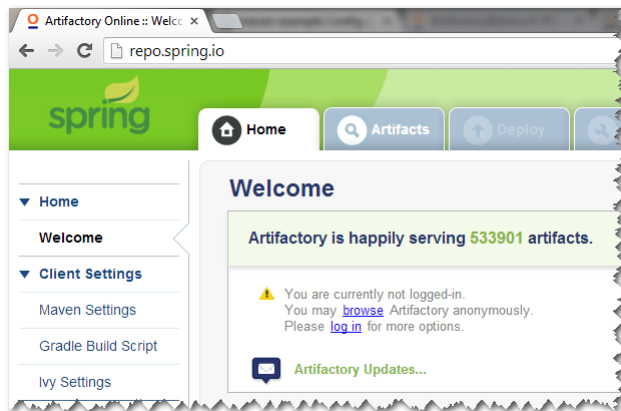
artifactory

Artifactory щито?

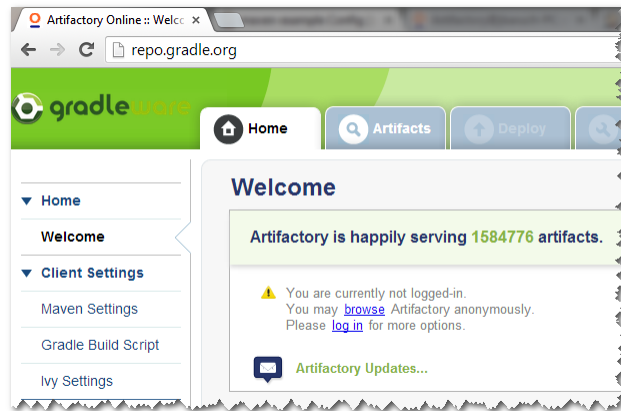
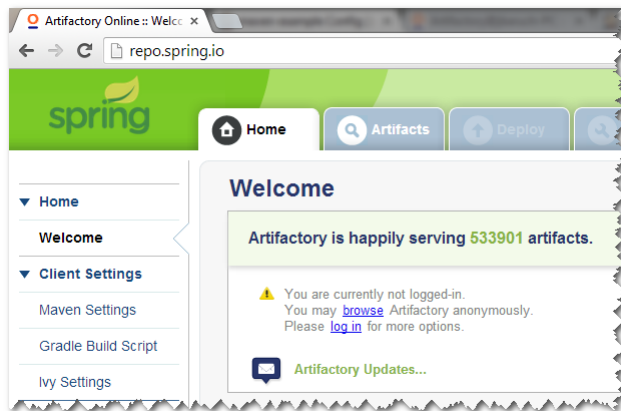


Усы, лапы, хвост...

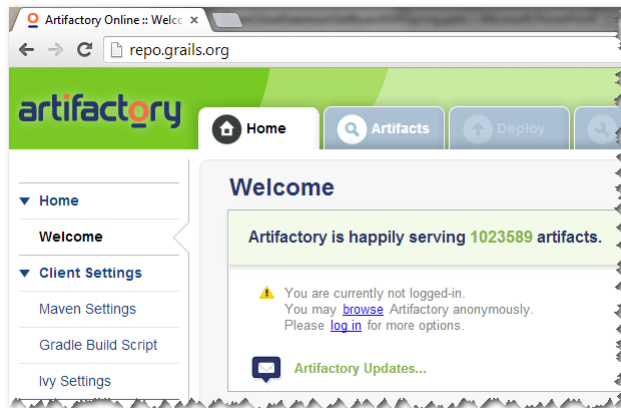
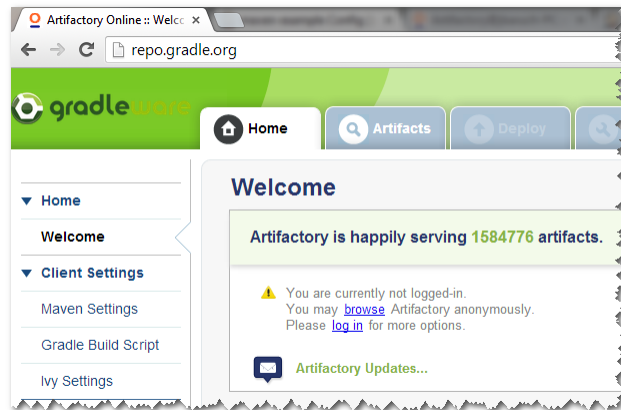
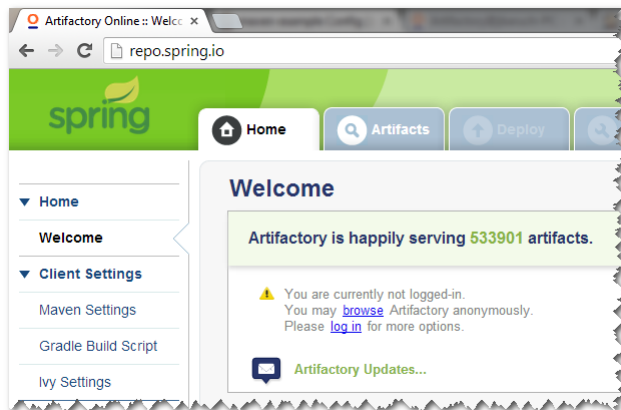
Усы, лапы, хвост...



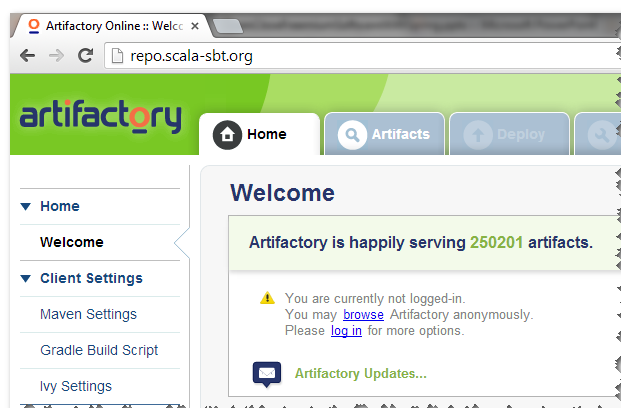
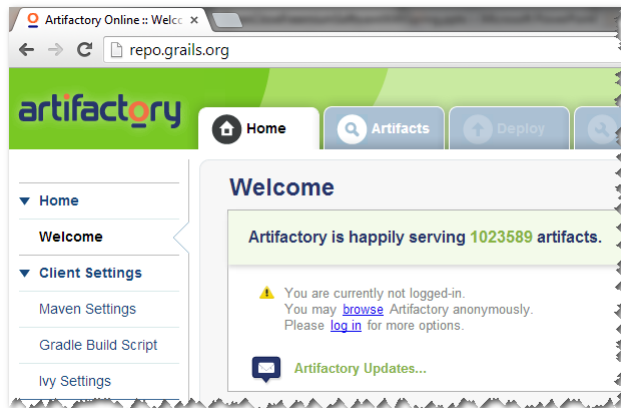
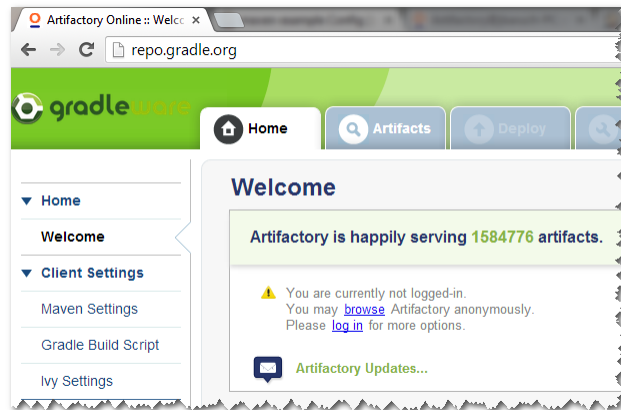
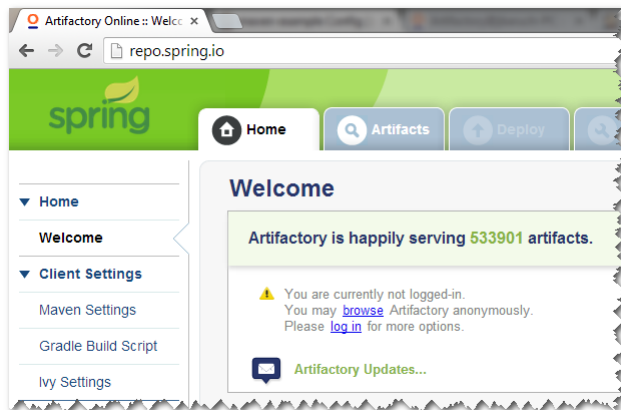
Усы, лапы, хвост...



Усы, лапы, хвост...



Усы, лапы, хвост...



Опросик замутим!



Опросик замутим!

Я пользуюсь бинарным
репозиторием:



Опросик замутим!

Я пользуюсь бинарным
репозиторием:

☒ Ясный перец, Artifactory!



Опросик замутим!

Я пользуюсь бинарным репозиторием:

- ☑ Ясный перец, Artifactory!
- ☑ Стыдно, конечно, но пока еще Nexus.



Опросик замутим!

Я пользуюсь бинарным репозиторием:

- ☑ Ясный перец, Artifactory!
- ☑ Стыдно, конечно, но пока еще Nexus.
- ☑ У меня 1995 год, у меня Archiva.



Опросик замутим!

Я пользуюсь бинарным репозиторием:

- ☑ Ясный перец, Artifactory!
- ☑ Стыдно, конечно, но пока еще Nexus.
- ☑ У меня 1995 год, у меня Archiva.
- ☑ Я пещерный человек. Ась? Бинарный кто?



Опросик замутим!

Я пользуюсь бинарным репозиторием:

- ☑ Ясный перец, Artifactory!
- ☑ Стыдно, конечно, но пока еще Nexus.
- ☑ У меня 1995 год, у меня Archiva.
- ☑ Я пещерный человек. Ась? Бинарный кто?

Итого: 146%

Одобрено
Чуровым!



О чем это я...



Мы, конечно, фантазёры...

[Spaces](#)[People](#)[Create](#)

[Pages](#) / ... / [Using Artifactory](#)

Artifactory REST API

Added by Yoav Landman, last edited by Shay Yaakov on Aug 29, 2013 ([view change](#)) [show comment](#)

TABLE OF CONTENTS

- **BUILDS**
 - All Builds
 - Build Runs
 - Build Info
 - Builds Diff
 - Build Promotion
 - Delete Builds
 - Build Rename
- **ARTIFACTS & STORAGE**
 - Folder Info
 - File Info

- Item Last Modified
- Item Properties
- Set Item Properties
- Delete Item Properties
- Retrieve Artifact
- Retrieve Latest Artifact
- Retrieve Build Artifacts Archive
- Trace Artifact Retrieval
- Archive Entry Download
- Create Directory
- Deploy Artifact
- Deploy Artifact by Checksum
- Deploy Artifacts from Archive
- File Compliance Info
- Delete Item
- Copy Item
- Move Item
- Scheduled Replication Status
- Pull/Push Replication
- Artifact Sync Download (Deprecated)
- Folder Sync (Deprecated)
- File List
- SEARCHES
 - Artifact Search (Quick Search)
 - Archive Entry Search (Class Search)
 - GAVC Search
 - Property Search

- Checksum Search
- Bad Checksum Search
- Artifacts Not Downloaded Since
- Artifacts Created in Date Range
- Pattern Search
- Builds for Dependency
- License Search
- Artifact Version Search
- Artifact Latest Version Search
- Build Artifacts Search
- SECURITY
 - Get Users
 - Get User Details
 - Create or Replace User
 - Update User
 - Delete User
 - Get Groups
 - Get Group Details
 - Create or Replace Group
 - Update Group
 - Delete Group
 - Get Permission Targets
 - Get Permission Target Details
 - Create or Replace Permission Target
 - Delete Permission Target
 - Effective Item Permissions
 - Security Configuration

- Save Security Configuration (Deprecated)
- REPOSITORIES
 - Get Repositories
 - Repository Configuration
 - Create or Replace Repository Configuration
 - Update Repository Configuration
 - Delete Repository
 - Remote Repository Configuration
 - Calculate YUM Repository Metadata
 - Calculate NuGet Repository Metadata
 - Calculate Maven Index
 - Calculate Maven Metadata
- SYSTEM & CONFIGURATION
 - System Info
 - System Health Ping
 - General Configuration
 - Save General Configuration
 - Version and Add-ons information
- PLUGINS
 - Execute Plugin Code
 - Retrieve All Available Plugin Info
 - Retrieve Plugin Info Of A Certain Type
 - Retrieve Build Staging Strategy
 - Execute Build Promotion
- IMPORT & EXPORT
 - Import Repository Content
 - Import System Settings Example

- Full System Import
- Export System Settings Example
- Export System

- Full System Import
- Export System Settings Example
- Export System

Но не до такой-же степени!

«У нас весьма хитро [REDACTED] юзеры»



– ЗАПЛАНИРОВАННЫЕ ЧИСТКИ

- ЗАПЛАНИРОВАННЫЕ ЧИСТКИ
- СВОЙ СОБСТВЕННЫЙ
СЕКЬЮРИТИ

- ЗАПЛАНИРОВАННЫЕ ЧИСТКИ
- СВОЙ СОБСТВЕННЫЙ
СЕКЬЮРИТИ
- ИЗМЕНЕНИЯ НА ЛЕТУ

- ЗАПЛАНИРОВАННЫЕ ЧИСТКИ
- СВОЙ СОБСТВЕННЫЙ
СЕКЬЮРИТИ
- ИЗМЕНЕНИЯ НА ЛЕТУ
- ПОДМЕНА КОНТЕНТА

- ЗАПЛАНИРОВАННЫЕ ЧИСТКИ
- СВОЙ СОБСТВЕННЫЙ
СЕКЬЮРИТИ
- ИЗМЕНЕНИЯ НА ЛЕТУ
- ПОДМЕНА КОНТЕНТА
- РЕАКЦИЯ НА СОБЫТИЯ

- ЗАПЛАНИРОВАННЫЕ ЧИСТКИ
- СВОЙ СОБСТВЕННЫЙ
СЕКЬЮРИТИ
- ИЗМЕНЕНИЯ НА ЛЕТУ
- ПОДМЕНА КОНТЕНТА
- РЕАКЦИЯ НА СОБЫТИЯ
- ХИТРЫЙ ПОИСК

- СВОЙ СОБСТВЕННЫЙ
СЕКЬЮРИТИ
- ИЗМЕНЕНИЯ НА ЛЕТУ
- ПОДМЕНА КОНТЕНТА
- РЕАКЦИЯ НА СОБЫТИЯ
- ХИТРЫЙ ПОИСК
- НОВЫЕ КОМАНДЫ REST

- ИЗМЕНЕНИЯ НА ЛЕТУ
- ПОДМЕНА КОНТЕНТА
- РЕАКЦИЯ НА СОБЫТИЯ
- ХИТРЫЙ ПОИСК
- НОВЫЕ КОМАНДЫ REST
- СВОЯ ЛОГИКА ПРИ ПРОМОТИРОВАНИИ

- ПОДМЕНА КОНТЕНТА
- РЕАКЦИЯ НА СОБЫТИЯ
- ХИТРЫЙ ПОИСК
- НОВЫЕ КОМАНДЫ REST
- СВОЯ ЛОГИКА ПРИ
ПРОМОТИРОВАНИИ
- НЕСТАНДАРТНАЯ METADATA

- РЕАКЦИЯ НА СОБЫТИЯ
- ХИТРЫЙ ПОИСК
- НОВЫЕ КОМАНДЫ REST
- СВОЯ ЛОГИКА ПРИ
ПРОМОТИРОВАНИИ
- НЕСТАНДАРТНАЯ METADATA
- И ПРОЧИЙ АД...



НЕЛЬЗЯ ПРОСТО ТАК ВЗЯТЬ,

И ПРЕДУГАДАТЬ ВСЕ

ПОТРЕБНОСТИ ПОЛЬЗОВАТЕЛЯ

**НА ТЕБЕ DSL, САМ ПИШИ
СВОЮ ЕРУНДУ**





- ПРИВЫЧНО ДЛЯ ДЖАВИСТОВ!

- ПРИВЫЧНО ДЛЯ ДЖАВИСТОВ!
- ПРОСТОЙ DSL!

- ПРИВЫЧНО ДЛЯ ДЖАВИСТОВ!
- ПРОСТОЙ DSL!
- ПРОСТОЙ ДЕПЛОЙМЕНТ!

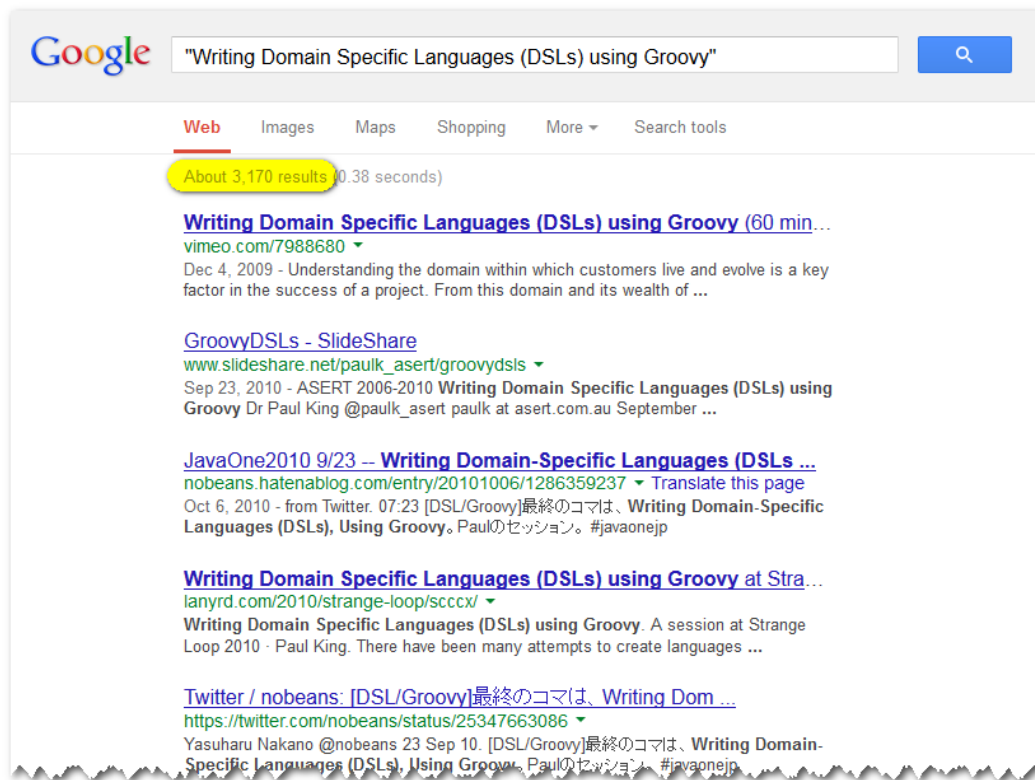
- ПРИВЫЧНО ДЛЯ ДЖАВИСТОВ!
- ПРОСТОЙ DSL!
- ПРОСТОЙ ДЕПЛОЙМЕНТ!
- И НИКАКИХ ПЕРЕЗАГРУЗОК!

- ПРИВЫЧНО ДЛЯ ДЖАВИСТОВ!
- ПРОСТОЙ DSL!
- ПРОСТОЙ ДЕПЛОЙМЕНТ!
- И НИКАКИХ ПЕРЕЗАГРУЗОК!

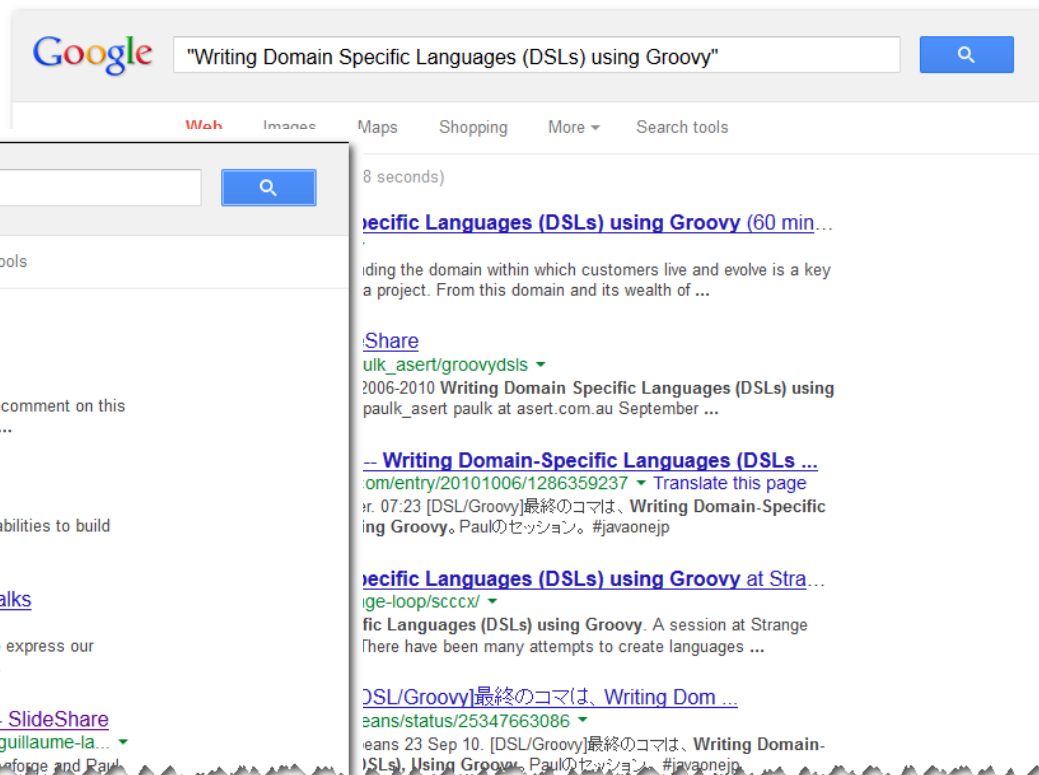
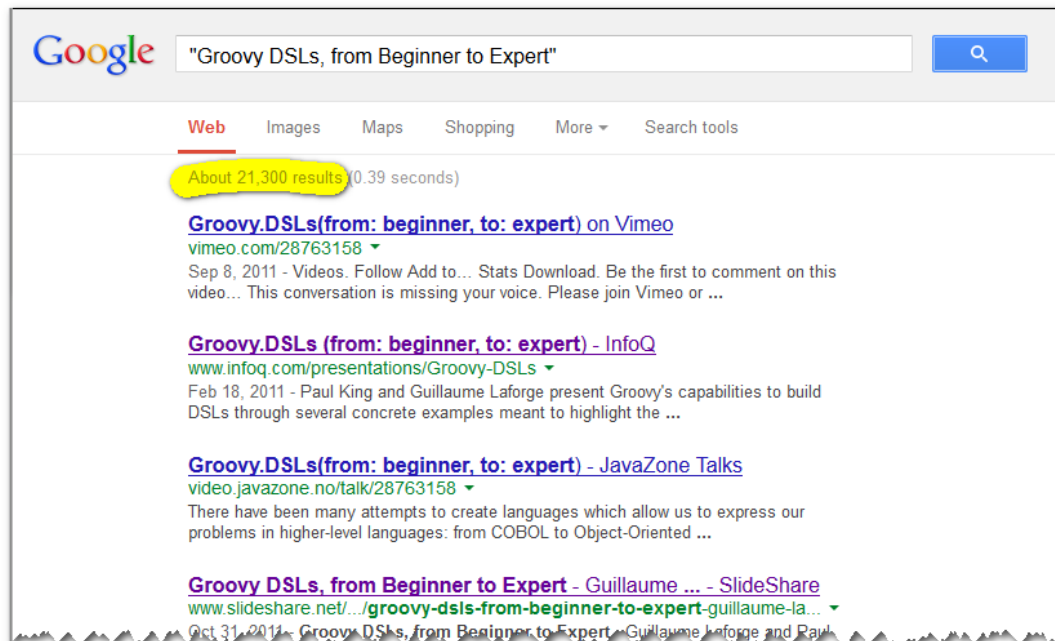


Спорим, вы всё знаете про DSL?

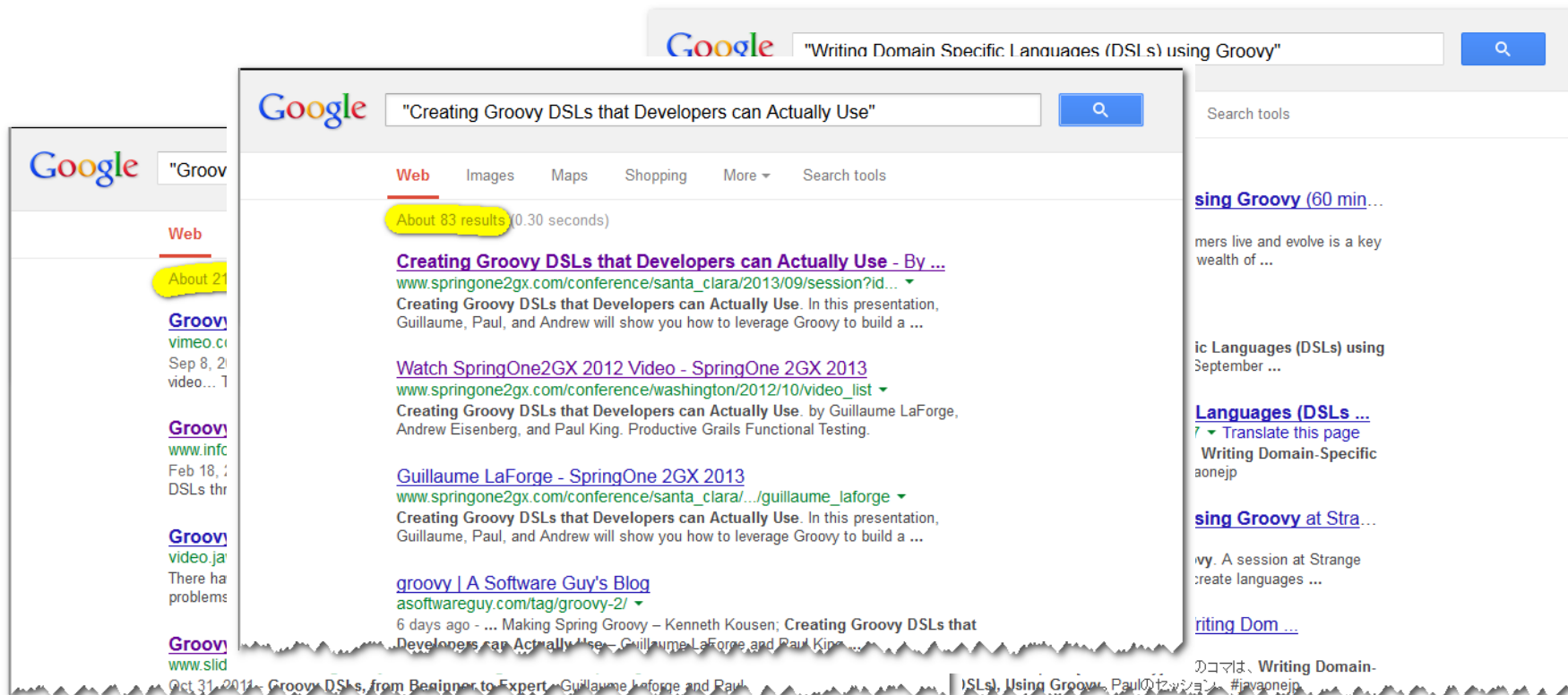
Спорим, вы всё знаете про DSL?



Спорим, вы всё знаете про DSL?



Спорим, вы всё знаете про DSL?



Ну ладно, расскажу...

Ну ладно, расскажу...

```
import static mars.Direction.*;
import mars.Robot;
public class Command{
    public static void main(String[] args){
        Robot robot= new Robot();
        robot.move(left);
    }
}
```

Ну ладно, расскажу...

```
import static mars.Direction.*;
import mars.Robot;
public class Command{
    public static void main(String[] args){
        Robot robot= new Robot();
        robot.move(left);
    }
}
```

```
move left
```


Л. Д. Б. 1938

**КТО УМЕН,
А КТО ДУРАК!
ОДИН ЗА КНИГУ,
ДРУГОЙ
В КАБАК**



Copyrighted Material

Community Experience Distilled

Groovy for Domain-Specific Languages

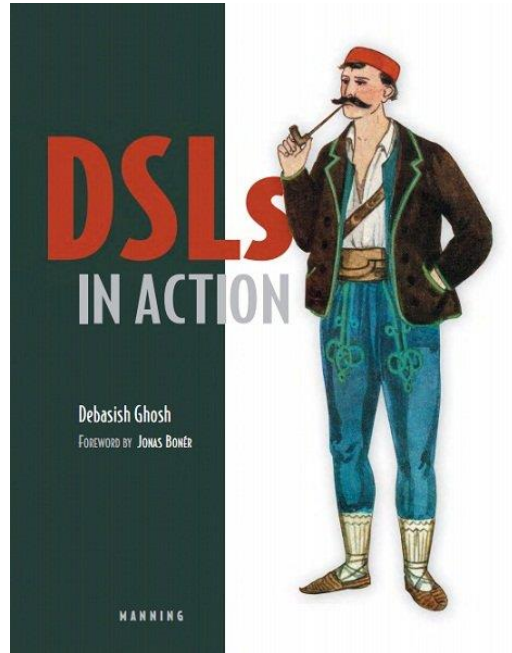
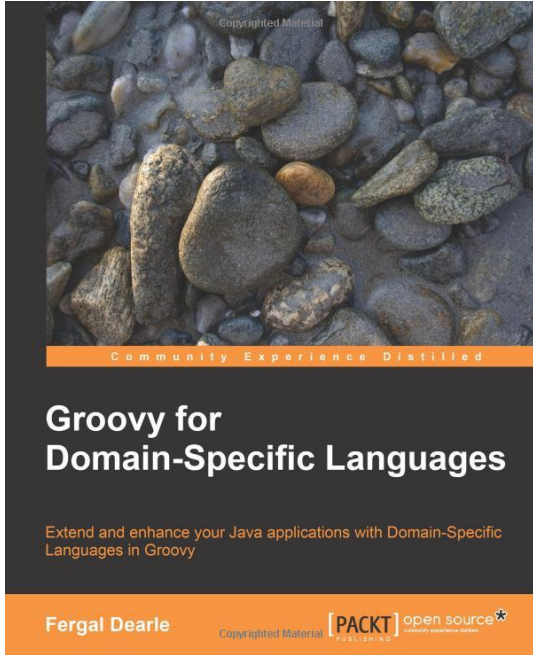
Extend and enhance your Java applications with Domain-Specific
Languages in Groovy

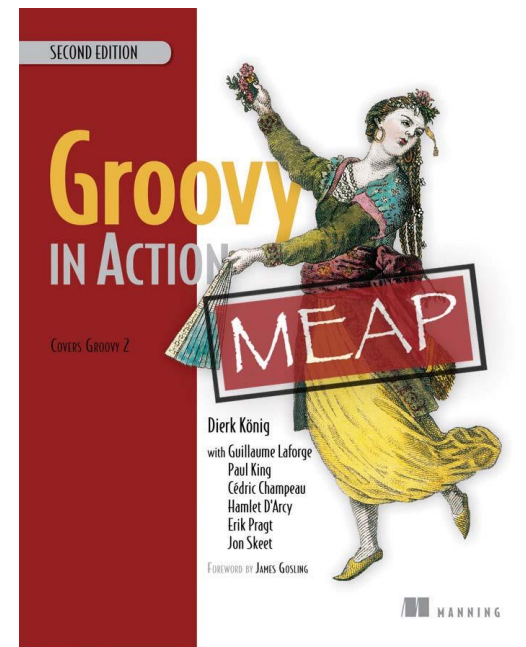
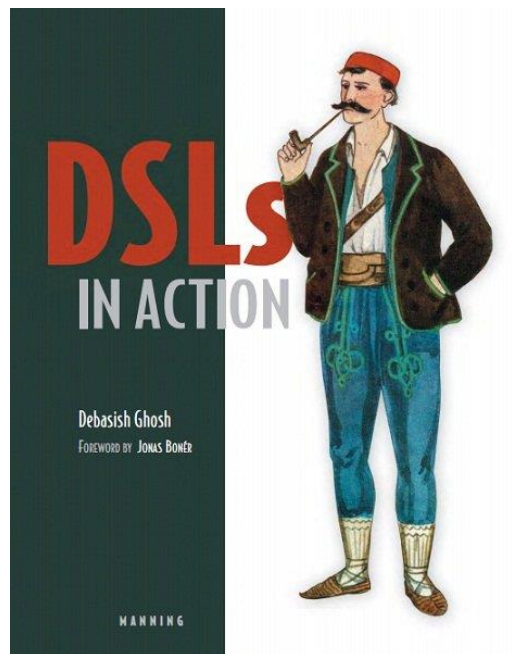
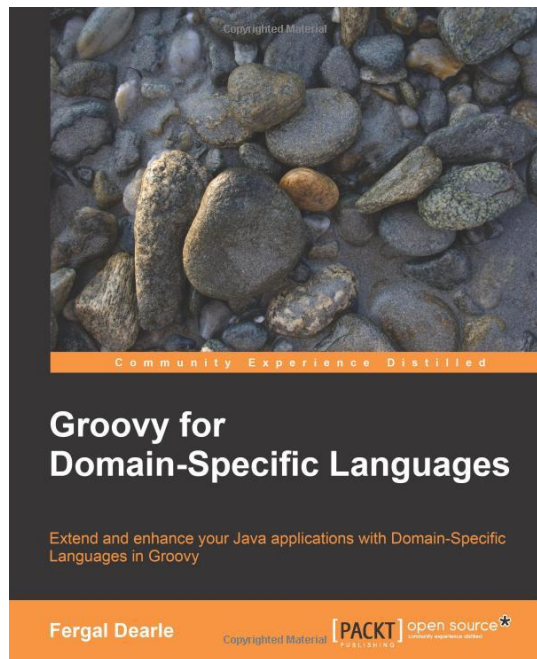
Fergal Dearle

Copyrighted Material



open source*
community experience distilled





Chapter 14: Working with XML and JSON
Chapter 15: Interacting with Web Services
Chapter 16: Integrating Groovy
Chapter 17: Unit testing with Groovy
Chapter 18: Concurrent Groovy with GPar
Chapter 19: Domain-Specific Languages
19.1 Groovy's flexible nature
19.1.1 Back on parentheses omission
19.2 Variable, constant and method injection
19.2.1 Injecting constants through the binding
19.2.2 Adding imports and static imports automatically
19.2.3 Injecting methods into a script
19.2.4 Adding closures to the binding
19.3 Adding properties to numbers
19.4 Leveraging named-arguments
19.5 Command chains
19.6 Your own control structures
19.7 Context switching with closures
19.8 Another technique for builders
19.9 Securing your DSLs
19.9.1 Introducing SecureASTCustomizer
19.9.2 The ArithmeticShell
19.9.3 Stopping the execution of your programs
19.9.4 Preventing cheating with meta-programming
19.10 Testing and error reporting

19

Domain-Specific Languages

Throughout this chapter, we'll cover:

- How Groovy allows you to write Domain-Specific Languages (nicknamed *DSLs*), i.e. languages tailored towards representing a particular domain of knowledge,
- How to concretely integrate DSLs in your applications,
- What the various techniques are to create readable and expressive languages,
- And how to test, secure and provide good error reporting.

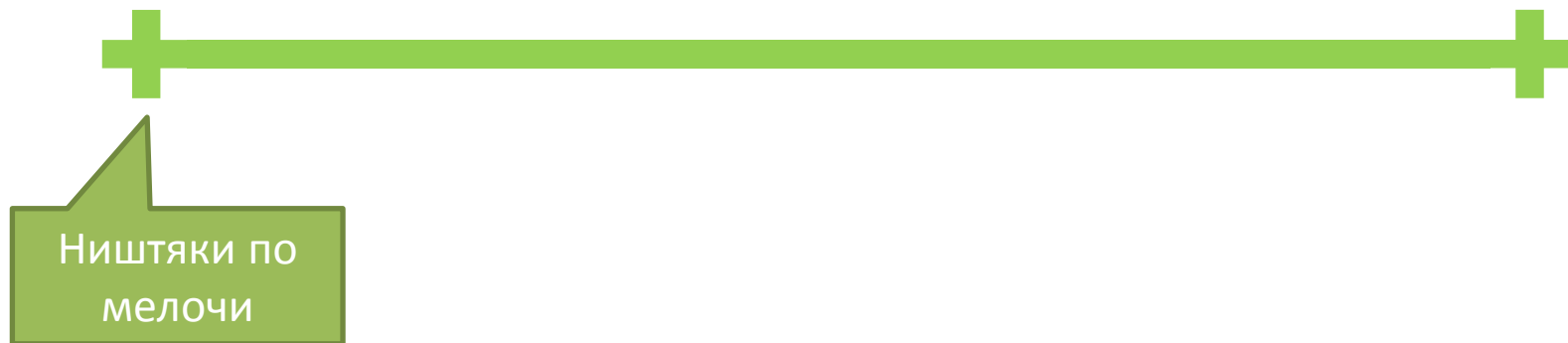
Various papers and studies will give you statistics about project successes and failures. On the other end of the spectrum, proponents of development methodologies will tell you about various techniques and approaches you should adopt to ensure that your projects have better chances to succeed. You can read a lot of interesting literature on those topics and I would argue that oftentimes, a common denominator in those sources about what leads a project on its path to success or failure is the quality of communication between the different parties involved, how the various stakeholders exchange information and cooperate together towards a common goal of producing quality software that delivers on their promises at solving a particular domain problem.

Languages are at the root of any kind of communication and involve two interlocutors. A *Subject Matter Expert* (often reduced to the SME acronym) can write specifications in his mother tongue, say, English, with tons of very domain specific words and concept names that will be read by software developers. A developer can also somehow speak with a computer with different languages to tell it about the business rules of the application the SME is longing to play with. The former will use a natural language while the latter will use one or several

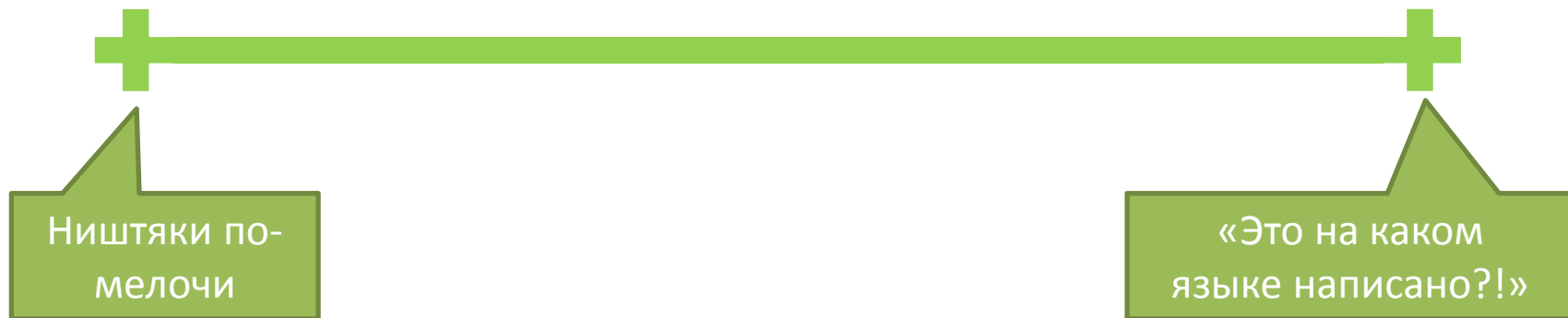
На самом деле, это диапазон



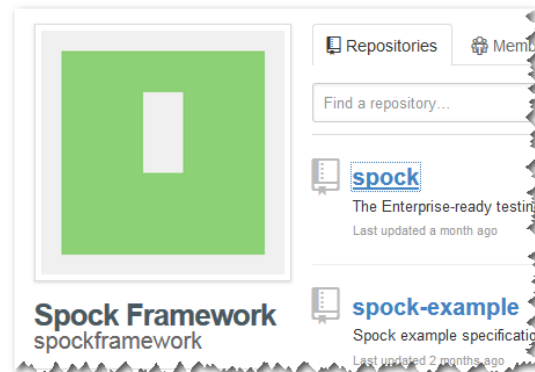
На самом деле, это диапазон



На самом деле, это диапазон



На самом деле, это диапазон

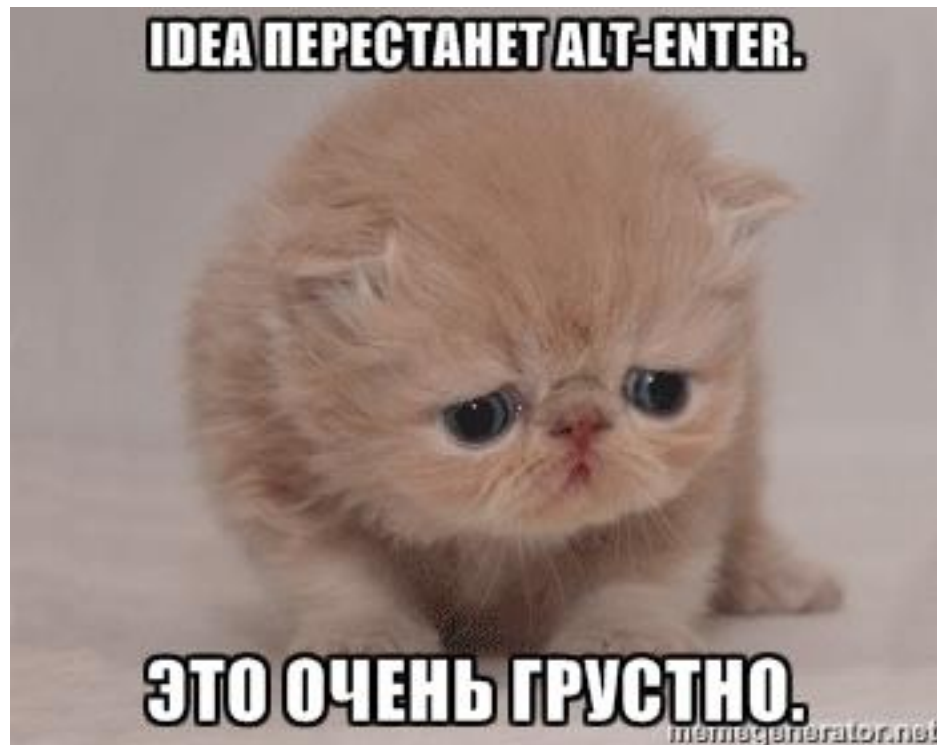


- [-] Chapter 19: Domain-Specific Languages
 - [-] 19.1 Groovy's flexible nature
 - [-] 19.1.1 Back on parentheses omission
 - [-] 19.2 Variable, constant and method injection
 - [-] 19.2.1 Injecting constants through the binding
 - [-] 19.2.2 Adding imports and static imports automatically
 - [-] 19.2.3 Injecting methods into a script
 - [-] 19.2.4 Adding closures to the binding
 - [-] 19.3 Adding properties to numbers
 - [-] 19.4 Leveraging named-arguments
 - [-] 19.5 Command chains
 - [-] 19.6 Your own control structures
 - [-] 19.7 Context switching with closures
 - [-] 19.8 Another technique for builders
 - [-] 19.9 Securing your DSLs
 - [-] 19.9.1 Introducing SecureASTCustomizer
 - [-] 19.9.2 The ArithmeticShell
 - [-] 19.9.3 Stopping the execution of your programs
 - [-] 19.9.4 Preventing cheating with meta-programming

На самом деле, это диапазон



Чё, правда, новый язык?



Петя Громов спешит на помощь



«Всё кул, посоны!»

Scripting IDE for DSL awareness

 11 Added by **Peter Gromov**, last edited by **Peter Gromov** on May 01, 2012 ([view change](#))

GroovyDSL scripting framework

GroovyDSL is a framework with a domain-specific language designed to define the behaviour of end-user DSLs as script files which are executed by the IDE on the fly, bringing **new reference resolution and code completion** logic into the scope of a project. GroovyDSL relies on the Program Structure Interface (PSI), a set of internal classes and interfaces of IntelliJ IDEA, which allow all programming languages to be described in a uniform way. Due to Groovy's meta-programming capabilities, the developer of DSL descriptions in GroovyDSL need not be aware of how PSI works. All interoperation with PSI is covered by calls to methods and properties of GroovyDSL scripts.

Ну, не совсем...

The screenshot displays the YouTrack web application interface. At the top, the YouTrack logo is on the left, and navigation links for 'Create Issue', 'Issues', 'Agile Board', and 'Reports' are in the center. The user 'Baruch Sadogursky' is logged in on the right. Below the navigation bar, a search bar contains 'Everything' and a filter 'by: me'. A toolbar with icons for navigation and actions is visible. The main content area shows a bug report for 'IDEA-113547' titled 'Support DSLs with ImportCustomizer in GDSDL'. The description states: 'If ImportCustomizer is added to CompilerConfiguration of the DSL script, all the classes imported by the customerizer should be resolvable without additional imports.' There is an 'Attach file...' button and a note 'Issue is visible to: All Users'. On the right, a sidebar lists issue details: Project (IntelliJ IDEA), Priority (Normal), Type (Feature), Fix versions (No Fix versions), State (Submitted), and Assignee (Maxim Medvedev). Several elements, including the title, 'No Fix versions', and the 'Attach file...' button, are highlighted with yellow boxes.

IDEA-113547 Support DSLs with ImportCustomizer in GDSDL

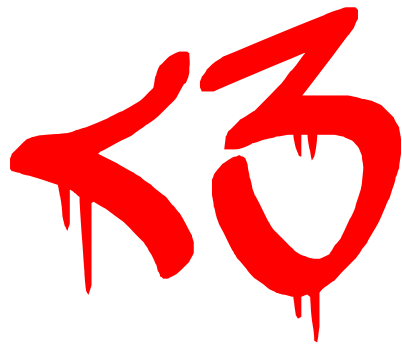
If ImportCustomizer is added to CompilerConfiguration of the DSL script, all the classes imported by the customerizer should be resolvable without additional imports.

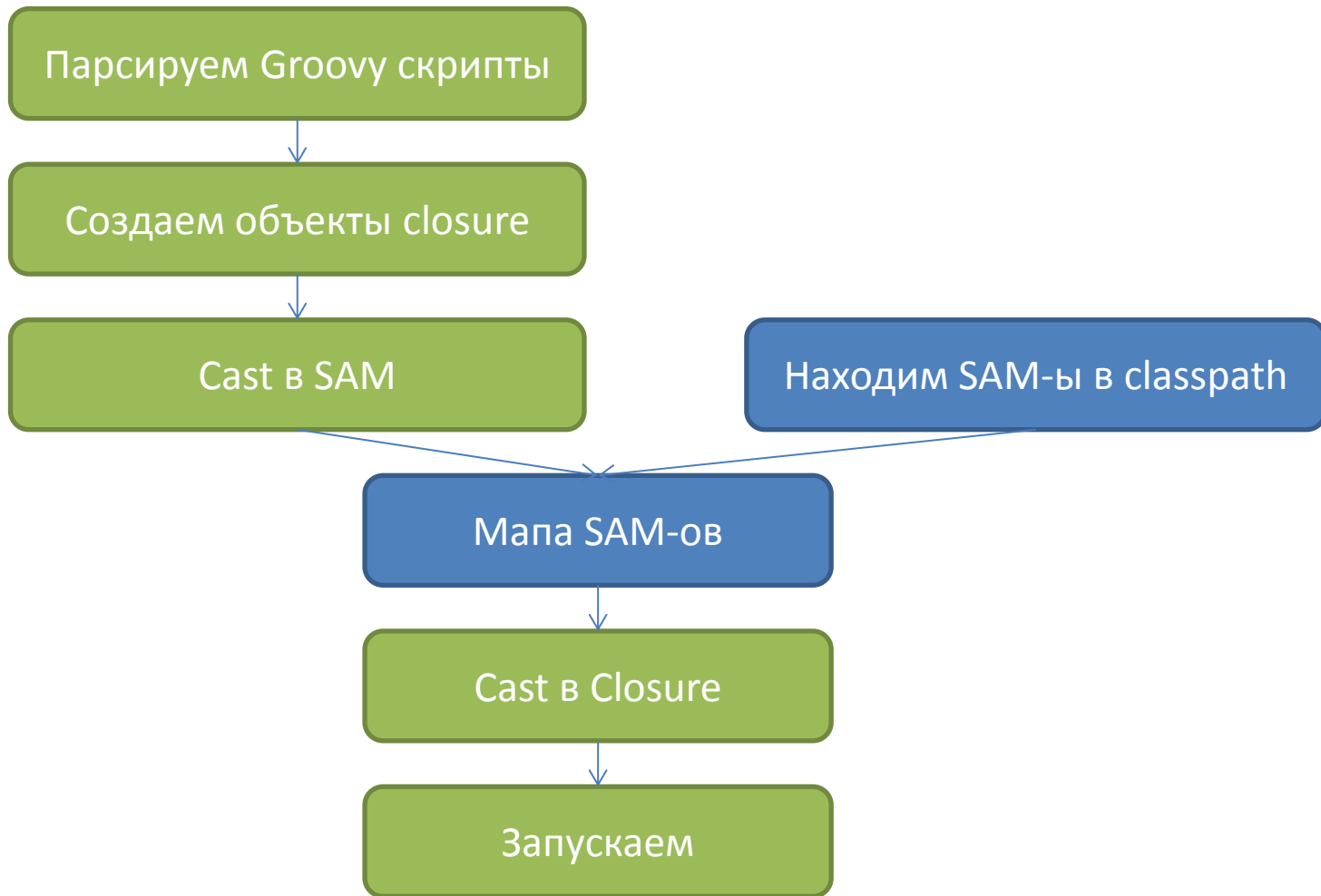
Attach file... or drag your files here

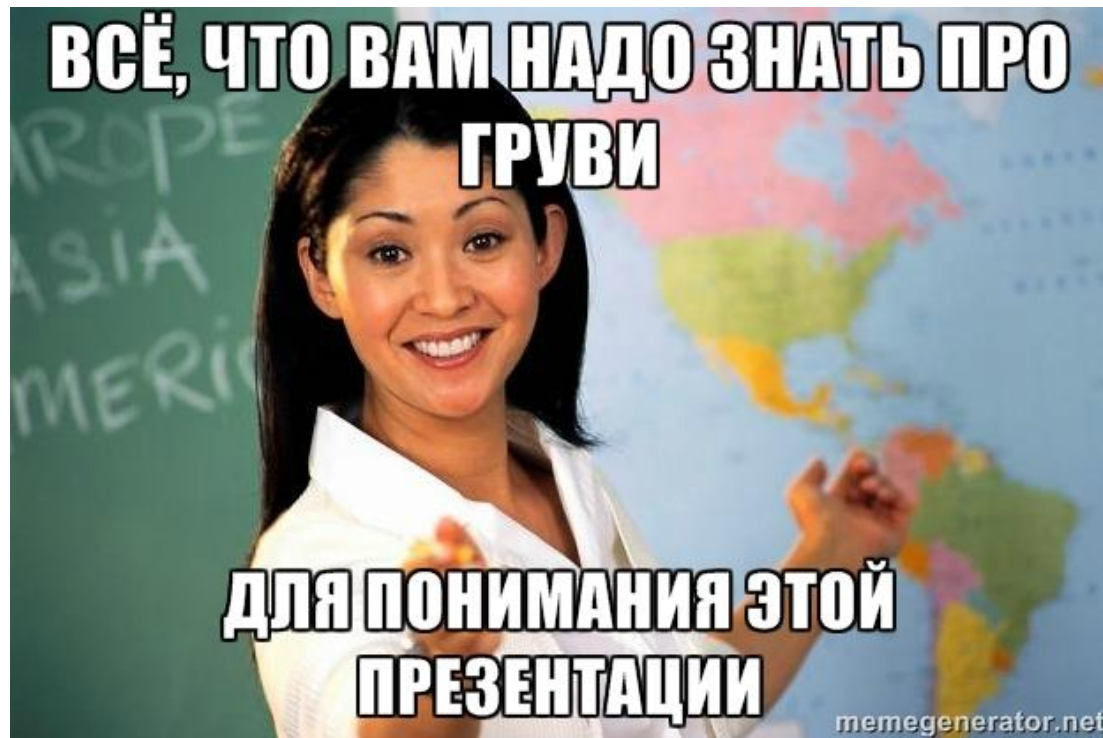
Issue is visible to: All Users

Showing issue #2 of 10 found

Project	IntelliJ IDEA
Priority	Normal
Type	Feature
Fix versions	No Fix versions
State	Submitted
Assignee	Maxim Medvedev







Closures & methodMissing

Эта ваша Java 7

//User Plugin

```
class MyAction implements AfterDownloadErrorAction {
```

```
    @Override
```

```
    void doAction() {
```

```
        System.out.println("Download Error!");
```

```
    }
```

```
}
```

Эта ваша Java 7

//API

```
interface AfterDownloadErrorAction extends Action {  
    void doAction();  
}
```


Эта ваша Java 7

//Plugin FW

```
List<Class<? extends Action>> afterDownloadErrorActions =  
ClasspathScanner.findClasses(AfterDownloadErrorAction.class)
```

```
actions.put(AfterDownloadErrorAction.class, afterDownloadErrorActions)
```

Эта ваша Java 7

//DownloadInterceptor

```
List<AfterDownloadErrorAction> actions =  
pluginAddon.afterDownloadErrorActions();  
for (AfterDownloadErrorAction action : actions) {  
    action.doAction();  
}
```

Отсыплем вам Грувэй

```
//User Plugin  
afterDownloadError {  
    println 'Download Error!'  
}
```

Магия...



Отсыплем вам Грувэй

//Plugin FW

```
void afterDownloadError(Closure closure){  
    actions[AfterDownloadErrorAction] << closure as Action  
}
```

Отсыплем вам Грувэй

```
//User Plugin  
Closure closure = {  
    println 'Download Error!'  
}  
pluginFramework.afterDownloadError(closure)
```


Отсыплем вам Грувэй

```
//User Plugin  
afterDownloadError {  
    println 'Download Error!'  
}
```



А теперь добавим динамизму

```
//User Plugin  
myCustomExecution {  
    println 'My Custom Execution :-P'  
}
```

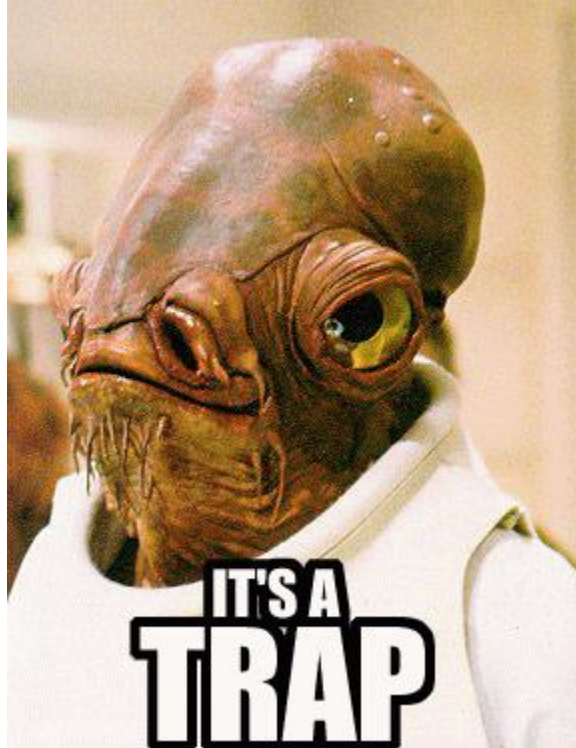


А теперь добавим динамизму

//Plugin FW

```
void ???(Closure closure){  
    actions[CustomExecutionAction] << closure as Action  
}
```

methodMissing FTW



А теперь добавим динамизму

//Plugin FW

```
def methodMissing(String name, args) {  
    actions[CustomExecutionAction] << args as Action  
}
```

Ну, пора!



Публичный API

All Classes

Packages

org.artifactory.addon.plugin
org.artifactory.build
org.artifactory.build.promotion
org.artifactory.build.staging

All Classes

Artifact
ArtifactList
BuildRun
Builds
BuildStagingStrategy
CancelException
ChecksumInfo
ChecksumsInfo
ChecksumType
Dependency
DependencyList
DetailedBuildRun
FileInfo
FileLayoutInfo
FolderInfo
HttpRepositoryConfiguration
Info
InfoEnabledPlugin
ItemInfo
LocalRepositoryConfiguration
MetadataInfo
Module
ModuleVersion
MutablePropertiesInfo
PathUtils
Plugin
PluginInfo
PromotionConfig
Properties

Overview

Package

Class

Use

Tree

Deprecated

Index

Help

Prev

Next

Frames

No Frames

Artifactory Public API 3.0.3

Packages

Package	Description
org.artifactory.addon.plugin	
org.artifactory.build	
org.artifactory.build.promotion	
org.artifactory.build.staging	
org.artifactory.checksum	
org.artifactory.common	
org.artifactory.exception	
org.artifactory.fs	
org.artifactory.md	
org.artifactory.repo	
org.artifactory.request	
org.artifactory.resource	
org.artifactory.search	
org.artifactory.security	
org.artifactory.util	

Overview

Package

Class

Use

Tree

Deprecated

Index

Help

Prev

Next

Frames

No Frames

Copyright © 2013. All Rights Reserved.

ПРОЕКТИРУЕМ РАРІ

- РАРІ ЭТО КОНТРАКТ

НЕЛЬЗЯ ПРОСТО ТАК ВЗЯТЬ,

**И СЛОМАТЬ ОБРАТНУЮ
СОВМЕСТИМОСТЬ В РАРІ**

ПРОЕКТИРУЕМ РАРІ

- РАРІ ЭТО КОНТРАКТ

ПРОЕКТИРУЕМ РАРІ

- РАРІ ЭТО КОНТРАКТ
- НЕ ЛОМАЕМ! ОБРАТНАЯ
СОВМЕСТИМОСТЬ - НАШЕ ВСЕ!

ПРОЕКТИРУЕМ РАРІ

- РАРІ ЭТО КОНТРАКТ
- НЕ ЛОМЛЕМ! ОБРАТНАЯ
СОВМЕСТИМОСТЬ - НАЩЕ ВСЕ!
- НАЧИНАЕМ ПО-МАЛЕНЬКУ

ПРОЕКТИРУЕМ РАРІ

- РАРІ ЭТО КОНТРАКТ
- НЕ ЛОМАЕМ! ОБРАТНАЯ
СОВМЕСТИМОСТЬ - НАЩЕ ВСЕ!
- НАЧИНАЕМ ПО-МАЛЕНЬКУ
- РУГАНЬ ПОЛЬЗОВАТЕЛЕЙ
ПОДСКАЖЕТ, ЧЕГО НЕ ХВАТАЕТ

ПРОЕКТИРУЕМ API

- НЕ ЛОМАЕМ! ОБРАТНАЯ
СОВМЕСТИМОСТЬ - НАШЕ ВСЕ!
- НАЧИНАЕМ ПО-МАЛЕНЬКУ
- РУКАНЬ ПОЛЬЗОВАТЕЛЕЙ
ПОДСКАЖЕТ, ЧЕГО НЕ ХВАТАЕТ
- FLUENT API - ЭТО ХОРОШО

ПРОЕКТИРУЕМ API

СОВМЕСТИМОСТЬ - НАШЕ ВСЕ!

- НАЧИНАЕМ ПО-МАЛЕНЬКУ
- РУГАНЬ ПОЛЬЗОВАТЕЛЕЙ ПОДСКАЖЕТ, ЧЕГО НЕ ХВАТАЕТ
- FLUENT API - ЭТО ХОРОШО
- НЕ ЛЮБИШЬ ПИСАТЬ ДОКМ? ТОГДА API ДОЛЖЕН БЫТЬ STRONGLY TYPED

Ещё немножко...



Groovy DSLs: Beginner to Expert [CON5389]

 **Paul King** - Director

This session examines how dynamic languages in general and Groovy in particular take us toward the goal of a dynamic language for the JVM. It offers many features that enable concise expressions, command chain expressions, operator overloading, named arguments, and other concise

Additional Information

Conference:	JavaOne
JavaOne Primary Track:	JAVAONE: Emerging Languages on the Java Virtual Machine
JavaOne Experience Level:	Intermediate
JavaOne Session Category:	Best Practices
Attendee Role:	Developer
Type:	Conference Session



AGENDA ▾

SPEAKERS ▾

TRAVEL

CONTACT

MEMBERS ▾

REGISTER

Creating Groovy DSLs that Developers can Actually Use

In this presentation, Guillaume, Paul, and Andrew will show you how to leverage Groovy to build a Domain-Specific Language (DSL) used to control a rover on Mars! Various metaprogramming techniques and integration mechanisms will be demonstrated. But the language itself is only the first part of the story. Developers cannot be expected to properly use a DSL without first-class IDE support and documentation.

The presentation will start by building the DSL from scratch, using the power of Groovy to create a concise and readable mini-language, and showing how to secure its integration. The second part of the presentation will demonstrate how to integrate the DSL into Groovy-Eclipse with custom content assist, navigation, searching, and inline documentation.

About the speaker



Guillaume LaForge

As Head of Groovy Development for SpringSource, Guillaume LaForge is the official Groovy Project Manager. He initiated the creation of the Grails web framework, and created the Gaelyk lightweight toolkit for Google App Engine. He is also a frequent conference speaker presenting Groovy and Grails at JavaOne, SpringOne, QCon, the Sun TechDays, and JavaPolis. Guillaume also co-authored Groovy in Action. Before founding G2One, which was acquired by SpringSource in late 2008, and taking the role of VP Technology, Guillaume worked for OCTO Technology, a consultancy focusing on architecture and agile methodologies. While at OCTO, Guillaume developed new offerings around Groovy and Grails for its customers.

[More About Guillaume »](#)

Paul King

Paul King leads ASERT, an organization based in Brisbane, Australia which provides software development, training and mentoring services to customers wanting to embrace new technologies, harness best practices and innovate. He has been contributing to open source projects for nearly 20 years and is an active committer on numerous projects including Groovy. Paul speaks at international conferences, publishes in software magazines and journals, and is a co-author of Manning's best-seller: Groovy in Action.

[More About Paul »](#)

Groovy



Paul Kir

This sessi
domain. C
metapro

Ac



**What about
security and
safety?**

Security and Safety

JVM Security Managers

SecureASTCustomizer

Sandboxing

Controlling script execution

**Да ну его
нафиг...**



Это-ж их сервер!

Ах, зависимости...

Интернет-зависимость

Материал из Википедии — свободной энциклопедии

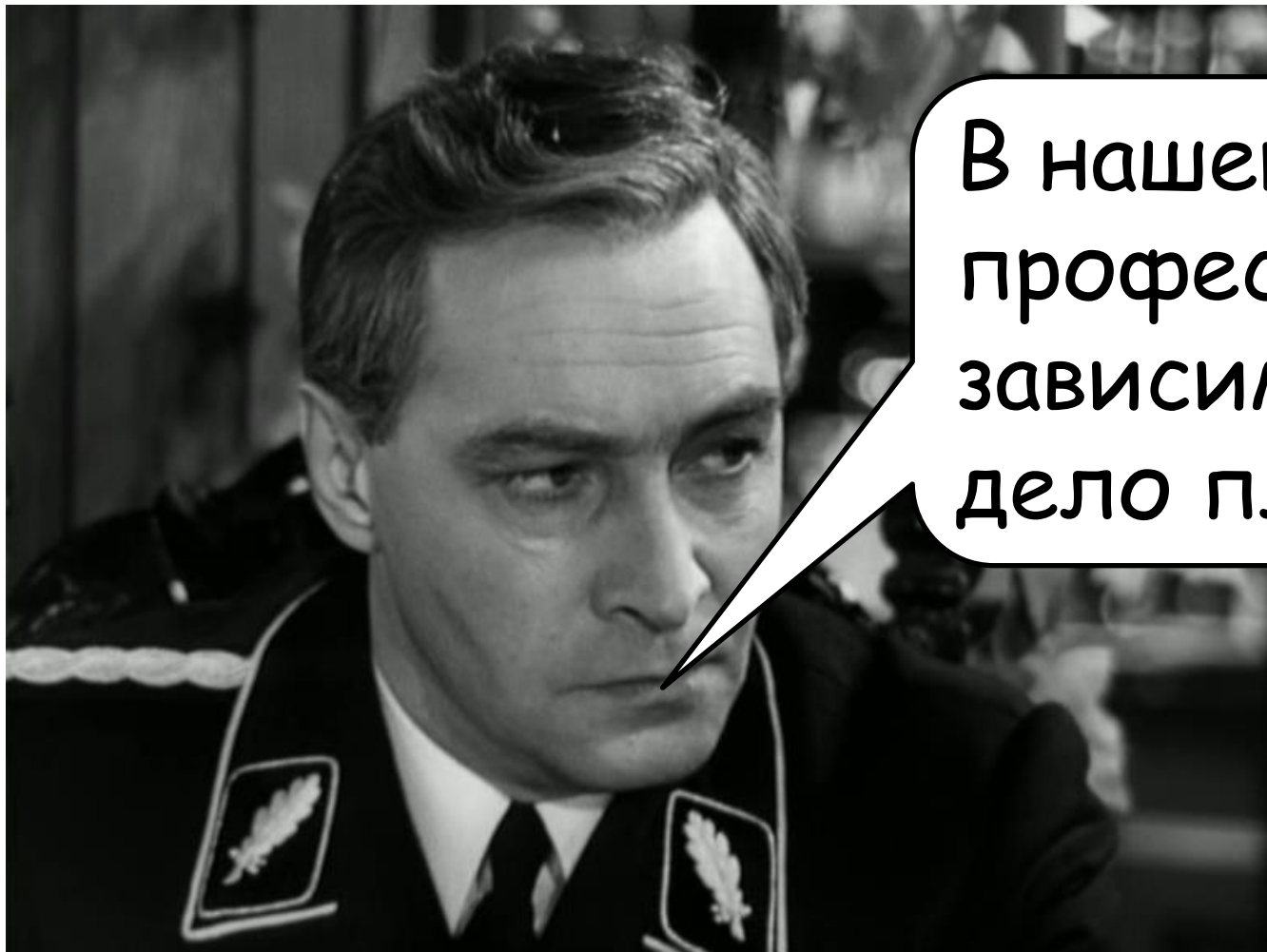
Интернёт-зави́симость (или **Интернет-адди́кция**) — навязчивое желание подключиться к [Интернету](#) и болезненная неспособность вовремя отключиться от Интернета. Интернет-зависимость является широко обсуждаемым вопросом, но

Ах, зависимости...





История с
радишкой Кэт
подтверждает:

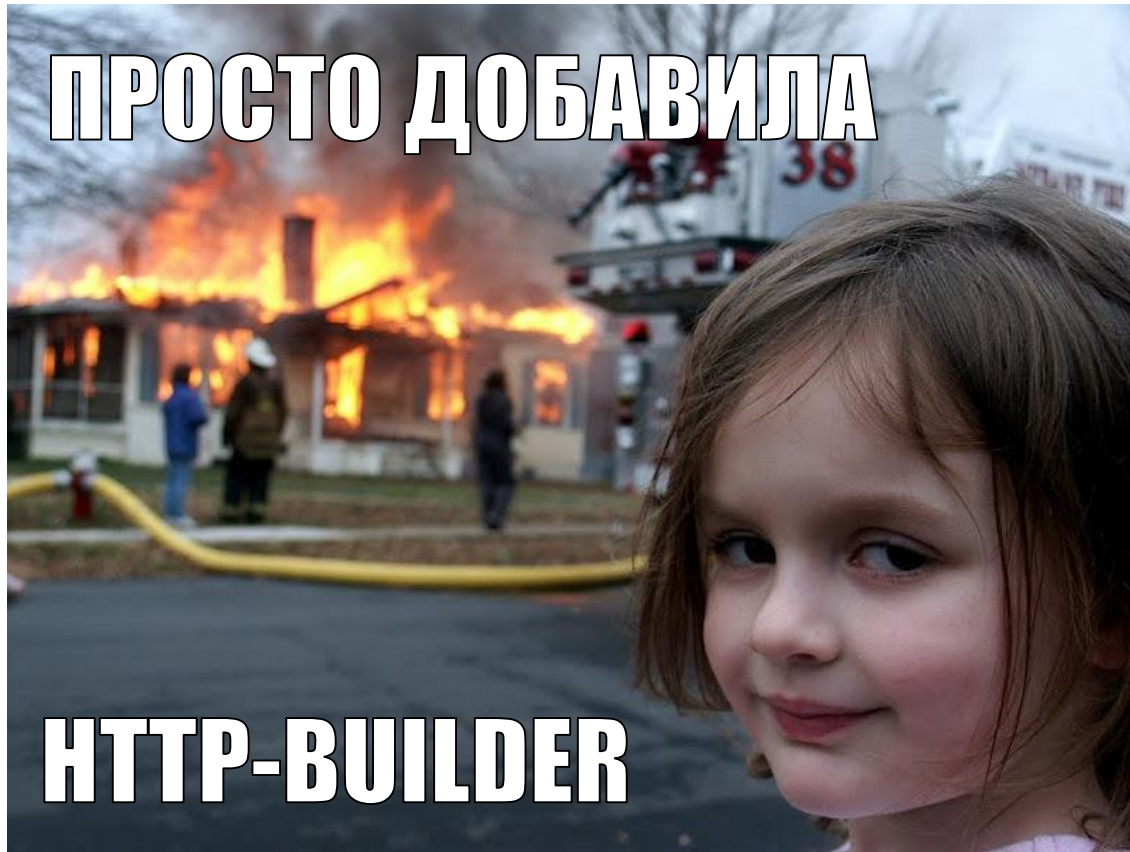


В нашей
профессии,
зависимости -
дело плохое!

Ах, зависимости...

ПРОСТО ДОБАВИЛА

HTTP-BUILDER



КОНФЛИКТ

КОНФЛИКТ

- НУЖЕН ДОСТУП К ДЖАРАЛА
ARTIFACTORY (PAP)

КОНФЛИКТ

- НУЖЕН ДОСТУП К ДЖАРАЛА
ARTIFACTORY (PARD)
- НУЖНА ИЗОЛЯЦИЯ ОТ ДЖАРОВ
ARTIFACTORY (3RD PARTY)





Давайте выбирать

Ох, какое всё
вкусное!



OSGi

OSGi



JBoss modules

JBoss modules



David M. Lloyd

@dmlloyd0

@jbaruch the manual at docs.jboss.org/author/display... is fairly complete to get you started, also check out the [#wildfly](#) distro.

JBoss modules



David M. Lloyd

@dmlloyd0

@jbaruch the manual at docs.jboss.org/author/display... is fairly complete to get you started, also check out the [#wildfly](#) distro.



Introduction



Added by [David Lloyd](#), last edited by [David Lloyd](#) on Jul 13, 2011 ([view change](#))

JBoss Modules is a standalone implementation of a *modular* (non-hierarchical) class loading and execution environment for Java. In other words, rather than a single class loader which loads all JARs into a flat class path, each library becomes a *module* which only links against the exact modules it depends on, and nothing more. It implements a thread-safe, fast, and highly concurrent delegating class loader model, coupled to an extensible module resolution system, which combine to form a unique, simple and powerful system for application execution and distribution.

JBoss Modules is designed to work with any existing library or application without changes, and its simple naming and resolution strategy is what makes that possible. Unlike OSGi, JBoss Modules does not implement a container; rather, it is a thin bootstrap wrapper for executing an application in a modular environment. The moment your application takes control, the modular environment is ready to load and link modules as needed. Furthermore, modules are never loaded (not even for resolution purposes) until required by a dependency, meaning that the performance of a modular application depends only on the number of modules actually used (and when they are used), rather than the total number of modules in the system. And, they may be unloaded by the user at any time.

Jigsaw

Jigsaw

Project Jigsaw. Take 2

Модульная система, встроенная в платформу JDK. Расширение языка Java. Изменение логики загрузки классов. Новая область видимости. Разбиение API платформы Java на десятки модулей. Обязательные и опциональные зависимости...

Так изначально планировалась функциональность [Project Jigsaw](#). Проект, который изначально планировался для JDK 8, был перенесен на JDK 9 и прямо сейчас подвергается детальному анализу и редизайну. Какими будут требования к проекту? Понадобится ли специальный язык для определения зависимостей? Будет ли новая реализация самостоятельной распределённой системой? В чем она будет...

Ну, пусть будет Java 9

А когда? Давайте считать!

There's not a moment to lose!



Mark Reinhold

Chief Architect, Java Platform Group at Oracle

San Francisco Bay Area | Computer Software

It's time for ... Plan B

2010/09/20 16:42:59 -07:00

In my [previous entry](#) I described two plausible plans for moving forward with JDK 7:

Plan A: JDK 7 (as currently defined) Mid 2012

Plan B: JDK 7 (minus Lambda, Jigsaw, and part of Coin) Mid 2012

JDK 8 (Lambda, Jigsaw, the rest of Coin, ++)

Late 2012

Thanks to everyone who responded to that entry, both directly and indirectly.

The voluminous feedback was strongly, though not uniformly, in favor of

Robert De Niro

Billy Crystal



**экстраполируй
это**

ДАНО:

РЕЛИЗ JAVA 7 07/2011

РЕЛИЗ JAVA 8 12/2012

НАЙТИ:

РЕЛИЗ JAVA 9 ?

JAVA 8: 12/2012

JAVA 8: 12/2012

-

JAVA 7: 07/2011

JAVA 8: 12/2012

-

JAVA 7: 07/2011

=

ЦИКЛ В 15 МЕСЯЦЕВ

JAVA 8: 12/2012

-

JAVA 7: 07/2011

=

ЦИКЛ В 15 МЕСЯЦЕВ

ВЫВОД: JAVA 9: 05/2013

JAVA 8: 12/2012

Это-ж полгода
назад!

Oh, wait...

Ещё-ж даже 8
не вышла!



ВЫВОД: JAVA 9: 05/2013

**Народная примета:
Когда Марк говорит про поезда,
это к задержке релиза**



Xe-xe...

There's not a moment to lose!

Mark Reinhold's Blog

Project Jigsaw: Late for the train

2012/07/17 08:58:00 -07:00

The aim of Project Jigsaw is to design and implement a standard module system for the Java SE Platform, and to apply that system to the Platform itself and to the JDK.

Jigsaw is currently slated for Java 8. The proposed development schedule for Java 8 expects work on major features to be finished by May 2013, in preparation for a final release around September. Steady progress is being made, but some significant technical challenges remain. There is, more importantly, not enough time left for the broad evaluation, review, and feedback which such a profound change to the Platform demands.

I therefore propose to defer Project Jigsaw to the next release, Java 9. In order to increase the predictability of all future Java SE releases, I further propose to aim explicitly for a regular two-year release cycle going forward.

JAVA 7: 07/2011

JAVA 7: 07/2011

+ 24 МЕСЯЦА =

JAVA 7: 07/2011

+ 24 МЕСЯЦА =

JAVA 8: 07/2013

Ща3...

There's not a moment to lose!

Mark Reinhold's Blog

Java 8: Secure the train

2013/04/18 08:48:37 -07:00

Security vulnerabilities related to Java running inside web browsers have lately received a lot of public attention. Here at Oracle we've mounted an intense effort to address those issues in a series of critical-patch update releases, the most recent of which we posted earlier this week. We've also upgraded our development processes to increase the level of scrutiny applied to new code, so that new code doesn't introduce new vulnerabilities.

Features vs. schedule I've previously argued that it's best to structure the Java development process as a continuous pipeline of innovation that's only exception to the general policy is therefore well justified.

If we adopt my proposal above then we should be able to complete the browser-related security work to which we've committed and then resume a regular two-year release cadence, with Java 8 due in early 2014 and Java 9 in early 2016.

~~JAVA 7: 07/2011~~

+ 24 МЕСЯЦА =

JAVA 8: 07/2013

~~JAVA 7: 07/2011~~

СКОЛЬКО МАРК СКАЖЕТ

+ 24 МЕСЯЦА =

JAVA 8: 01/2014

~~JAVA 7: 07/2011~~

СКОЛЬКО МАРК СКАЖЕТ

+ 24 МЕСЯЦА =

JAVA 8: 01/2014

+ 24 МЕСЯЦА =

~~JAVA 7: 07/2011~~

СКОЛЬКО МАРК СКАЖЕТ

+ 24 МЕСЯЦА =

JAVA 8: 01/2014

+ 24 МЕСЯЦА =

JAVA 9: 01/2016

~~JAVA 7: 07/2011~~

СКОЛЬКО МАРК СКАЖЕТ

+ 24 МЕСЯЦА =

JAVA 8: 01/2014

+ 24 МЕСЯЦА =

JAVA 9: 01/2016

Думаете это всё?

There's not a moment to lose!

Mark Reinhold's Blog

Java 8: Hold the train

2013/04/26 09:44:25 -07:00

Last week I proposed to delay the release of Java 8 in order to finish [Project Lambda](#), which has been delayed due to Oracle's renewed focus on the security of the Java Platform.

Thanks to everyone who responded to that proposal, in comments on my blog entry, on Twitter, and elsewhere. The feedback was generally in favor, though understandably tinged with disappointment. As of today we have a new plan of record: The target release date for Java 8 is now 2014/3/18.

Further schedule details are available on the [JDK 8 Project](#) pages.

~~JAVA 7: 07/2011~~

СКОЛЬКО МАРК СКАЖЕТ

+ 24 МЕСЯЦА =

JAVA 8: 03/2014

+ 24 МЕСЯЦА =

JAVA 9: 03/2016

(ЕСЛИ НЕ БУДЕТ БОЛЬШЕ
ПОСТОВ ПРО ПОЕЗД)

ФИГАСЕ, ЭТО-Ж ЧЕРЕЗ ПОЛГОДА

ПОСЛЕ ХОВЕРБОРДОВ И ВСЕГО ФАРША!

quickmeme.com

**Здесь была шутка про
Марка и Чопту Роисси,
но я вспомнил, что тут
полно народу из
Oracle.**

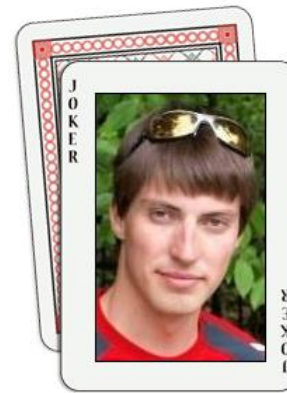
Зд

Ма

НО

по.

Ор

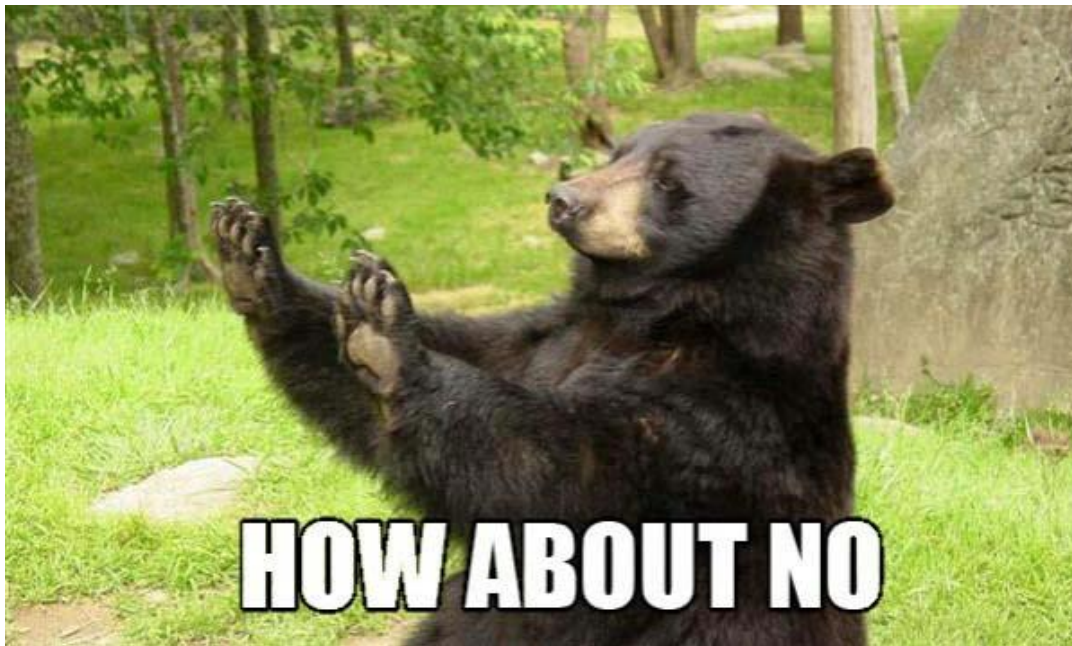


Ю

И,

Т

Jigsaw

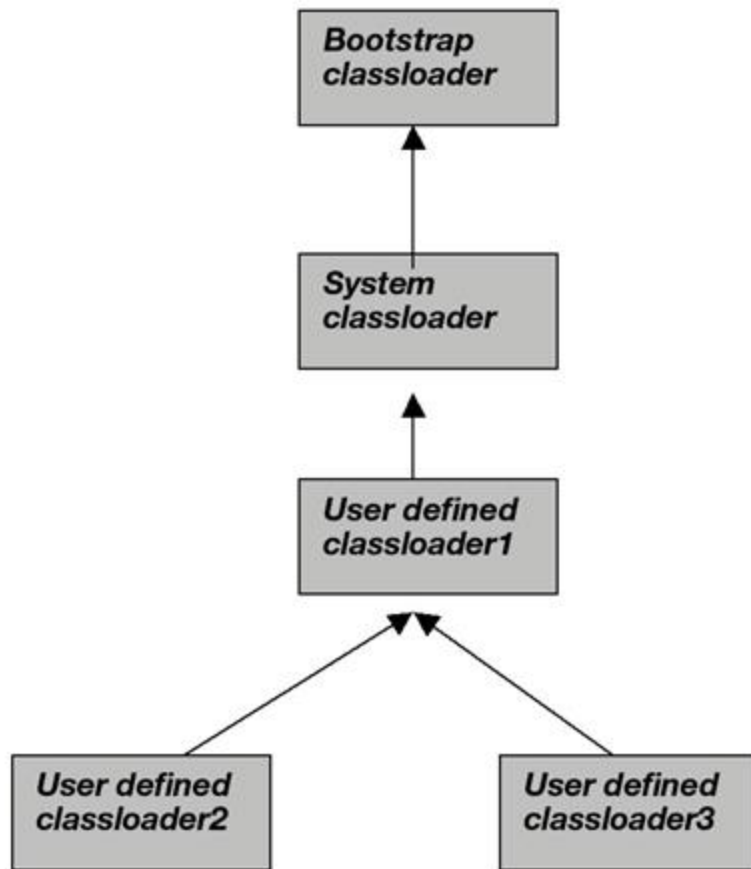


[статья](#)[обсуждение](#)[править](#)[история](#)

Фатальный недостаток

Фатальный недостаток — локальный мем айтишной тусовки. Вкратце означает, что кто-то от нежелания платить за [авторские права](#), [ЧСВ](#), зависти, скуки (или просто [выгнали](#)) **решает сделать что-то своё**, только [с блекджеком и шлюхами](#). Как правило, в результате очередное унылое творение тихо погибает [в гордом одиночестве](#).





DELOB TO...

ДЕЛОВ ТО...

1. ВЫТЯГИВАЕМ НАВЕРХ ВСЕ
ОБЩЕЕ

ДЕЛОВ ТО...

1. ВЫТЯГИВАЕМ НАВЕРХ ВСЕ
ОБЩЕЕ
2. ИЗОЛИРУЕМ ВНИЗУ ВСЕ
ОСТАЛЬНОЕ

ДЕЛОВ ТО...

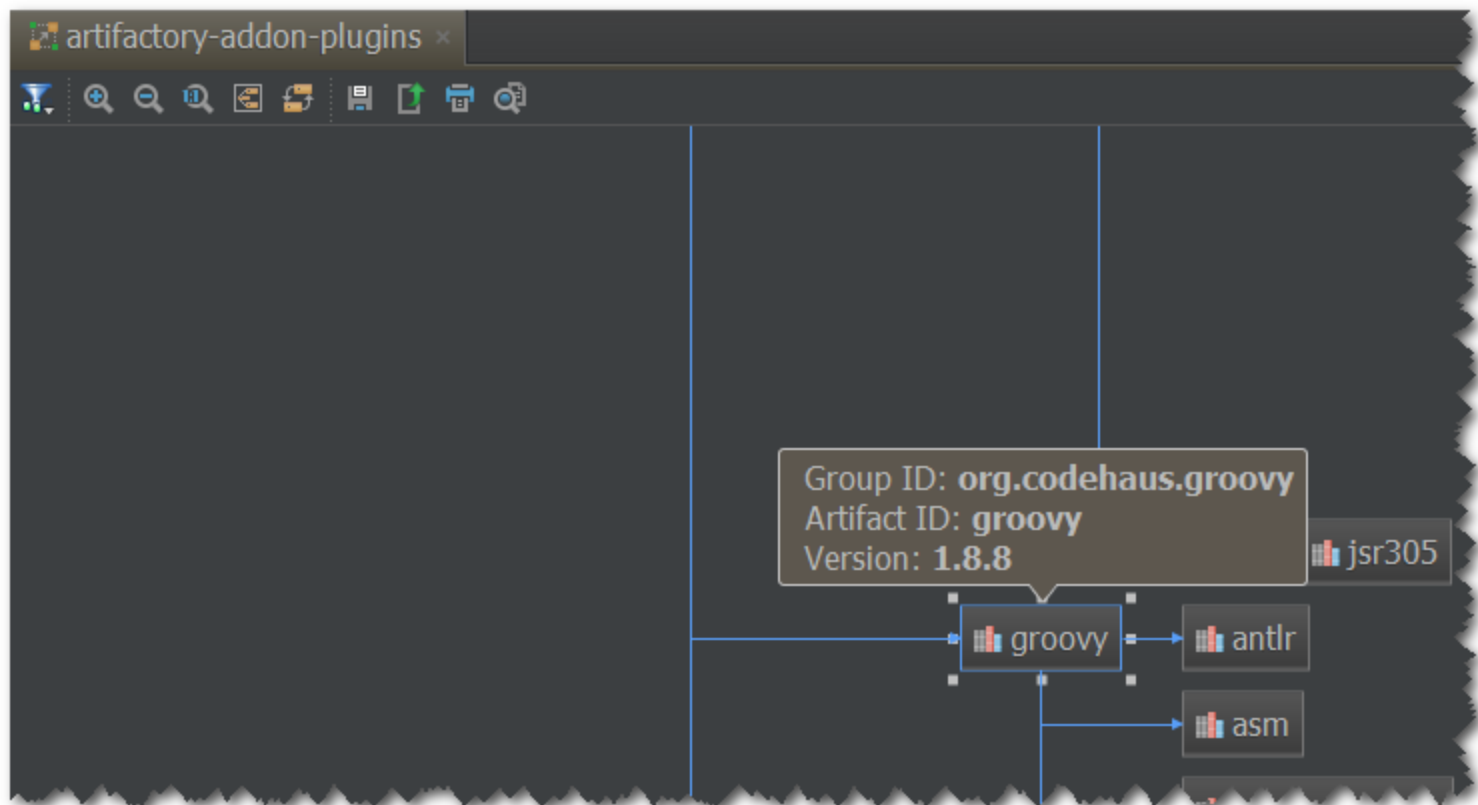
1. ВЫИ
ОБЦ
2. ИЗО
ОСТ

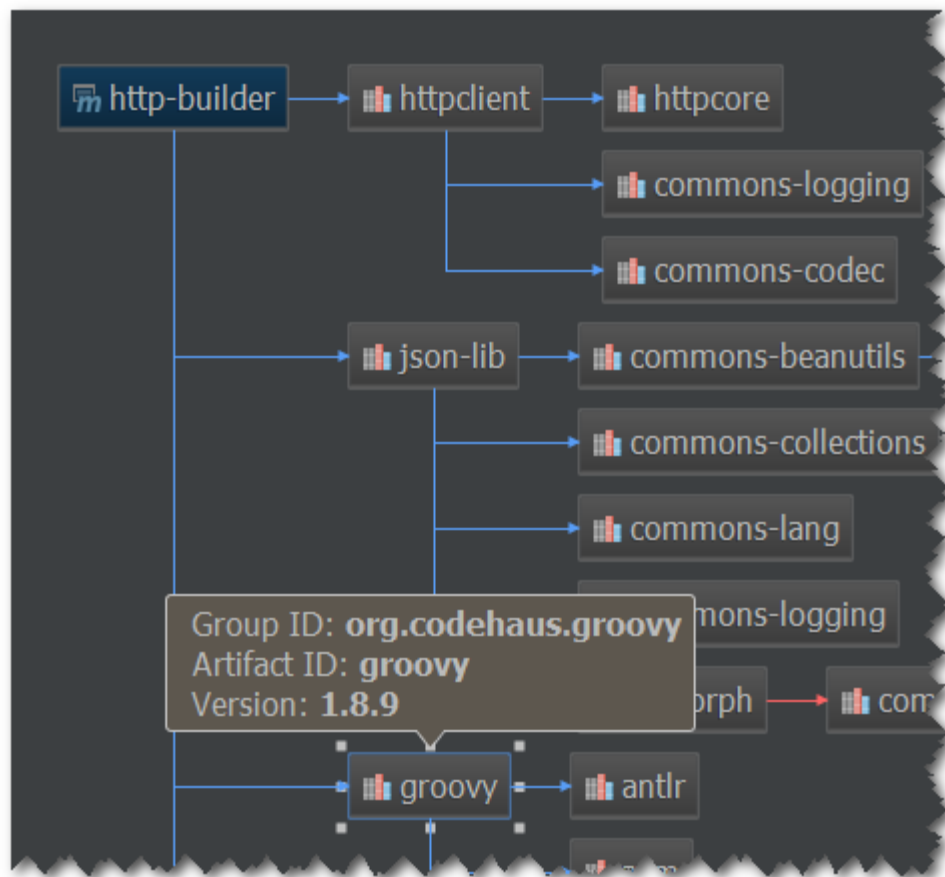
РАЗ ПЛЮНУТЬ.



Not so fast!









Вот что получилось:

The screenshot shows the GitHub interface for the repository **JFrogDev / artifactory-user-plugins**. The repository is public and has 73 commits, 1 branch, 0 releases, and 9 contributors. The current branch is **master**. A merge pull request #11 from itamarb/master is highlighted. Below this, a list of commits is shown, each with a folder icon, a commit message, and a timestamp.

Sample Artifactory User Plugins <http://wiki.jfrog.org/confluence/display/RTF/User+Plugins>

73 commits 1 branch 0 releases 9 contributors

branch: master artifactory-user-plugins

Merge pull request #11 from itamarb/master

itamarb authored a month ago latest commit 1fbd5d3b23

build	added staging example modifying module version map	6 months ago
checkums	size added, comments	9 months ago
cleanup	Update artifactCleanup.groovy	4 months ago
filestore	Added exposing filestore with symlinks	9 months ago
governance	more strict property check + license change	8 months ago
ivy2pom	ivy2pom and pom2ivy	a year ago
nexus-push	applied new Nexus 2.4 API	5 months ago
pgp-sign	support for folder cleanup when generated asc's are moved (as part of...	a year ago
pom2ivy	ivy2pom and pom2ivy	a year ago
security	added a plugin for protecting properties edit and delete	6 months ago
stats	Fixing example typo, should be repos not repo	a month ago
ArtifactoryUserPlugins.gdsl	private ctx removed from gdsl	6 months ago

Что вы, что вы, спасибо вам!

