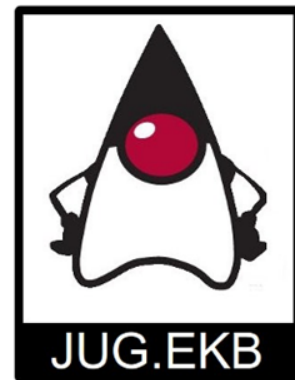
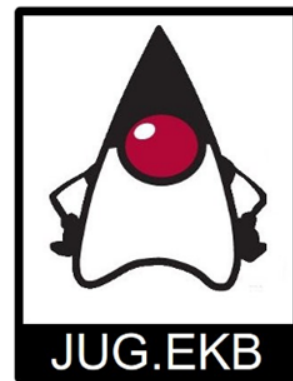


Как не наступить на грабли, внедряя скрипты в java приложение

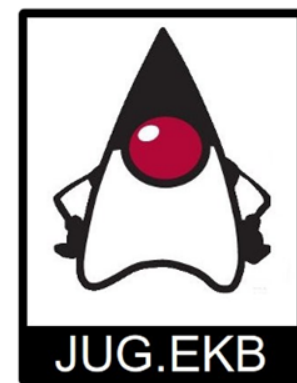
```
/**  
 * Сергей Целовальников  
 * Naumen  
 * sergeicelov@gmail.com  
 * @serceman  
 */
```



Это грабли



Когда ваше приложение требует динамичности





AR



Carbine



Short Carbine



Compact



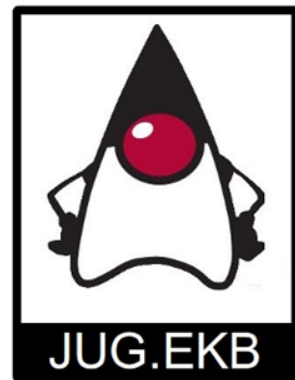
Sharpshooter



LMG

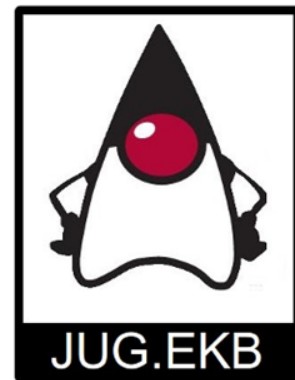


XM8R



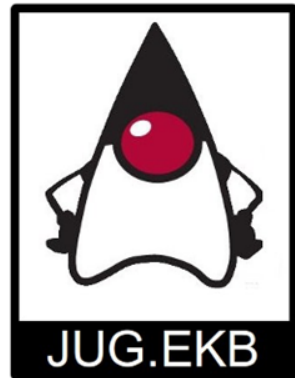
Исполняем код динамически

```
public static void main(String[] args) {  
    String script = "return 45;";  
    Object result = executeScript(script);  
    System.out.println(result);  
}
```



JSR 199: Java Compiler API

```
private static String template =  
    "public class Main {" +  
    "    public static Object doSome() {" +  
    "        %s" +  
    "    }" +  
    "}";
```



JSR 199: Java Compiler API

```
private static Object executeScript(String script) {
    JavaSourceFromString javaSource =
        new JavaSourceFromString("Main", format(template, script));
    JavaCompiler compiler = ToolProvider.getSystemJavaCompiler();

    StringWriter sw = new StringWriter();
    Boolean result = compiler.getTask(sw, null, null, null, null,
        asList(javaSource)).call();
    if (result) {
        try (URLClassLoader ucl = new URLClassLoader(new URL[] {
            new File("Main.class").getAbsolutePath()
                .getParentFile().toURI().toURL() })) {

            Method method = ucl.loadClass("Main").getMethod("doSome");
            return method.invoke(null, null);
        } catch (Exception e) {
            throw new RuntimeException("Error during execution", e);
        }
    } else {
        throw new RuntimeException("Error during compiling:\n" + sw.toString());
    }
}
```



Javassist

```
private static Object executeScript(String script) {  
    try {  
        ClassPool pool = ClassPool.getDefault();  
        CtClass cc = pool.makeClass("Main");  
        CtMethod method = CtMethod.make(  
            "public static Object doSome() {" +  
                script +  
                "}", cc);  
        cc.addMethod(method);  
        Method m = cc.toClass().getMethod("doSome");  
        return m.invoke(null, null);  
    } catch (Exception e) {  
        throw new RuntimeException(e);  
    }  
}
```



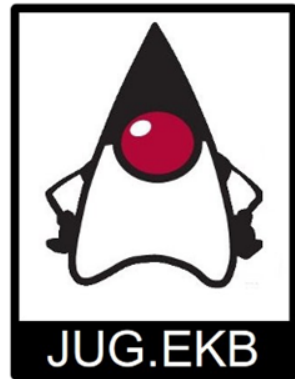
Groovy

```
private static Object executeScript(String script) {  
    GroovyShell shell = new GroovyShell();  
    return shell.evaluate(script);  
}
```



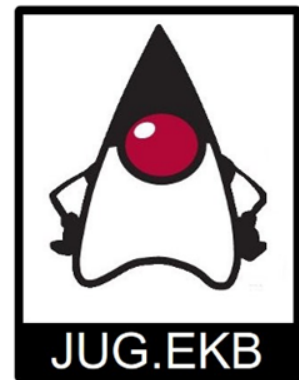
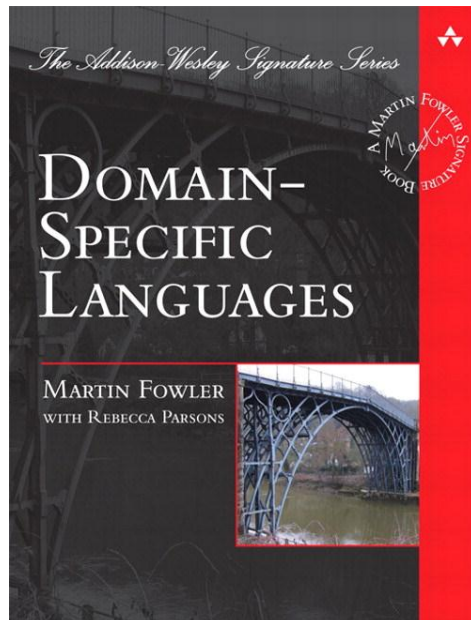
Why Groovy?

- Синтаксический сахар
- Динамичность
- Простота интеграции
- Java код – валидный groovy код
- Просто построить DSL



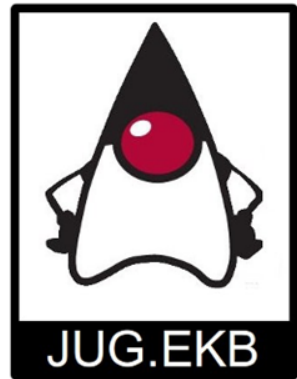
DSL

A domain-specific language (DSL) is a computer language specialized to a particular application domain



Groovy in 5 minutes

```
println 'Hello, World!'
```



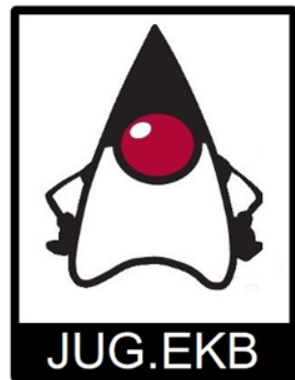
```
def a = 5

def hello() {
    'Hello'
}

assert hello() == 'Hello'

def clos = {
    ++a
}

assert clos() == 6
```



Spaceship	<=>	Useful in comparisons, returns -1 if left is smaller 0 if == to right or 1 if greater than the right
Regex find	=~	Find with a regular expression? See Regular Expressions
Regex match	==~	Get a match via a regex? See Regular Expressions
Java Field Override	.@	Can be used to override generated properties to provide access to a field
Spread	*.	Used to invoke an action on all items of an aggregate object
Spread Java Field	*.@	Amalgamation of the above two
Method Reference	.&	Get a reference to a method, can be useful for creating closures from methods
asType Operator	as	Used for groovy casting, coercing one type to another.
Membership Operator	in	Can be used as replacement for collection.contains()
Identity Operator	is	Identity check. Since == is overridden in Groovy with the meaning of equality we need some fallback to check for object identity.
Safe Navigation	?.	returns nulls instead of throwing NullPointerExceptions
Elvis Operator	?:	Shorter ternary operator

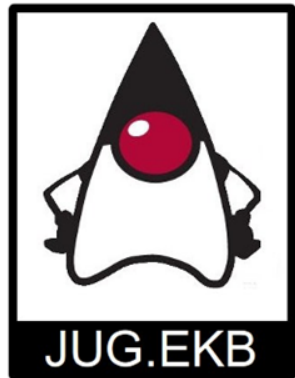
```
def name = 'JUG'

println "Hello, ${name}!"

def longstr = '''
    Hello,
    Multi
    Line
'''

// new methods
assert '123'.reverse() == '321'
assert 'abc'.capitalize() == 'Abc'

// etc...
```



```
def set = [] as Set
```

```
def list = [1, 2, 3, 4]
```

```
list << 4
```

```
assert list == [1, 2, 3, 4, 4]
```

```
assert list.findAll { it == 4 }.size() == 2
```

```
assert 2 in list
```

```
assert [1, 2, 3].join('-') == '1-2-3'
```

```
assert [1] + [2] == [1, 2]
```

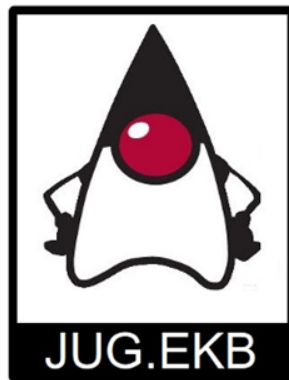
```
def someMap = [ 'a' : 1, 'b' : 2 ]
```

```
assert someMap['a'] == 1
```

```
assert someMap.b == 2
```

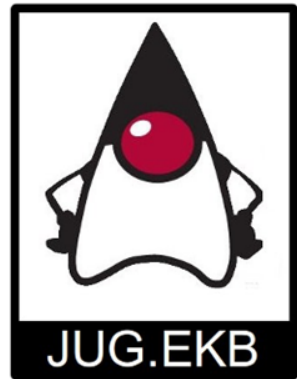


```
class Pair {  
    def l  
    def r  
  
    def setL(def n) {  
        r = l = n  
    }  
}  
  
Pair p = new Pair()  
  
p.setL(1) // ok  
p.l = 2   // setter called  
  
assert p.r == 2 // getter called
```



JSR 223: Scripting for the Java Platform

The specification will describe mechanisms allowing scripting language programs to access information developed in the Java Platform and allowing scripting language pages to be used in Java Server-side Applications.

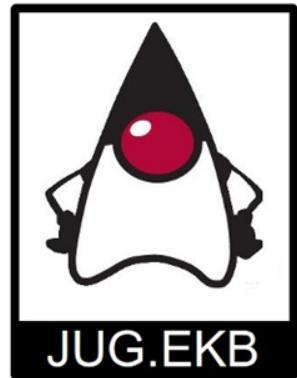


JSR 223: Scripting for the Java Platform

```
ScriptEngineManager factory = new ScriptEngineManager();  
ScriptEngine engine = factory.getEngineByName("js");
```

```
// basic example
```

```
System.out.println(engine.eval("var a = 2; a*a"));
```

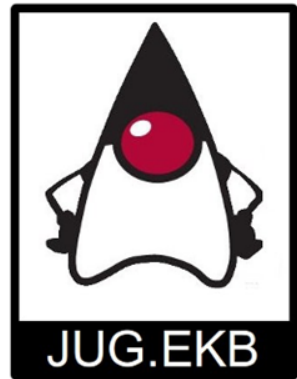


Пишем ScriptService

```
public class ScriptService {  
  
    ScriptEngine engine;  
    Bindings bindings;  
  
    public ScriptService() {  
        ScriptEngineManager factory = new ScriptEngineManager();  
        engine = factory.getEngineByName("groovy");  
  
        bindings = engine.createBindings();  
        bindings.put("user", new User());  
        bindings.put("timeService", new TimeService());  
    }  
  
    public Object execute(String script) throws ScriptException {  
        return engine.eval(script, bindings);  
    }  
}
```



```
if (user.name == 'Bender') {  
    timeService.goTo(3000)  
}
```

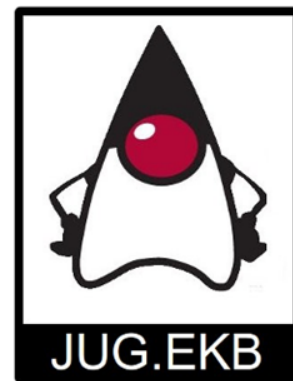


Мы несем ответственность за свой API



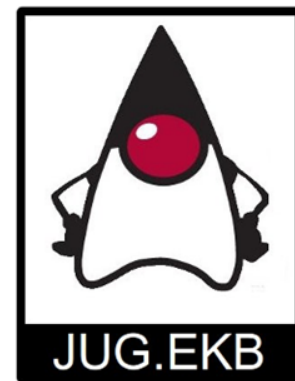
Правила проектирования скриптового API

1. Обратная совместимость



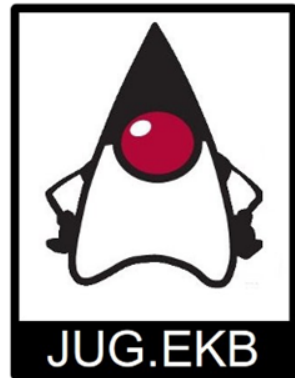
Правила проектирования скриптового API

1. Обратная совместимость
2. Документация



Правила проектирования скриптового API

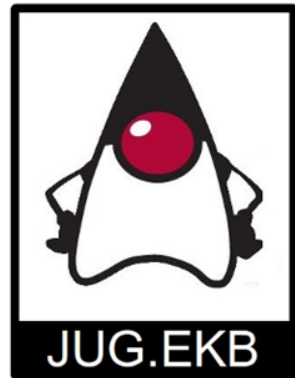
1. Обратная совместимость
2. Документация
3. Immutable объекты в контексте с информацией +
Stateless сервисы взаимодействия



Как написать хороший скрипт

1. Шапка скрипта

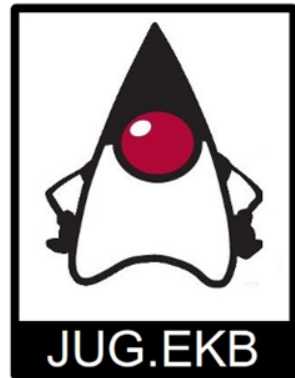
```
//Author: SerCe  
//Creation Date: 10.05.2014  
//API Version: 1.1  
/**  
 * Отправляет нас в 3000 год, но только если  
 * пользователь - Бендер!  
 */
```



Как написать хороший скрипт

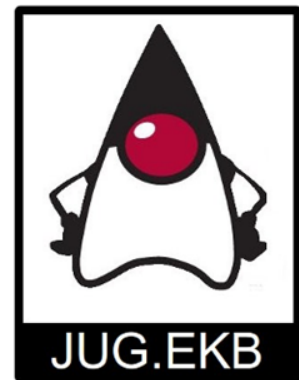
1. Шапка скрипта
2. Логирование

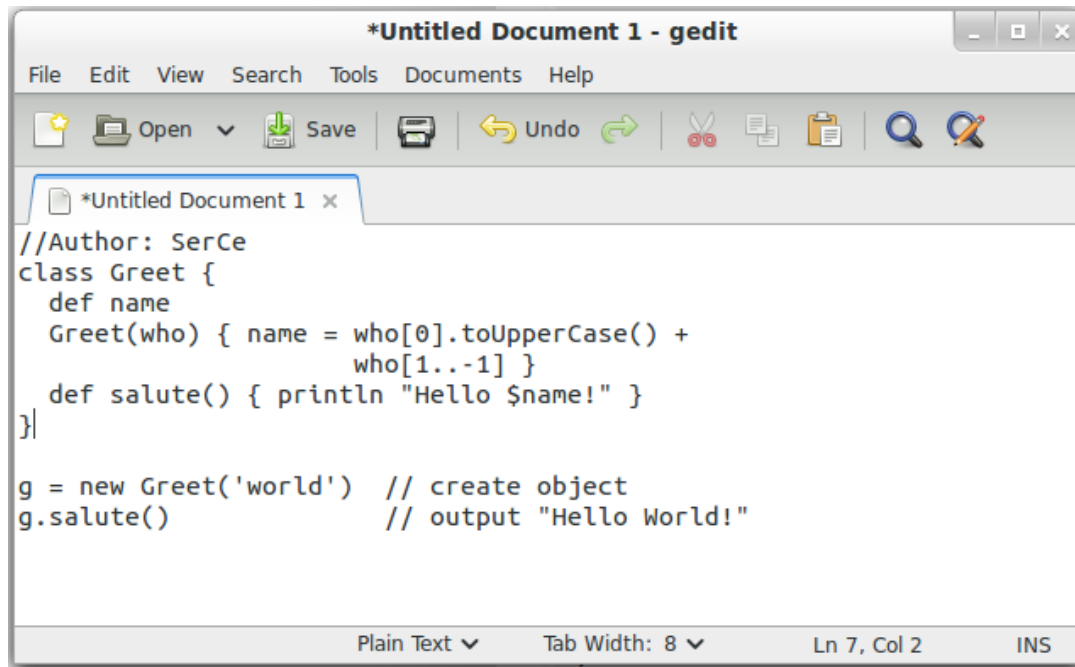
```
logger.info("User ${user.name} send to future")
```



Как написать хороший скрипт

1. Шапка скрипта
2. Логирование
3. Используйте преимущества Groovy





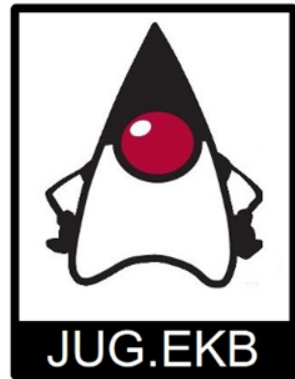
```
//Author: SerCe
class Greet {
  def name
  Greet(who) { name = who[0].toUpperCase() +
                who[1..-1] }
  def salute() { println "Hello $name!" }
}

g = new Greet('world') // create object
g.salute()             // output "Hello World!"
```



Groovy IDE

1. Groovy Eclipse Plugin
2. IntelliJ IDEA



```
//Author: SerCe
class Greet {
  def name
  Greet(who) { name = who[0].toUpperCase() +
                who[1..-1] }
  def salute() { println "Hello $name!" }
}

g = new Greet('world') // create object
g.salute()              // output "Hello World!"
```

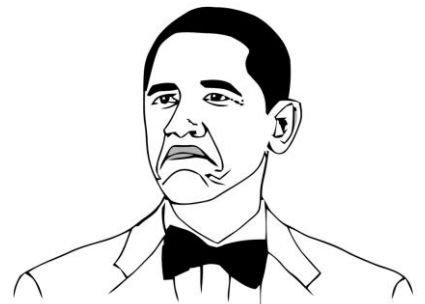
Save



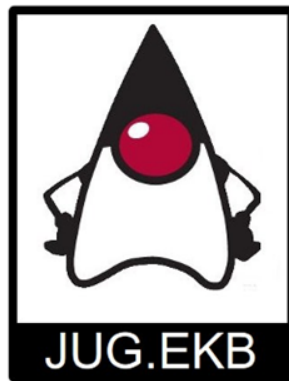
CodeMirror

```
1 //Author: SerCe
2 class Greet {
3   def name
4   Greet(who) { name = who[0].toUpperCase() + who[1..-1] }
5   def salute() { println "Hello $name!" }
6 }
7
8 g = new Greet('world') // create object
9 g.salute()             // output "Hello World!"
10
11
12 def gstr = "${g.salute()} !"
13
14
15 def printgstr = {
16   print gstr
17 }
18
```

Save




```
<script>
var editor = CodeMirror.fromTextArea(
    document.getElementById("code"), {
        lineNumbers : true,
        matchBrackets : true,
        styleActiveLine : true,
        lineWrapping : true,
        highlightSelectionMatches : true,
        autoCloseBrackets : true,
        mode : "text/x-groovy"
    }
);
</script>
```



```
//Author: SerCe
//Creation Date: 10.05.2014
//API Version: 1.1
/**
 * Отправляет нас в 3000 год, но только если
 * пользователь - Бендер!
 */

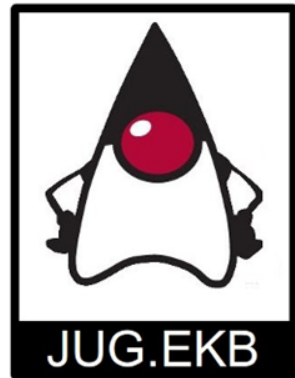
if (user.name == 'Bender') {
    logger.info("Age = ${user.age}")

    timeService.goTo(3000)
}
```



DSL D

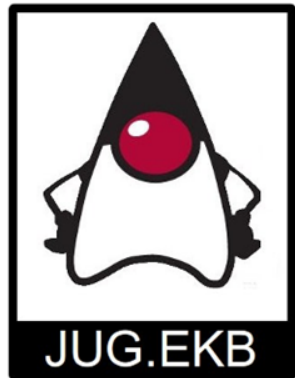
```
contribute(currentType(subType(Number))) {  
  property name: 'km', type: 'Distance.KM', doc: 'kilometer'  
  property name: 'm' , type: 'Distance.M' , doc: 'meter'  
  property name: 'cm', type: 'Distance.CM', doc: 'centimeter'  
}
```



DSL

```
contribute(currentType(subType(Number))) {  
  property name: 'km', type: 'Distance.KM', doc: 'kilometer'  
  property name: 'm' , type: 'Distance.M' , doc: 'meter'  
  property name: 'cm', type: 'Distance.CM', doc: 'centimeter'  
}
```

```
def result = 3.km + 2.m - 5.cm
```



```
//Author: SerCe
//Creation Date: 10.05.2014
//API Version: 1.1
/**
 * Отправляет нас в 3000 год, но только если
 * пользователь - Бендер!
 */

if (user.name == 'Bender') {
    logger.info("Age = ${user.age}")

    timeService.goTo(3000)
}
```



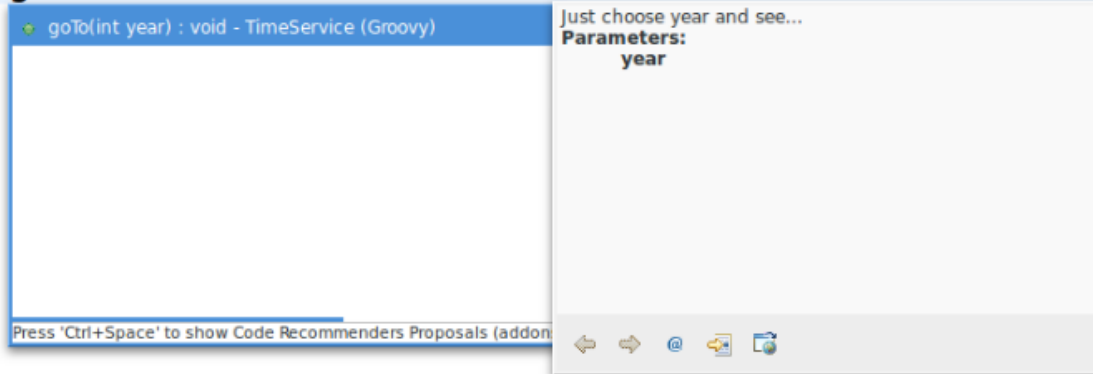
```
contribute(currentType(subType(groovy.lang.Script))) {  
    property name: 'user',  
              type: 'ru.serce.sample.User',  
              doc:  'A user! May be Bender???'  
  
    property name: 'timeService',  
              type: 'ru.serce.sample.TimeService',  
              doc:  'You can control the time!'  
  
    property name: 'logger',  
              type: 'ru.serce.sample.Logger',  
              doc:  'Just log it!'  
}
```

```
//Author: SerCe
//Creation Date: 10.05.2014
//API Version: 1.1
/**
 * Отправляет нас в 3000 год, но только если
 * пользователь - Бендер!
 */

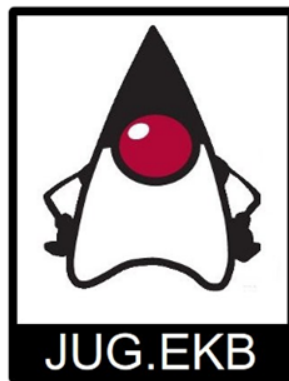
if (user.name == 'Bender') {
    logger.info("Age = ${user.age}")
}
```

timeService.go

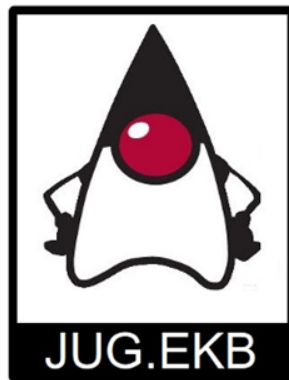
}



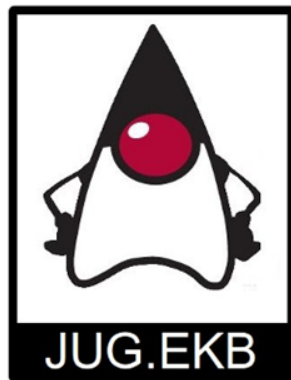
```
// к аннотированным элементам
contribute(annotatedBy("java.lang.Deprecated")) {
    method name:      'recreate',
              type:    'java.lang.Number',
              isStatic: true,
              doc:      'Simple method'
}
```




```
// ко всем в папке  
contribute(sourceFolderOfCurrentType("src/main/groovy")) {  
    delegatesTo List  
}
```



```
// к классу Bar с методом методу foo  
contribute(currentType(name("ru.serce.sample.Bar") & methods("foo"))) {  
  
}
```

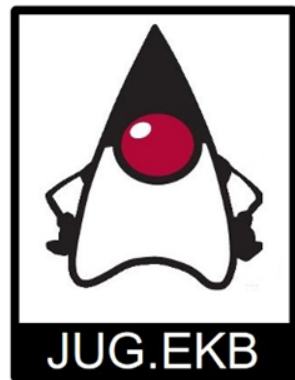


I will not write any more bad code
I will not write any more bad code
I will not write any more bad code
I will not write any more bad code
I will not write any more bad code
I will not write any more bad code
I will not write any more bad code
I will not write any more bad code
I will not write any more bad code
I will not write any more bad code
I will not write any more bad code





“CodeNarc analyzes Groovy code for defects, bad practices, inconsistencies, style issues and more”



Web - CodeNarc-Plugin/src/org/codenarc/eclipse/SelectionUtils.groovy - Eclipse

File Edit Source Refactor Navigate Search Project Run Window Help

SelectionUtils.groovy

```
1 package org.codenarc.eclipse
2
3 import java.util.Iterator
4 UnnecessaryGroovyImport
5
6 import org.eclipse.core.resources.IContainer
7 import org.eclipse.core.resources.IFile
8 import org.eclipse.core.resources.IResource
9 import org.eclipse.jface.viewers.IStructuredSelection
10
11 class SelectionUtils {
12
13     static List<IFile> getGroovyFiles(IStructuredSelection selection) {
14         def files = []
15         addFileResources(selection.toList(), files)
16         return files
17     }
18
19     static private void addFileResources(List resources, List files) {
20         Iterator iterator = resources.iterator()
21         while (iterator.hasNext()) {
```

Problems Properties Console

7 items

Description	Resource	Location	Type
▼ i Other (7 items)			
FieldName: The fieldname log in class org.codenarc.eclipse.RuleSetProvider does not match [A-Z][A-Z0-9_]*	RuleSetProvider.groovy	line 15	CodeNarc Violation (Priority 2)
StatelessClass: The class org.codenarc.eclipse.Activator is marked as stateless but contains the non-final field "plugin"	Activator.groovy	line 15	CodeNarc Violation (Priority 2)
UnnecessaryGetter: getDefault() can probably be rewritten as default	RuleSetProvider.groovy	line 15	CodeNarc Violation (Priority 2)
UnnecessaryGetter: getLog() can probably be rewritten as log	RuleSetProvider.groovy	line 15	CodeNarc Violation (Priority 2)
UnnecessaryGroovyImport	SelectionUtils.groovy	line 3	CodeNarc Violation (Priority 3)
UnnecessaryGroovyImport	SelectionUtils.groovy	line 4	CodeNarc Violation (Priority 3)
UnnecessaryReturnKeyword: The return keyword is not needed and can be removed	SelectionUtils.groovy	line 16	CodeNarc Violation (Priority 3)

Writable Smart Insert 4 : 1

СІ для скриптов



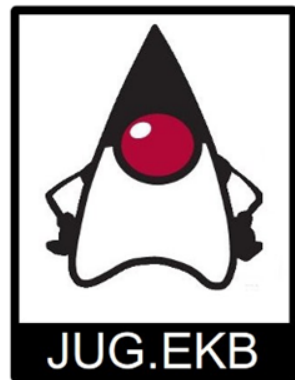
```
<plugin>
  <groupId>org.codehaus.mojo</groupId>
  <artifactId>codenarc-maven-plugin</artifactId>
  <version>${codenarc-maven-plugin.version}</version>
  <configuration>
    <codeNarcVersion>${codenarc-maven-plugin.codenarc.version}</codeNarcVersion>
    <rulesetfiles>file://${basedir}/src/main/resources/ruleset.xml</rulesetfiles>
    <!-- Сборка падает, если есть хотя бы 1 предупреждение -->
    <maxPriority2Violations>0</maxPriority2Violations>
    <!-- Добавляем api в контекст, т.к. используется в нашем правиле -->
    <additionalArtifacts>ru.jukekb:api:${project.build.api-version}</additionalArtifacts>
    <!-- Добавляем правило в classpath -->
    <addToClasspath>${basedir}/src/main/resources</addToClasspath>
  </configuration>
  <executions>
    <execution>
      <id>codenarc</id>
      <phase>compile</phase>
      <goals>
        <goal>codenarc</goal>
      </goals>
    </execution>
  </executions>
</plugin>
```

```
<ruleset xmlns="http://codenarc.org/ruleset/1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://codenarc.org/ruleset/1.0 http://codenarc.org/ruleset-schema.xsd"
  xsi:noNamespaceSchemaLocation="http://codenarc.org/ruleset-schema.xsd">

  <description>Правила для CodeNarc. Содержит правило, проверяющее, что
    не используются методы API, которых не существуют</description>

  <rule-script path='classpath:CheckScriptUtilsRule.groovy' />
</ruleset>
```

```
class CheckScriptUtilsRuleRule extends AbstractAstVisitorRule {
    String name = 'CheckScriptUtilsRule'
    String description = 'Проверяет, что в скриптах не используются несуществующие методы API'
    int priority = 2
    Class astVisitorClass = CheckScriptUtilsRuleAstVisitor
}
```




```

class CheckScriptUtilsRuleAstVisitor extends AbstractAstVisitor {
    final classesToCheck = ['timeService' : 'ru.jugekb.api.TimeService']

    @Override
    public void visitMethodCallExpression(MethodCallExpression call) {
        super.visitMethodCallExpression(call)
        if (call.objectExpression instanceof VariableExpression) {
            def var = call.objectExpression.variable
            if (classesToCheck[var] != null) {
                def method = call.method.value
                def argCount = call.arguments.expressions.size();
                def utilsMethods = Class.forName(classesToCheck[var]).methods.findAll {
                    it.name == method &&
                        (it.parameterTypes.length == argCount || isVarArg(it)
                            && argCount >= it.parameterTypes.length - 1)
                }
                if (utilsMethods.isEmpty()) {
                    addViolation(call, "Method ${call.text} not found in API")
                }
            }
        }
    }
}

/**
 * В Groovy массив Object[] в качестве последнего аргумента также может быть вызван как varargs
 */
private boolean isVarArg(Method m) {
    return m.isVarArgs() || m.parameterTypes.length > 0 &&
        Object[].isAssignableFrom(m.parameterTypes.last())
}
}

```



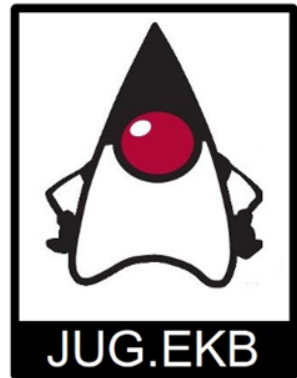
Я только запустил скрипт

А оно кэшельме бешельме шайтан консоль

@TimedInterrupt

```
import groovy.transform.TimedInterrupt
import java.util.concurrent.TimeUnit

@TimedInterrupt(value = 300L, unit = TimeUnit.SECONDS)
class MyClass {
    def method() {
        println '...'
    }
}
```



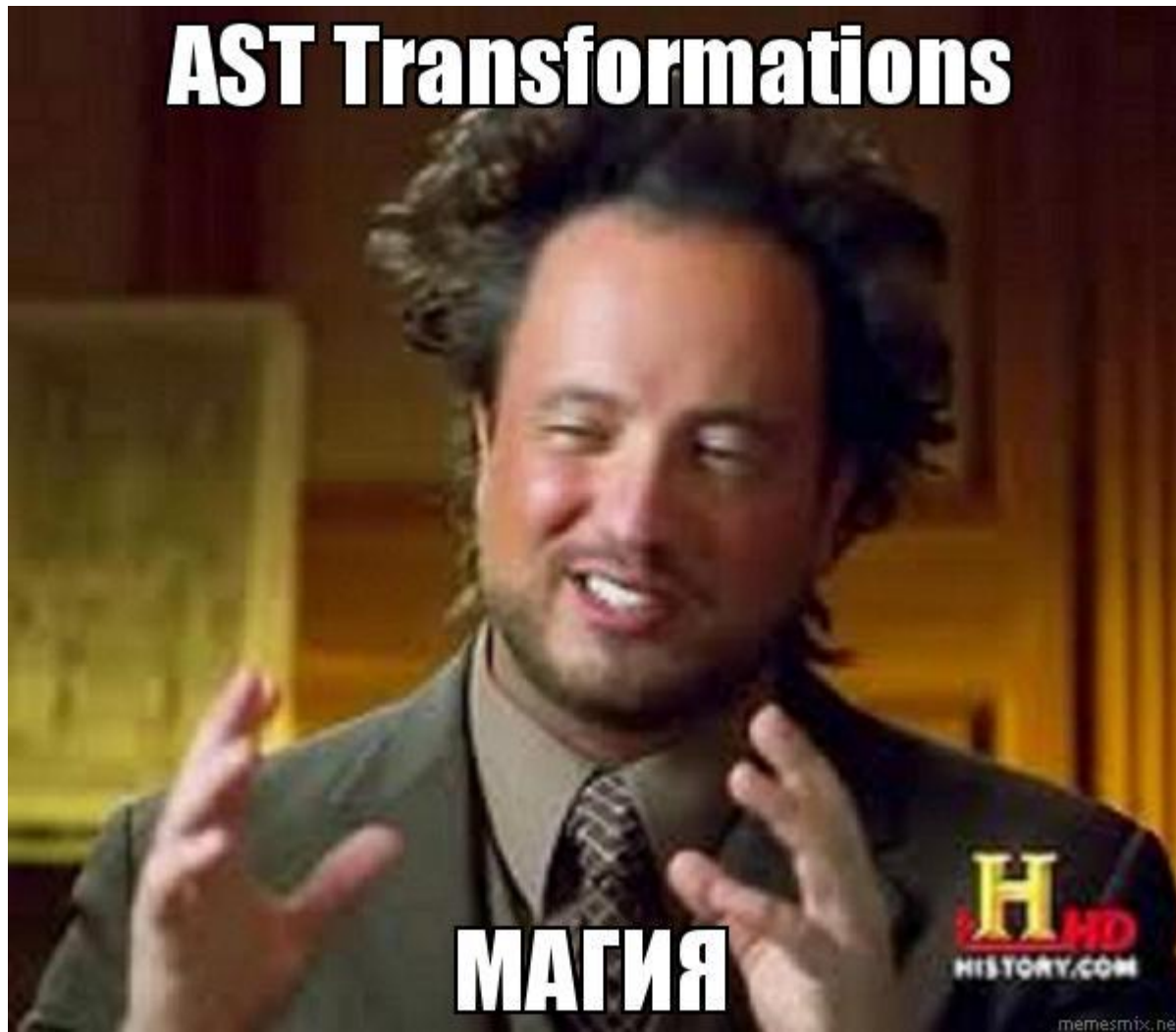
```
import java.util.concurrent.TimeUnit
import java.util.concurrent.TimeoutException

public class MyClass {
    // XXXXXX below is a placeholder for a hashCode value at runtime
    final private long timedInterruptXXXXXX$expireTime
    final private java.util.Date timedInterruptXXXXXX$startTime

    public MyClass() {
        timedInterruptXXXXXX$expireTime = System.nanoTime() +
            TimeUnit.NANOSECONDS.convert(300, TimeUnit.SECONDS)
        timedInterruptXXXXXX$startTime = new java.util.Date()
    }

    public java.lang.Object method() {
        if (timedInterruptXXXXXX$expireTime < System.nanoTime()) {
            throw new TimeoutException('Execution timed out after 300 units. ' + '
                Start time: ' + timedInterruptXXXXXX$startTime)
        }
        return this.println('...')
    }
}
```

AST Transformations



МАГИЯ

H
ISTORY
HISTORY.COM

memesmbc.net

```
public class ScriptService {

    GroovyShell shell;

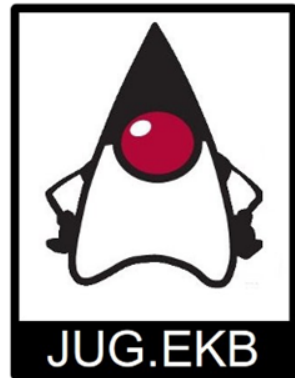
    public ScriptService() {
        Binding bindings = new Binding();
        bindings.setVariable("user", new User());
        bindings.setVariable("timeService", new TimeService());

        CompilerConfiguration configuration = new CompilerConfiguration();
        Map<String, Object> map = new HashMap<>();
        map.put("value", 5);
        map.put("unit", new PropertyExpression(
            new ClassExpression(ClassHelper.make(TimeUnit.class)), "SECONDS"));
        configuration.addCompilationCustomizers(
            new ASTTransformationCustomizer(map, TimedInterrupt.class));
        configuration.setOutput(new PrintWriter(System.out));
        |
        shell = new GroovyShell(bindings, configuration);
    }

    public Object execute(String script) throws ScriptException {
        return shell.evaluate(script);
    }
}
```

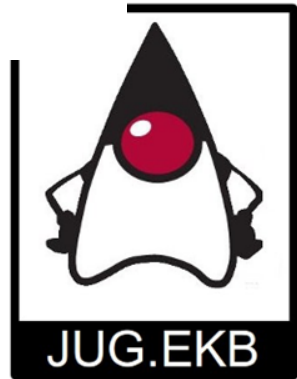

@ThreadInterrupt

```
//...  
configuration.addCompilationCustomizers(  
    new ASTTransformationCustomizer(ThreadInterrupt.class));  
//...
```



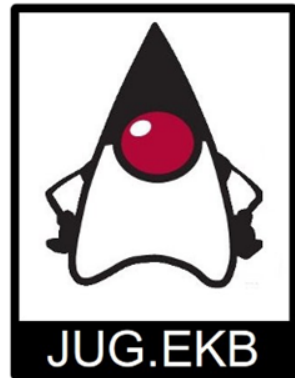
@ConditionalInterrupt

```
@ConditionalInterrupt({ counter++> 10})  
import groovy.transform.ConditionalInterrupt  
  
counter = 0  
def scriptMethod() {  
    4.times {  
        println 'executing script method...'  
    }  
}
```



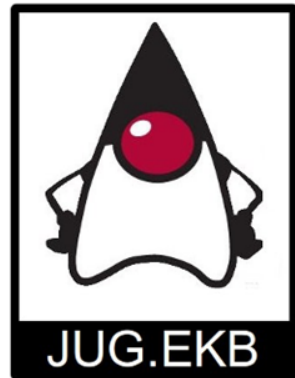
@ConditionalInterrupt

```
//...
Map<String, Object> map = new HashMap<>();
try {
    map.put("value", new ClosureExpression(Parameter.EMPTY_ARRAY,
        GeneralUtils.returnS(
            GeneralUtils.eqX(
                GeneralUtils.fieldX(
                    FieldNode.newStatic(
                        ScriptService.class, "run")),
                GeneralUtils.constX(false))
            ));
} catch (SecurityException | NoSuchFieldException e) {
    throw new RuntimeException(e);
}
configuration.addCompilationCustomizers(
    new ASTTransformationCustomizer(map, ConditionalInterrupt.class));
//...
```



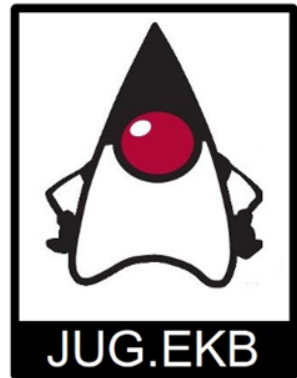
ImportCustomizer

```
//..  
ImportCustomizer importCustomizer = new ImportCustomizer();  
importCustomizer.addImports("sun.misc.Unsafe");  
importCustomizer.addStaticImport("java.lang.Double", "POSITIVE_INFINITY");  
configuration.addCompilationCustomizers(importCustomizer);  
//..
```



SecureASTCustomizer

```
//...  
SecureASTCustomizer secureASTCustomizer = new SecureASTCustomizer();  
secureASTCustomizer.setClosuresAllowed(false);  
secureASTCustomizer.setImportsBlacklist(asList("java.lang.Math"));  
// etc...  
configuration.addCompilationCustomizers(secureASTCustomizer);  
//...
```



```

private boolean isPackageAllowed = true;
private boolean isMethodDefinitionAllowed = true;
private boolean isClosuresAllowed = true;

// imports
private List<String> importsWhitelist;
private List<String> importsBlacklist;

// static imports
private List<String> staticImportsWhitelist;
private List<String> staticImportsBlacklist;

// star imports
private List<String> starImportsWhitelist;
private List<String> starImportsBlacklist;

// static star imports
private List<String> staticStarImportsWhitelist;
private List<String> staticStarImportsBlacklist;

// statements
private List<Class<? extends Statement>> statementsWhitelist;
private List<Class<? extends Statement>> statementsBlacklist;
private final List<StatementChecker> statementCheckers = new LinkedList<StatementChecker>();

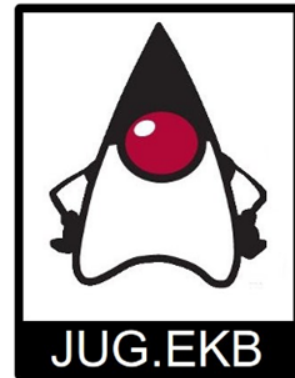
// expressions
private List<Class<? extends Expression>> expressionsWhitelist;
private List<Class<? extends Expression>> expressionsBlacklist;
private final List<ExpressionChecker> expressionCheckers = new LinkedList<ExpressionChecker>();

// tokens from Types
private List<Integer> tokensWhitelist;
private List<Integer> tokensBlacklist;

// constant types
private List<String> constantTypesWhiteList;
private List<String> constantTypesBlackList;

// receivers
private List<String> receiversWhiteList;
private List<String> receiversBlackList;

```



```
SecureASTCustomizer secure = new SecureASTCustomizer()
secure.with {
    closuresAllowed = false
    methodDefinitionAllowed = false

    importsWhitelist = []
    staticImportsWhitelist = []
    staticStarImportsWhitelist = ['java.lang.Math']

    tokensWhitelist = [
        PLUS, MINUS, MULTIPLY, DIVIDE, MOD, POWER,
        PLUS_PLUS, MINUS_MINUS, COMPARE_EQUAL, COMPARE_NOT_EQUAL,
        COMPARE_LESS_THAN, COMPARE_LESS_THAN_EQUAL,
        COMPARE_GREATER_THAN, COMPARE_GREATER_THAN_EQUAL,
    ].asImmutable()

    constantTypesClassesWhiteList = [
        Integer, Float, Long, Double, BigDecimal,
        Integer.TYPE, Long.TYPE, Float.TYPE, Double.TYPE
    ].asImmutable()

    receiversClassesWhiteList = [ Math, Integer, Float, Double, Long,
        BigDecimal ].asImmutable()
}
```

Спасибо за внимание

```
/**  
* Сергей Целовальников  
* Naumen  
* sergeicelov@gmail.com  
* @serceman  
*  
* github.com/SerCeMan/jugekb1.git  
*/
```

