BeeCon 2016

GroovyRunner
Scripting on steroids

Joost Horward
Open-T

About me

- Founder of Open-T
- Alfresco experience since 2007
- Lead dev for
  - Workflow4people
  - Alice
  - GroovyRunner

What is GroovyRunner?

- Run Groovy script in repository
- **Full access to the entire Java API**
  - Auto-inject all Spring beans
  - Some helpers to make things easier
- Full transaction control
- → It's like scripting, but with the Java API

What can you do with it?

- Repairs
- Imports/exports
- Configuration
- Test
- Analysis
- Troubleshooting

WARNING

- With great power comes great responsibility
- You can break literally anything
- alfresco-global-properties:
  - groovyrunner.requireAdmin=true
  - This is the default for obvious reasons

Installation

- Download from:

  www.open-t.nl/projects/groovyrunner

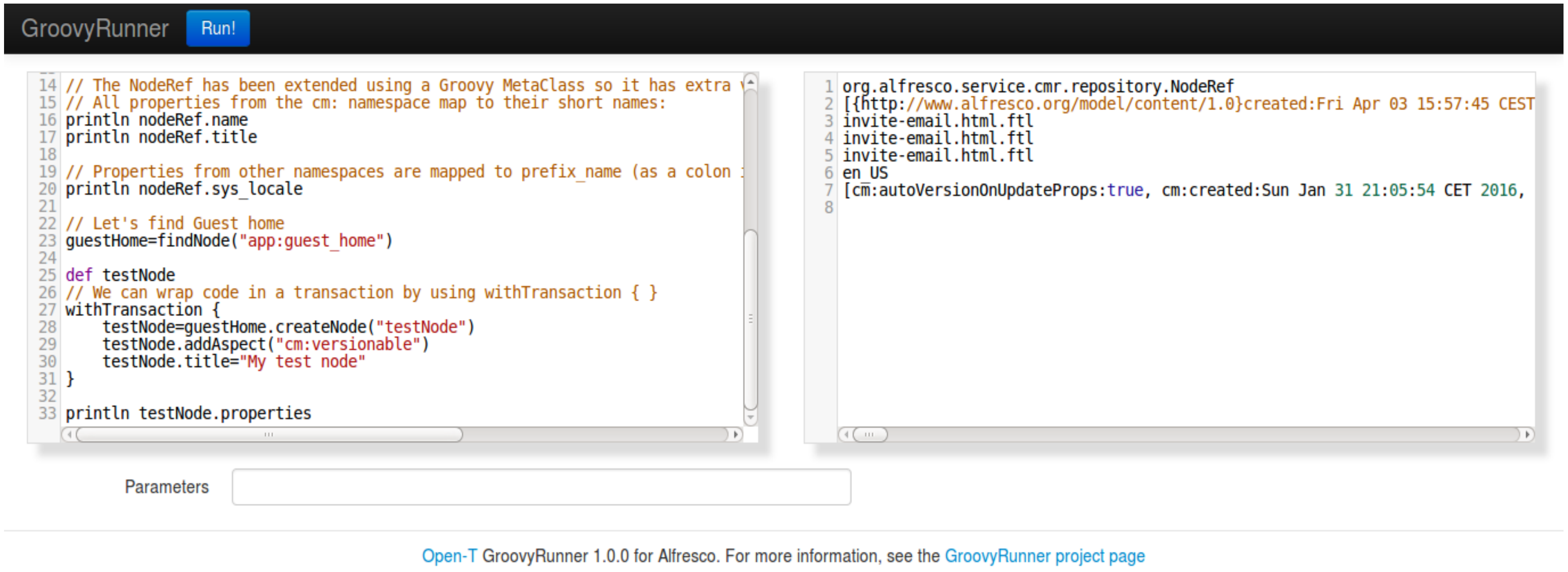
- Copy alfresco-groovy-runner.jar to tomcat/webapps/alfresco/WEB-INF/lib
- Restart alfresco

Running

- http://[hostname]:[port]/alfresco/service/open_t/groovyrunner
- Or commandline:

```
runscript.sh script.groovy
```

GroovyRunner in Action



```
// The NodeRef has been extended using a Groovy MetaClass so it has extra
// All properties from the cm: namespace map to their short names:
println nodeRef.name
println nodeRef.title

// Properties from other namespaces are mapped to prefix_name (as a colon
println nodeRef.sys_locale

// Let's find Guest home
guestHome=findNode("app:guest_home")

def testNode
// We can wrap code in a transaction by using withTransaction { }
withTransaction {
    testNode=guestHome.createNode("testNode")
    testNode.addAspect("cm:versionable")
    testNode.title="My test node"
}

println testNode.properties
```

```
org.alfresco.service.cmr.repository.NodeRef
[{http://www.alfresco.org/model/content/1.0}created:Fri Apr 03 15:57:45 CEST
invite-email.html.ftl
invite-email.html.ftl
invite-email.html.ftl
en_US
[cm:autoVersionOnUpdateProps:true, cm:created:Sun Jan 31 21:05:54 CET 2016,
```

Parameters

Open-T GroovyRunner 1.0.0 for Alfresco. For more information, see the GroovyRunner project page

Groovy in 60 seconds

- Java-like syntax
- Compiles to normal .class
- “def” - untyped (like Object)
- Closures
- Simple List and Map: Syntax [] and [:]
- Iterators (.each {})
- GString

Let's run something really simple

- Instantiate a Java Date object:

```
Date d = new Date()
println d
```

Example #1

BeeCon 2016

Access to the Java API

- Spring beans are auto-injected by name
- e.g. nodeService

```
println nodeService.getStores()
```

```
Example #2
```

BeeCon 2016

Access to the Java API

```
nodeService.getStores().each { store ->
  println "Store: ${store}"
  nodeService.getAllRootNodes(store).each { node
->
    println node
    nodeService.getProperties(node).each { name,
value ->
      println "  ${name}=${value}"
    }
  }
}
```

Example #3

BeeCon 2016

Work with Spring Beans

- Get bean names:

```
applicationContext.getBeanDefinitionNames().each
{ name -> println name }
```

- Get bean by name:

```
def gp=applicationContext.getBean("global-
properties")
gp.each { println it }
```

Example #4-5

Finding a node

- Simple helpers for finding and traversing:
  - findNode(path)
    - Path can be node names or association names
  - nodeRef.list

`Example #6`

BeeCon 2016

More Groovy in >60 seconds

- MetaClass
  - Add methods to existing classes
- Dynamic properties
  - Add properties to existing classes
  - Implement getProperty and/or setProperty on the MetaClass

Dynamic properties

- Classic way to get the title of a node:
  - NodeService.getProperty(nodeRef,ContentModel.CM_NAME)
- Dynamic property:
  - nodeRef.name
  - NodeRef.cm_name

```
println swsdp.name
println swsdp.creator
println swsdp.modifier
```

Example #7

Dynamic properties

- Easy way to get type and aspects:

```
println swsdp.type
println swsdp.aspects
```

Example #8

Transaction control

- Wrap in withTransaction { } :

```
withTransaction {
    doc.title="Test"
    doc.addAspect("cm:versionable")
}
```

Example #9

BeeCon 2016

Creating nodes

- NodeRef.createNode(name, type, properties, aspects)
- createPath(path)
- createSite(name, title, description)
- createPerson(properties, pasword)
- createGroup(name)

`Example #10`

Test helper

- Run closure with title and catch any exceptions:

```
test.run "Check versionable aspect is not present", {
    assert !node.aspects.contains("cm:versionable")
}
```

- Use “log” for output in tests

`Example #11`

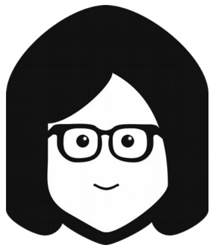
BeeCon 2016

There's more to explore

- http helper
- search
- createPath
- createSite
- upload
- putText
- createPerson
- transform
- runAs
- createGroup etc.

Questions?

- Anyone?
- Really?
- Thank you!
- You can always find me at joost@open-t.nl and often at orderofthebee IRC (xoost)



BeeCon 2016