Groovy, Grails und NetBeans

Matthias Schmidt
Software Engineer
Sun Microsystems, Inc.

Agenda

- Die Groovy Programmiersprache
- Grails Web Framework
- Groovy und Grails Unterstützung in NetBeans

Wieso Groovy?

- Dynamische Sprachen sind ein Paradigmenwechsel.
- Ist syntaktisch an Java angelehnt.
- Läuft in der JVM, kein JNI, kein Marshalling.
- GDK ist Obermenge/Anreicherung des JDK.
- JDK Klassen sofort benutzbar.
- ca. 30 % weniger LOC.

Alles ist ein Objekt!

- Darum geht sowas:

```
3.times { println 'Hi' }

[0, 1, 2].each { number ->
    println number
}

[0, 1, 2].each { println it}

def printit = { println it }
[0, 1, 2].each printit
```

Höhere Ausdrucksfähigkeit

- Optionale Semikolons:

```
System.out.println("Hello, World!");
println 'Hello, World!'
```

- Default imports:

  * java.io.*
  * java.lang.*
  * java.math.BigDecimal
  * java.math.BigInteger
  * java.net.*
  * java.util.*
  * groovy.lang.*
  * groovy.util.*

Höhere Ausdrucksfähigkeit 2

- GStrings:

```
def name = 'Guillaume'
println "$name, I'll get the car."
```

- Beans:

```
class Customer {
    Integer id
    String name
    Date dob

    static void main(args) {
        def customer = new Customer(id:1, name:"Gromit", dob:new Date())
        println("Hello ${customer.name}")
    }
}
```

Closures 1

- Code als Daten

```
doubleNum = { num -> num * 2 }
println doubleNum(4)   // 8
```

- Lesbarkeit, Flexibilität

```
doubleNum = { num -> println num * 2 }

def worker (Closure doit) {
    (10..1).each { count ->
        doit(count)
    }
}

worker { println it }
worker { doubleNum(it) }
```

Closures 2

- GDK ist genauso “closifiziert” wie das JDK5 generifiziert wurde.

```
new File('data.txt').eachLine { println it }
```

- Currying treibt das ganze auf die Spitze:

```
def adder = { x, y -> return x + y}
def addOne = adder.curry(1)

println addOne(5)
```

Collections: List, Range

- List: def list = [5, 6, 7, 8]
- Range:

```
for (i in 1..10) {
  println "Hello ${i}"
}
```

```
switch (years) {
   case 1..10: interestRate = 0.076; break;
  case 11..25: interestRate = 0.052; break;
      default: interestRate = 0.037;
}
```

Collections: Map

- Map:

```
def map = [name:"Gromit", likes:"cheese", id:1234]
assert map.name == "Gromit"
assert map.id == 1234
```

- Operations:

```
assert [1, 3, 5] == ['a', 'few', 'words']*.size()
```

GDK mixins

Type	JDK	Groovy
Array	length field	size()
Array	java.lang.reflect.Array.getLength(Array)	size()
String	length()	size()
StringBuffer	length()	size()
Collection	size()	size()
Map	size()	size()
File	length()	size()
Matcher	groupCount()	size()

Builder

```
import groovy.xml.*
def page = new MarkupBuilder()
page.html {
  head { title 'Hello' }
  body {
    ul {
      for (count in 1..10) {
        li "world $count"
} } } }
```

Was ist Grails?

- Ruby on Rails -> Groovy/Grails
- Prinzip des *“Convention over Configuration”.*
- Projektstruktur vorgegeben.
- Model-View-Controller Aufbau.
- Spring, Hibernate, SiteMesh
- Grails-Kommandos bestimmen den Arbeitsablauf.

Framework + Ablaufumgebung

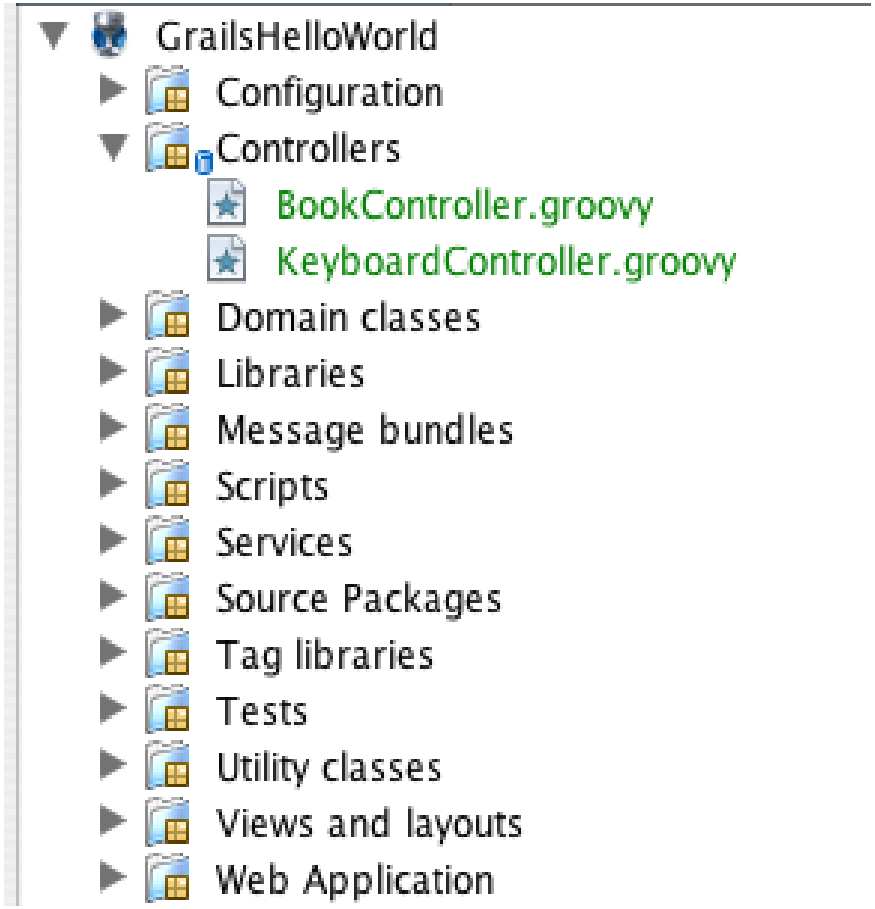
- Grails ist sowohl API, ...
- als auch Runtime (Jetty) sowie
- Shell-Kommando: “grails run-app”.

- Keine Konfiguration per XML.
- Sinnvolle Defaulteinstellungen.

Wiederverwendung

- Spring
- Hibernate
- SiteMesh
- Ant
- Plugin-Architektur (~ 70)
- AJAX per plugins, Oder Flex, oder ...

Projektstruktur



Agile/Rapid Prototyping

- grails create-app
- grails create-domain-class

Book.groovy

```
class Book {
    String title
    String author
}
```

- grails create-controller
- grails run-app

Groovy Server Pages

- Vergleichbar mit JSP.
- Per Tag-Libs erweiterbar.

```
<html>
  <head>
    <title>Our books</title>
  </head>
  <body>
    <ul>
      <g:each it="books">
        <li>${it.title} (${it.author.name})</li>
      </g:each>
    </ul>
  </body>
</html>
```

Scaffolding

- Grails create-controller

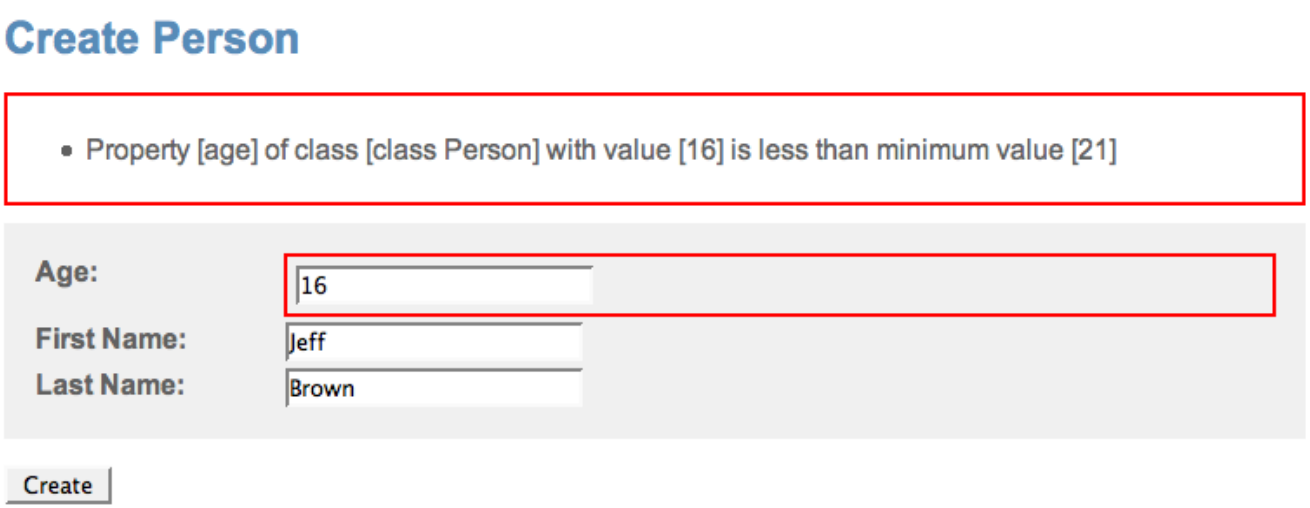
```
class BookController {
    scaffold = true
}
```

Book List

Id	Title	Author	Category	Release Date	
1	The Shining	Stephen King	Fiction	2006-01-27 11:51:00.0	Show
2	Pet Semetary	Stephen King	Fiction	2005-12-27 11:52:00.0	Show

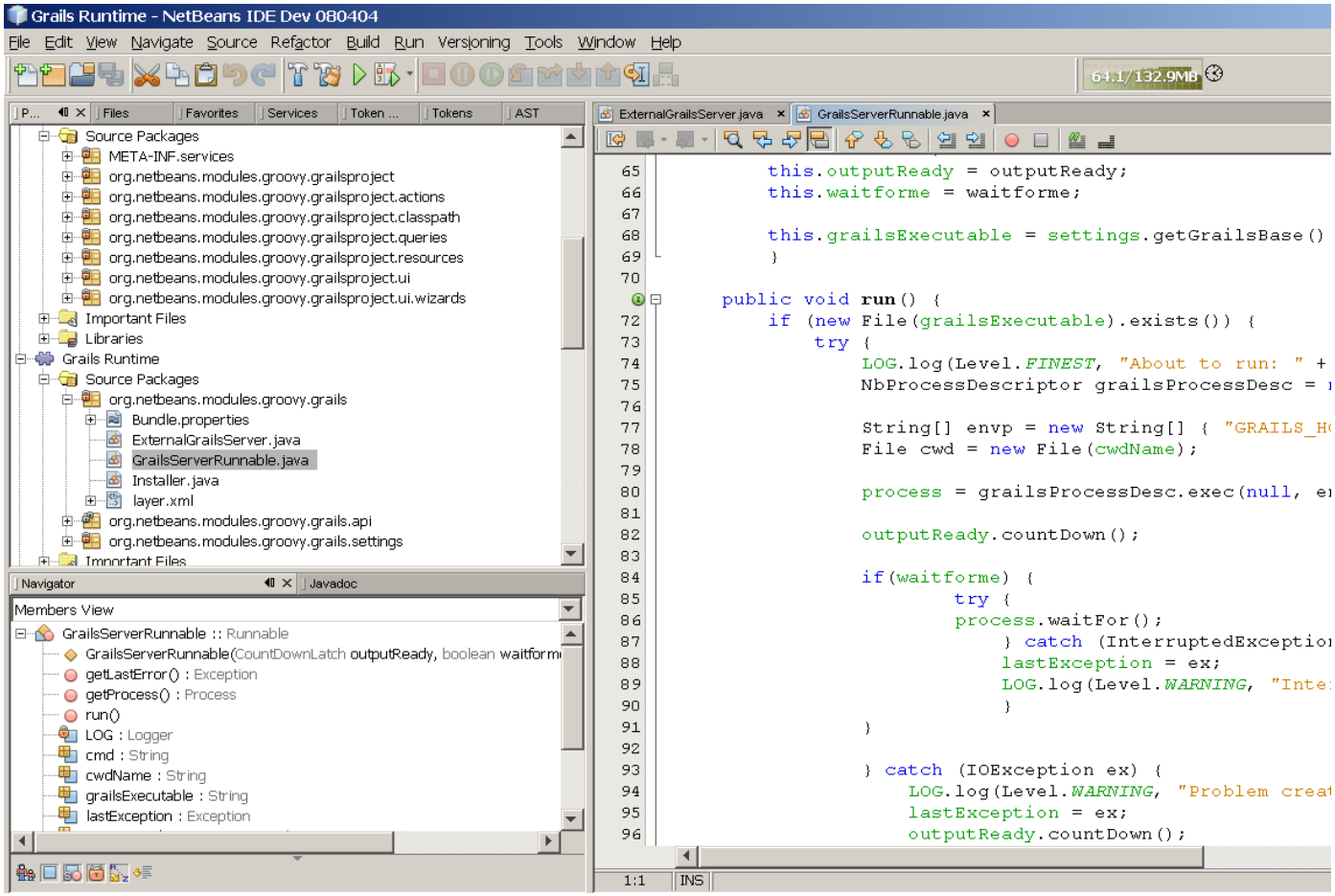
Constrains

```
class Person {
    String firstName
    String lastName
    Integer age

    static constraints = {
        age(min:21)
        lastName(length:2..35)
        firstName(length:2..35)
    }
}
```



Datenbankanbindung

```
environments {
    development {
        dataSource {
            dbCreate = "create-drop" // one of 'create', 'create-drop','update'
            url = "jdbc:hsqldb:mem:devDB"
        }
    }
    test {
        dataSource {
            dbCreate = "update"
            url = "jdbc:hsqldb:mem:testDb"
        }
    }
    production {
        dataSource {
            dbCreate = "update"
            url = "jdbc:hsqldb:file:prodDb;shutdown=true"
        }
    }
}
```

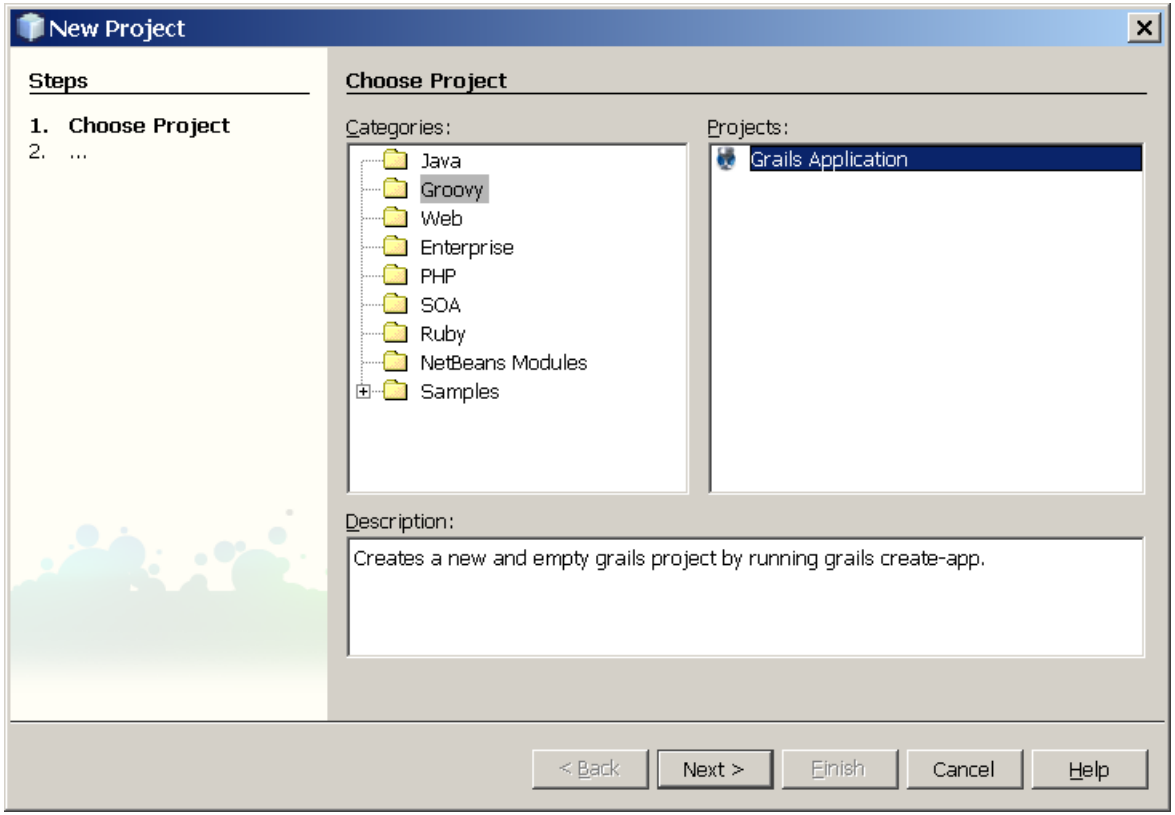

Grails Runtime - NetBeans IDE Dev 080404
File Edit View Navigate Source Refactor Build Run Versioning Tools Window Help
64.1/132.9MB
Files
Favorites
Services
Token ...
Tokens
AST
Source Packages
META-INF.services
org.netbeans.modules.groovy.grailsproject
org.netbeans.modules.groovy.grailsproject.actions
org.netbeans.modules.groovy.grailsproject.classpath
org.netbeans.modules.groovy.grailsproject.queries
org.netbeans.modules.groovy.grailsproject.resources
org.netbeans.modules.groovy.grailsproject.ui
org.netbeans.modules.groovy.grailsproject.ui.wizards
Important Files
Libraries
Grails Runtime
Source Packages
org.netbeans.modules.groovy.grails
Bundle.properties
ExternalGrailsServer.java
GrailsServerRunnable.java
Installer.java
layer.xml
org.netbeans.modules.groovy.grails.api
org.netbeans.modules.groovy.grails.settings
Important Files
Navigator
Javadoc
Members View
GrailsServerRunnable :: Runnable
GrailsServerRunnable(CountDownLatch outputReady, boolean waitforme
getLastError() : Exception
getProcess() : Process
run()
LOG : Logger
cmd : String
cwdName : String
grailsExecutable : String
lastException : Exception
ExternalGrailsServer.java
GrailsServerRunnable.java
this.outputReady = outputReady;
this.waitforme = waitforme;
this.grailsExecutable = settings.getGrailsBase()
}
public void run() {
if (new File(grailsExecutable).exists()) {
try {
LOG.log(Level.FINEST, "About to run: " +
NbProcessDescriptor grailsProcessDesc =
String[] envp = new String[] { "GRAILS_H
File cwd = new File(cwdName);
process = grailsProcessDesc.exec(null, e
outputReady.countDown();
if(waitforme) {
try {
process.waitFor();
} catch (InterruptedExceptio
lastException = ex;
LOG.log(Level.WARNING, "Inte
}
}
} catch (IOException ex) {
LOG.log(Level.WARNING, "Problem crea
lastException = ex;
outputReady.countDown();
1:1
INS

NetBeans Integration

- GSF Framework als Basis des Scripting-Supports.
- Benutzer: JavaScript, PHP, Scala, Erlang, Groovy, Ruby usw.
- GSF Basis seit NetBeans 6.0.
- Groovy/Grails vorraussichtlich Teil von NB 6.5.
- Grails-Projekte werden nicht verändert.

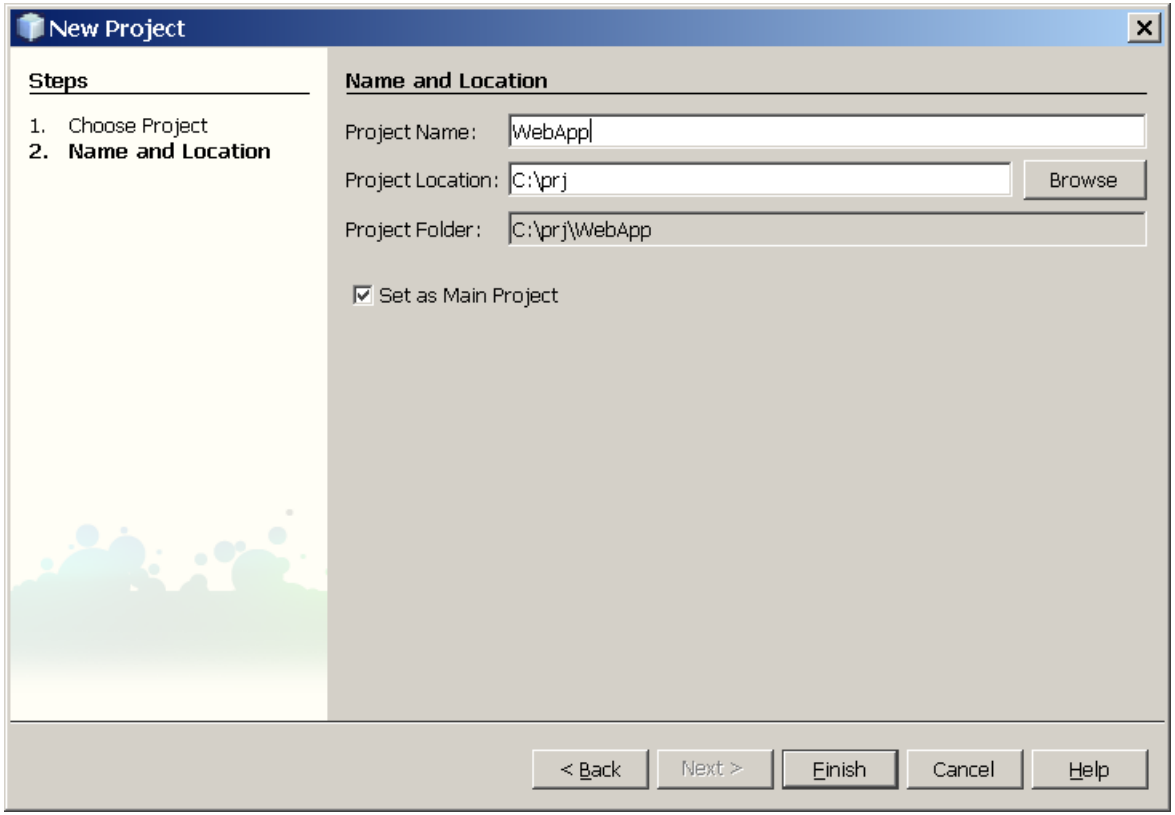
NetBeans

- Laden u. Erzeugen von Projekten per Wizard:

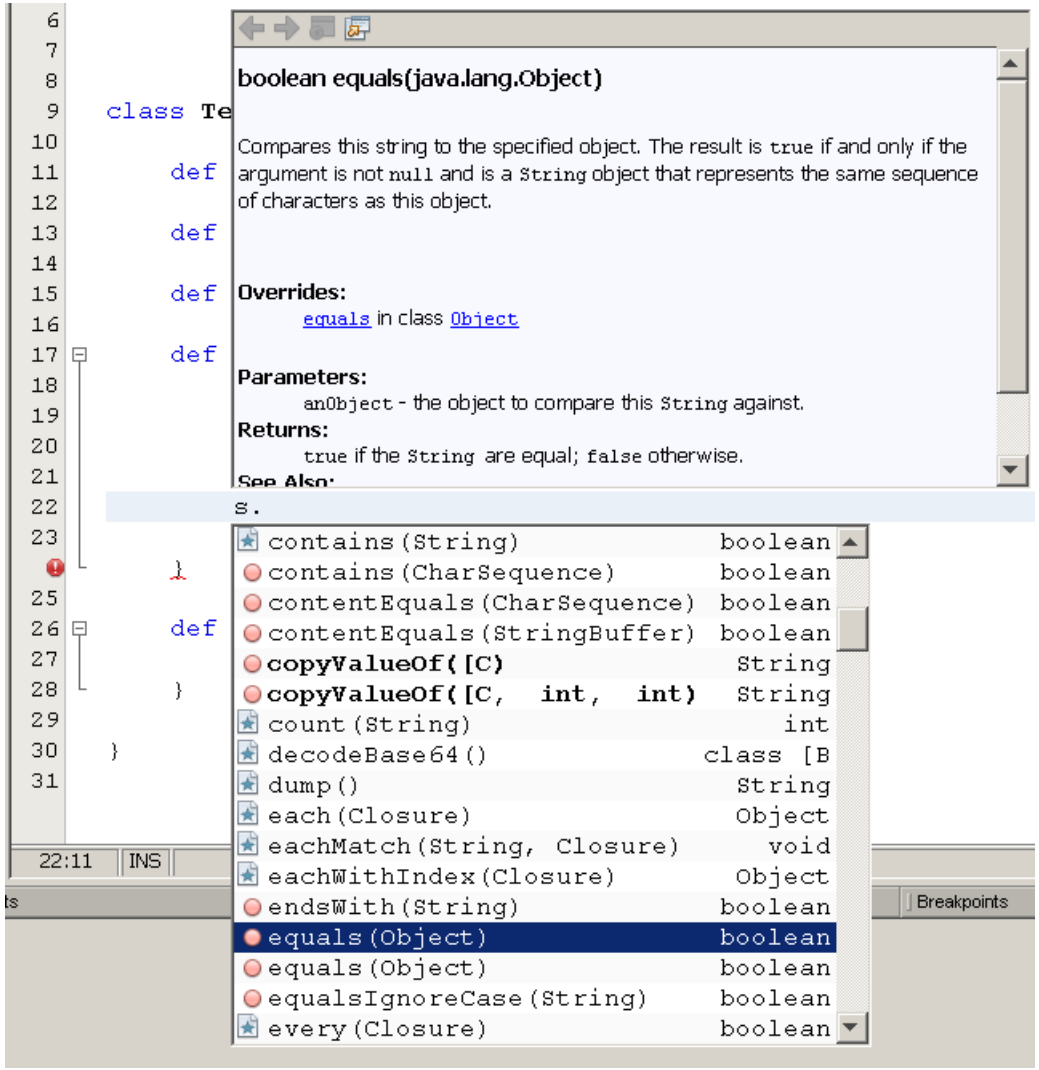


NetBeans

- Laden u. Erzeugen von Projekten per Wizard:



NetBeans



Code Completion

NetBeans

- Code Folding

```
import java.util.HashSet;
import java.util.Iterator;
import java.util.List;
import java.util.Set;

/** This is a multiline
 *  comment.
 *  I wanna have it folded!
 */

new Test().method()


...

class Test {

    /* Not a JavaDoc one,
     * just a simple comment
     *  I wanna have it folded as well
     */


    def method (){...}

    def myclosure = {...}
}
```

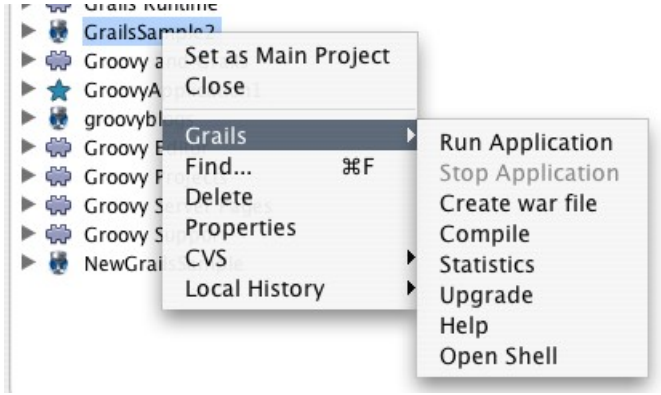
NetBeans

- Syntax Highlighting

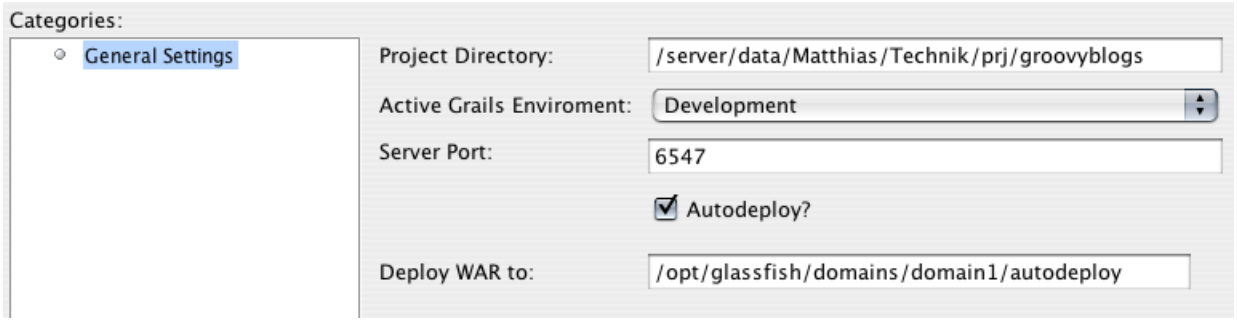
```
class AccountController extends BaseController {

        MailService mailService
        CryptoService cryptoService
        FeedService feedService
        SearchService searchService


    def index = { redirect(action:'edit',params:params) }


    def edit = {
        def account = Account.get( session.account.id )

        if(!account) {
                flash.message = "Account not found with id ${session.account.id}"
                redirect(action:'list')
        }
        else {
            return [ account : account ]
        }
    }

    def update = {
        def account = Account.get( params.id )
        if(account.id == session.account.id) {
             account.properties = params
             account.password = cryptoService.sha1(params.password.getBytes())
            if(account.save()) {
                flash.message = "Updated successfully"
                redirect(action:'edit', model:[account:account])
            }
            else {
                render(view:'edit',model:[account:account])
            }
```

NetBeans

- Start/Stop Server

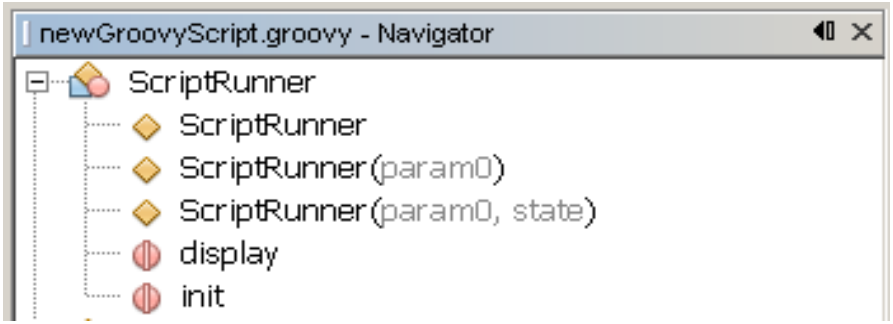


- Customizing

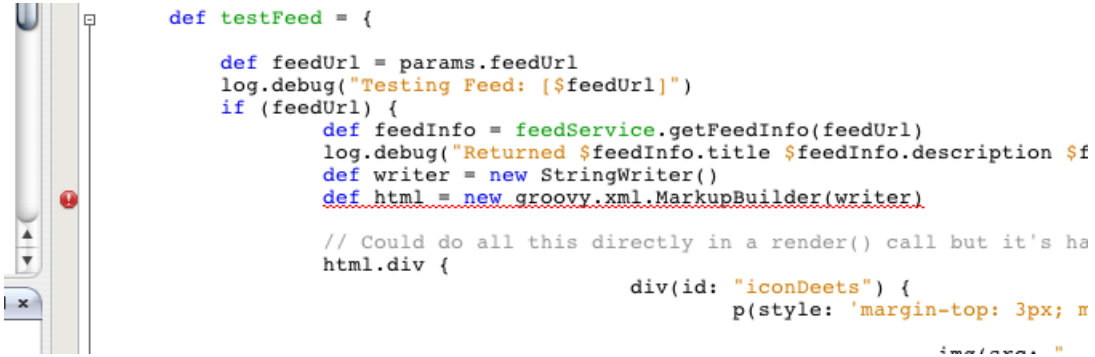


NetBeans

- Navigator View

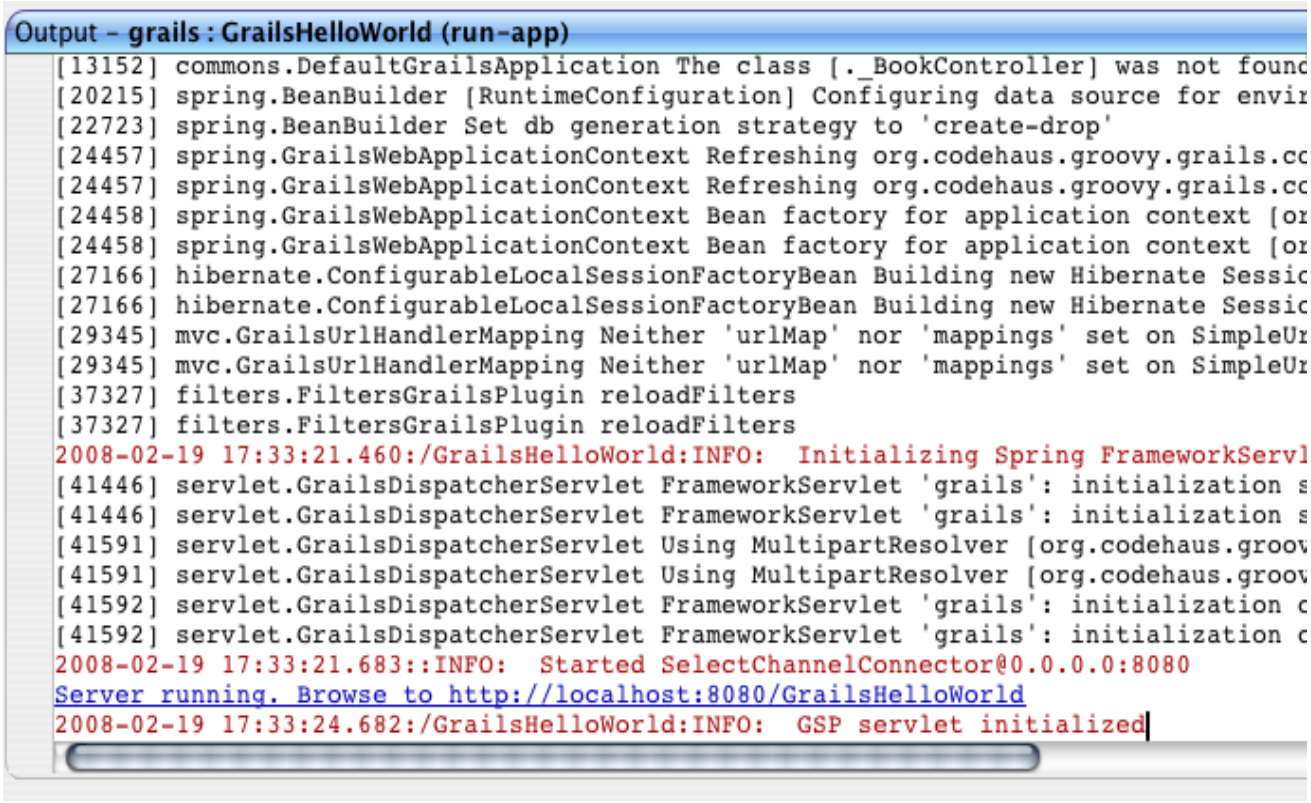


- Error-Marking



NetBeans

- Ausgabefenster



Links

- http://martin.adamek.sk
- http://www.netbeans.org
- http://blogs.sun.com/tor/
- http://hg.netbeans.org/main/summary
- http://hg.netbeans.org/main/contrib/summary
- http://wiki.netbeans.org/WorkingWithNetBeansSources
- http://grails.org/
- http://groovy.codehaus.org/

Thanks

Matthias Schmidt
schmidtm@sun.com