



**Activate Persistence
Framework
Beginner Guide
version 0.3**

Histórico

Versão	Data	Comentário	Autor
0.1	23/05/2012	Estruturação do documento	Aristofânio
0.2	27/05/2012	Instruções de Instalação	Aristofânio
0.3	30/05/2012	Preparação do SBT e do Activate (cont.)	Aristofânio
0.4	06/06/2012	Finalização do teste de 1 minuto	Aristofânio

Revisão

Versão	Data	Comentário	Autor

Índice

1. Introdução.....	4
2. Sobre STM.....	5
3. Preparação do Ambiente e Instalação.....	6
4. Guia de 1 minuto.....	12
5. Guia de 3 minutos.....	14
6. Guia de 15 minutos.....	15
7. Referências.....	16

1. Introdução

O Activate é um framework para persistência de objetos em Scala. Ele é um [STM](#) (Software Transactional Memory) durável, com persistência plugável. Seu núcleo é o [RadonSTM](#), que provê um poderoso mecanismo de controle de transações em memória, análogo às transações dos bancos de dados, para controle de concorrência otimista.

A forma de persistência depende do plugin utilizado, sendo que através desse plugin pode-se utilizar persistência em diferentes paradigmas de armazenamento, tais como:

- relacional: que utiliza-se das configurações do JDBC;
- prevalência: que utiliza o framework Prevayler para manter os dados em memória;
- não relacional: que utiliza-se de drivers nativos de comunicação com o MongoDB, um DBMS-NOSQL (Sistemas Gerenciadores de Bancos de Dados Não-SQL).

2. Sobre STM

<explicar STM>

3. Preparação do Ambiente e Instalação

3.1. Ambiente

Para a instalação do Activate é necessário a configuração do ambiente, cujos requisitos básicos são:

- Plataforma: Java Runtime Environment (JRE);
- Compilador: Scala;
- Sistema operacional: Ubuntu 11.xx ou 12.xx e Win7;
- Bancos de Dados: MongoDB ou PostgreSQL ou MySQL;
- Activate Persistence Framework;
- Simple Build Tools (SBT)

Os testes foram realizados utilizando as seguintes versões:

Software	Versão	Link para download
JDK	1.6_18	http://www.oracle.com/technetwork/java/javase/downloads/jdk-6u32-downloads-1594644.html
Scala	2.9.2	http://www.scala-lang.org/downloads
MongoDB	2.0.4	http://www.mongodb.org/downloads
PostgreSQL	8.4	http://www.postgresql.org/download/
MySQL	5.5	http://dev.mysql.com/downloads/
Activate (sources)	1.0	https://nodeload.github.com/fwbrasil/activate/zipball/master
SBT	0.11.3	http://www.scala-sbt.org/download.html

3.2. Dependências

Para a compilação e execução do Activate algumas dependencias precisam ser resolvidas antes. Apesar destas dependências serem baixadas automaticamente pelo SBT, estão listadas abaixo caso algum desenvolvedor deseje construir sua própria maneira de compilar e distribuir o Activate. Todas estas bibliotecas podem ser encontradas no link <https://github.com/arigarcia/scala-activate/tree/master/activate-dist>.

Lista de dependências

commons-collections-3.2.1.jar
dom4j-1.6.jar
google-collections-1.0.jar
grizzled-slf4j_2.9.1-0.6.6.jar
guava-r08.jar
javassist-3.16.1-GA.jar
joda-convert-1.2.jar
joda-time-2.0.jar
jug-2.0.0-lgpl.jar
log4j-1.2.16.jar
logback-core-0.9.29.jar
objenesis-1.2.jar
radon-stm_2.9.2-1.0-SNAPSHOT.jar
reflections-0.9.5.jar
slf4j-api-1.6.4.jar
slf4j-log4j12-1.6.4.jar
java-uuid-generator-3.1.3.jar

Outras dependências geradas na compilação do Activate

activate-jdbc_2.9.2-1.0-SNAPSHOT.jar
activate-prevayler_2.9.2-0.9-SNAPSHOT.jar

Caso alguém queira pular esta etapa e ir direto para a parte de codificação, está disponível um arquivo com todas as dependências e com o biblioteca do Activate pronta para download e distribuição no seguinte abaixo:

<https://github.com/arigarcia/scala-activate/tree/master/activate-dist/dist>

3.3. Instalação do SBT

3.3.1. Ubuntu

Faça o download do arquivo compactado do SBT e depois o desempacote em um diretório conhecido. Para o teste utilizamos o /home/ubuntu/dev/sbt.

```
$ cd /home/ubuntu/dev  
$ wget http://scalasbt.artifactoryonline.com/scalasbt/sbt-native-packages/org/scala-  
sbt/sbt-launcher/0.11.3/sbt.tgz  
$ tar -xvf sbt.tgz  
$ cd sbt
```

Para testar digite (dentro do diretório /home/ubuntu/dev/sbt) o comando abaixo e então deverá aparecer uma listagem de opções sobre o SBT.

```
$ ./bin/sbt -help

Usage: sbt [options]
-h | -help      print this message
-v | -verbose   this runner is chattier
-d | -debug     set sbt log level to debug
-no-colors      disable ANSI color codes
...
```

Para melhorar o trabalho com o SBT você ainda pode colocá-lo dentre os comandos disponíveis digitando o comando abaixo:

```
$ sudo ln -fs /home/ari/dev/sbt/bin/sbt /usr/local/bin/
```

Faça o teste novamente:

```
$ sbt -help

Usage: sbt [options]
-h | -help      print this message
-v | -verbose   this runner is chattier
-d | -debug     set sbt log level to debug
-no-colors      disable ANSI color codes
...
```

3.3.2. Windows

Crie uma pasta em um local conhecido para a ferramenta SBT. Para nossos testes

criamos “[c:\sbt](#)”.

```
> md c:\sbt  
> cd c:\sbt
```

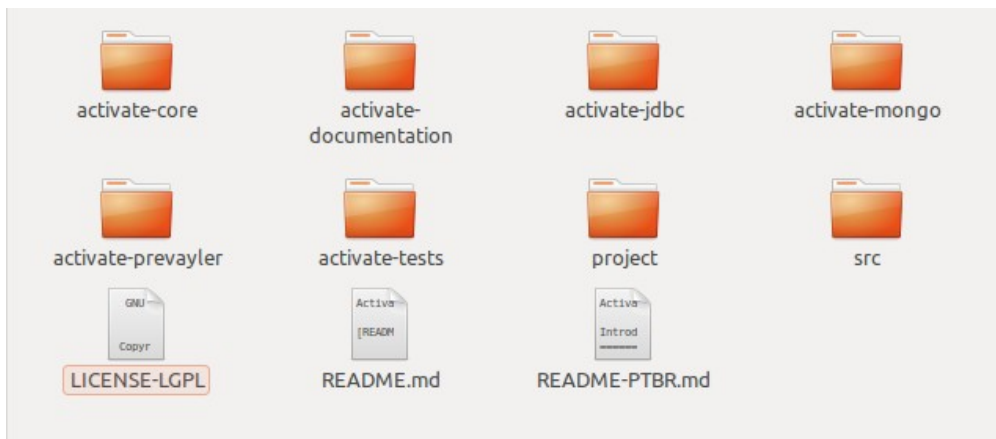
Agora faça o download do arquivo compactado do SBT (recomendamos o arquivo do tipo zip) e depois desempacote-o no diretório “c:\sbt”. Para testar execute o arquivo de lote “sbt.bat” que está dentro do diretório bin. Se tudo estiver funcionando corretamente aparecerá informações sobre as opções de execução do SBT.

```
> cd c:\sbt  
> bin\sbt.bat -help  
  
Usage: sbt [options]  
-h | -help      print this message  
-v | -verbose   this runner is chattier  
-d | -debug     set sbt log level to debug  
-no-colors      disable ANSI color codes  
...
```

3.4. Instalação do Activate

Faça o download do código fonte do Activate e desempacote em uma pasta conhecida. No nosso caso utilizamos a pasta “/home/ubuntu/dev/activate” (para linux) e “c:\activate” para windows.

Ao final, a pasta contendo todos os códigos fontes do Activate será ter a seguinte estrutura:



Dentro da pasta, a partir do console do window ou do ubuntu, você deve digitar o comando abaixo. Após isto, todas dependências serão resolvidas¹.

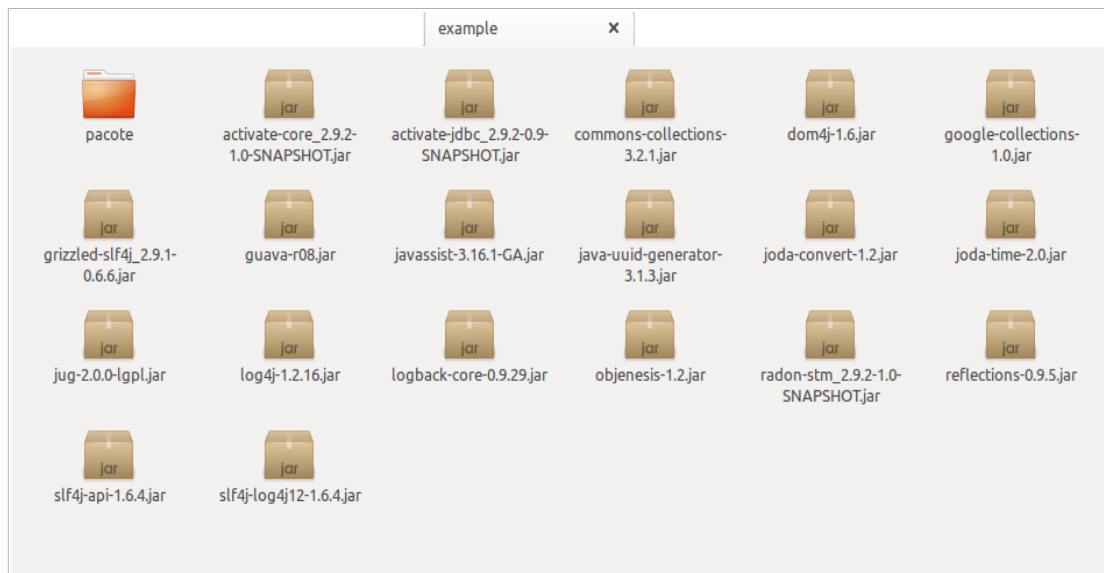
```
> cd c:\activate  
> sbt  
...
```

Ao final, o SBT ficará aguardando mais alguns comandos, então digite:

```
sbt> compile  
...  
sbt> package  
...
```

Com isto você terá todos os pacotes disponíveis, em cada um dos projetos, dentro da pasta "target/scala-2.9.2". Reuna todos os arquivos de dependência, e as bibliotecas gerados para o activate-core, activate-jdbc e activate-prevayler em uma nova pasta (conforme a imagem abaixo), a qual será utilizada como nosso ambiente de testes.

¹ Existem duas modificações que devem ser realizadas, no caso de haver erro de dependência por causa das versões das bibliotecas: 1) dentro da pasta "project", no arquivo "plugins.sbt" comente a instrução **`addSbtPlugin("com.typesafe.sbteclipse" % "sbteclipse-plugin" % "2.0.0-M2")`**; 2) Altere a instrução **`libraryDependencies <+= sbtVersion(v => "com.github.siasia" %% "xsbt-web-plugin" % (v + "-0.2.11"))`** para **`libraryDependencies <+= sbtVersion(v => "com.github.siasia" %% "xsbt-web-plugin" % ("0.11.2-0.2.11"))`**.



A pasta “pacote” sera utilizada, criada posteriormente, será utilizada como o pacote principal para nossos códigos².

Para testar se tudo ocorre bem, adicione todas as bibliotecas ao classpath e execute o seguinte comando:

```
$ scala -cp activate-core_2.9.2-1.0-SNAPSHOT.jar:activate-jdbc_2.9.2-0.9-SNAPSHOT.jar:commons-collections-3.2.1.jar:dom4j-1.6.jar:google-collections-1.0.jar:guava-r08.jar:grizzled-slf4j_2.9.1-0.6.6.jar:javassist-3.16.1-GA.jar:java-uuid-generator-3.1.3.jar:joda-convert-1.2.jar:joda-time-2.0.jar:jug-2.0.0-lgpl.jar:log4j-1.2.16.jar:logback-core-0.9.29.jar:objenesis-1.2.jar:radon-stm_2.9.2-1.0-SNAPSHOT.jar:reflections-0.9.5.jar:slf4j-api-1.6.4.jar:slf4j-log4j12-1.6.4.jar
```

```
Welcome to Scala version 2.9.2 (Java HotSpot(TM) Server VM, Java 1.6.0_18).
Type in expressions to have them evaluated.
Type :help for more information.
```

```
scala> import net.fwbrasil.activate.ActivateContext
import net.fwbrasil.activate.ActivateContext
```

```
scala> print (ActivateContext.getClass())
class net.fwbrasil.activate.ActivateContext$
```

Uma observação: verifique que o comando “-noverify” foi introduzido como um parâmetro para a VM, conforme solicita os criadores do Activate. Estuda-se uma mudança no Activate para remover este parâmetro.

Se o fluxo de comando ocorrer da maneira como descrito acima, significa que o Activate está disponível no classpath para iniciarmos as atividades de testes.

² O Activate possui um BUG conhecido, que já está sendo resolvido, que o limita a utilizar estruturas de pacotes, sob pena de ser lançada uma exceção – NullPointerException – na execução de uma aplicação com o Activate.

4. Guia de 1 minuto

Para nosso primeiro exemplo, iremos mostrar uma forma simples de armazenar os dados em memória e recuperá-los utilizando o Preveyleyler. Para tanto crie três arquivos: o primeiro deve conter os dados de configuração do contexto que você deseja (*memoContext.scala*); o segundo deve conter as classes que serão utilizadas como entidade (*person.scala*) e o terceiro deverá conter as operações de manipulação de dados (*testMemContext.scala*).

```
//file: memoContext.scala
package pacote

import net.fwbrasil.activate.ActivateContext
import net.fwbrasil.activate.storage.memory.MemoryStorage

object memoContext extends ActivateContext {
  def contextName = "memoryContext"
  val storage = new MemoryStorage
}
```

```
//file: person.scala
package pacote

import net.fwbrasil.activate.entity.Entity

abstract class Person(var name: String) extends Entity {}
class NaturalPerson(name: String, var motherName: String) extends Person(name){}
class LegalPerson(name: String, var director: NaturalPerson) extends Person(name){}
```

```
//file: testMemContext.scala
package pacote

import pacote._
import pacote.memoContext._

object testMemContext {

  def main(args: Array[String]) {
    //transação 1
    transactional {
      val person = new NaturalPerson("John", "Marie")
      println(person.name)
      person.name = "John2"
      println(person.name)
    }
  }
}
```

```
//transação 2
transactional {
  val result =
    query {
      (person: Person) => where(person.name := "John2") select(person)
    }
  for (person <- result){
    var p = person.asInstanceOf[NaturalPerson]
    println(p.motherName)
  }
}
}
```

Abra o console e compile os arquivos³:

```
$ scalac -J-noverify -cp activate-core_2.9.2-1.0-SNAPSHOT.jar:activate-jdbc_2.9.2-0.9-SNAPSHOT.jar:commons-collections-3.2.1.jar:dom4j-1.6.jar:google-collections-1.0.jar:guava-r08.jar:grizzled-slf4j_2.9.1-0.6.6.jar:javassist-3.16.1-GA.jar:java-uuid-generator-3.1.3.jar:joda-convert-1.2.jar:joda-time-2.0.jar:jug-2.0.0-lgpl.jar:log4j-1.2.16.jar:logback-core-0.9.29.jar:objenesis-1.2.jar:radon-stm_2.9.2-1.0-SNAPSHOT.jar:reflections-0.9.5.jar:slf4j-api-1.6.4.jar:slf4j-log4j12-1.6.4.jar pacote/memoContext.scala pacote/person.scala pacote/testMemContext.scala
```

Agora execute:

```
$ scala -J-noverify -cp activate-core_2.9.2-1.0-SNAPSHOT.jar:activate-jdbc_2.9.2-0.9-SNAPSHOT.jar:commons-collections-3.2.1.jar:dom4j-1.6.jar:google-collections-1.0.jar:guava-r08.jar:grizzled-slf4j_2.9.1-0.6.6.jar:javassist-3.16.1-GA.jar:java-uuid-generator-3.1.3.jar:joda-convert-1.2.jar:joda-time-2.0.jar:jug-2.0.0-lgpl.jar:log4j-1.2.16.jar:logback-core-0.9.29.jar:objenesis-1.2.jar:radon-stm_2.9.2-1.0-SNAPSHOT.jar:reflections-0.9.5.jar:slf4j-api-1.6.4.jar:slf4j-log4j12-1.6.4.jar pacote.testMemContext.scala

log4j:WARN No appenders could be found for logger (pacote.memoContext$).
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
John
John2
Marie
```

Verifique que a construção das transações é extremamente simples. Na transação 1, os dados da entidade Person foram salvos automaticamente na memória. Na segunda transação, utilizando-se da sintaxe de consulta do Activate, foram recuperados os dados deste Person, utilizando-se como filtro o nome.

No próximo exemplo será explorado um pouco mais a sintaxe e a utilização de Bancos de Dados Relacionais.

³ Em outras versões deste documento serão criados os passos-a-passos utilizando o console do Scala e o SBT.

5. Guia de 3 minutos

<explicar STM>

6. Guia de 15 minutos

<explicar STM>

7. Referências

<https://github.com/fwbrasil/activate/blob/master/README-PTBR.md>

<http://www.mongodb.org/display/DOCS/Java+Tutorial>

<http://pt.wikipedia.org/wiki/Prevayler>

http://prevayler.org/old_wiki/Welcome.html