

Guia itexto

Essencial

Usando Jenv

Henrique Lobo Weissmann



Configurando seu ambiente de desenvolvimento Java com Jenv

Henrique Lobo Weissmann

Version 1.0, 20/10/2015

Índice

Sobre este guia	1
Sobre a itexto	2
Como instalar o Jenv	4
Linux	4
Windows	4
Conceito importante: candidatos (Candidates)	5
Instalando um candidato	6
Escolhendo a versão padrão de um candidato	7
E como faço para desinstalar?	7
Usando uma versão diferente do candidato	8
Lidando com repositórios	9
Atualizando o próprio Jenv	9
Hackeando o Jenv	10
Criando um link simbólico no Linux	10
Criando um link simbólico no Windows	11
Link simbólico ou cópia?	11
Jenv ou SDKMan?	12

Sobre este guia

Uma das dificuldades mais comuns enfrentada por desenvolvedores Java diz respeito à configuração do seu ambiente de trabalho quando precisamos lidar com diferentes versões do JDK, ferramentas, tais como Maven, Ant, Gradle ou mesmo frameworks que oferecem interfaces de linha de comando, como Grails, Spring Boot ou Vert.x.

No dia a dia de desenvolvedores que se encaixem neste perfil não é raro se ver repetidas vezes alterando variáveis de ambiente como o PATH durante o dia. Muitos inclusive criam scripts que configuram o ambiente de acordo com o projeto no qual se está trabalhando.

Visando resolver estes problemas surgiu o Jenv ('Java Environment Manager' - <http://jenv.io>). Sua função é esta: gerenciar em seu computador as diferentes versões das ferramentas baseadas em Java com as quais você trabalha em seu dia a dia. Pense no Jenv como uma evolução dos scripts que mencionei no parágrafo acima.

Imagine nunca mais precisar alterar uma variável de ambiente quando precisar alterar a sua versão do JDK ou seu framework de trabalho. Jenv te permite isto de uma forma muito fácil.

Pense na seguinte situação: você trabalha com diferentes projetos baseados no framework Grails (<http://www.grails.org>) e também do JDK. Com Jenv, como você faz para usar a versão 1.6.0_45 do JDK com a 1.3.9 do Grails? Assim:

```
jenv use java 1.6.0_45  
jenv use grails 1.3.9
```

Outra tarefa que pode ser um pouco chata é instalar estes componentes. Como isto seria feito com Jenv? Usando dois comandos:

```
jenv install grails 1.3.9  
jenv install java 1.6.0_45
```

Fácil, não? Neste guia rápido iremos entender como o Jenv funciona.

Sobre a itexto



Somos uma consultoria focada no desenvolvimento de software sediada em Belo Horizonte. Desenvolvemos sistemas de altíssima qualidade e ajudamos nossos clientes na execução desta tarefa através da nossa consultoria.

Atualmente nas seguintes frentes:

Treinamentos para empresas

Ajudamos a sua empresa a dominar as tecnologias nas quais somos referência como, por exemplo, Groovy, Grails, Spring, Java EE

Treinamentos para desenvolvedores independentes

Através da [Formação itexto](#) oferecemos treinamentos voltados para desenvolvedores independentes nas tecnologias que dominamos.

Consultoria

Projeto e crítica arquitetural, suporte a equipes de desenvolvimento e auxílio de aquisição

Desenvolvimento de sistemas customizados

Atuamos no projeto, construção e implantação de soluções customizadas para nossos clientes

Evolução de sistemas legados

Extraímos máximo valor da infraestrutura que nossos clientes já possuem

Atuamos em projetos tanto no Brasil quanto no exterior nas mais variadas áreas: financeira, comércio eletrônico, mineração, saúde, startups, produtos alimentícios, engenharia, tecnologia da informação, educação e pesquisa.

Além disto também atuamos junto com a comunidade de desenvolvimento de software a partir de algumas das nossas iniciativas:

Groovy & Grails Brasil

uma das maiores comunidades do mundo de desenvolvedores que usam a linguagem Groovy e tecnologias relacionadas.

/dev/All

nosso agregador de blogs de desenvolvedores

/dev/Kico

blog do nosso co-fundador, Henrique Lobo Weissmann (autor deste guia), que é considerado uma das principais referências sobre Groovy & Grails no Brasil

Formação itexto

aonde ministramos nossos treinamentos para desenvolvedores independentes nas tecnologias em que somos referência

Você pode entrar em contato conosco através do nosso site (<http://www.itexto.com.br>), telefone (31 2555-1979) ou e-mail (administrativo@itexto.com.br).

Como instalar o Jenv

Linux

Na sua interface do Terminal, basta executar o comando abaixo:

```
curl -L -s get.jenv.io | bash
```

Logo na sequência, reinicie sua interface de linha de comando para poder usar o Jenv ou, se preferir, execute o comando abaixo:

```
source /[sua pasta home]/.jenv/bin/jenv-init.sh
```

Para testar sua instalação, basta executar o comando `jenv help`.

Windows

A instalação no Windows deve ser feita usando o PowerShell a partir da versão 4.0. Se sua versão do Windows for anterior à 8.1, os seguintes componentes devem estar pré-instalados em sua máquina:

- .Net Framework 4.5 (ou posterior)
- Windows Management Framework 4.0

Inicie o PowerShell com direitos de Administrador e, em seguida, execute os dois comandos abaixo:

```
set-executionpolicy remotesigned
```

Este comando permite a execução de scripts assinados remotamente, que é o caso do Jenv. Para finalizar a instalação, execute a linha a seguir:

```
(new-object Net.WebClient).DownloadString("http://get.jenv.io/GetJenv.ps1") | iex
```

É recomendável reiniciar o PowerShell finalizada a instalação. Tal como no caso do Linux, para verificar se o procedimento foi executado com sucesso, execute o comando `jenv help`.

É importante mencionar que a implementação do Jenv para Windows é diferente da usada no Linux. Sendo assim algumas inconsistências entre as duas versões podem aparecer.

Conceito importante: candidatos (Candidates)

No jargão do Jenv, um **candidate** representa um software que é executado na JVM. Uma das principais vantagens do Jenv é a imensa quantidade de candidatos suportados pelo sistema. Para acessar a lista completa, execute o comando abaixo:

```
jenv candidates
```

Se executou o comando acima, viu que a lista é imensa! Se quiser, existe uma versão online desta lista neste link: <http://jenv.io/candidates>.

Instalando um candidato

A instalação de qualquer software pelo Jenv é bastante simples, e consiste no fluxo abaixo:

- Liste todas as versões disponíveis para o candidato
- Mandar instalar a versão desejada

O que pode ser refletido em dois comandos: o primeiro é o `jenv list [nome do candidato]`. A listagem a seguir expõe um exemplo ao listarmos todas as versões do Java que podem ser instaladas pelo Jenv:

```
jenv list java
Available java Versions
=====
>* 1.8.0_71
   1.8.0_60
   1.8.0_51
   1.8.0_11
   1.8.0_05
   1.8.0
* 1.7.0_75
  1.7.0_67
  1.7.0_40
  1.7.0_25
* 1.6.0_45
```

O asterisco indica as versões instaladas no sistema. O caractere de **maior que** (>) referencia a versão usada por padrão pelo usuário (veremos mais sobre isto adiante).

Imagine que você deseja instalar a versão 1.8.0_60 do JDK. Como proceder? Basta usar o comando `jenv install [candidato] [versão]`, tal como no exemplo a seguir:

```
jenv install java 1.8.0_60
```

O comando irá buscar pela versão pedida em um repositório central do Jenv. Uma vez encontrada, é iniciado o processo de download e subsequente instalação do mesmo.

Escolhendo a versão padrão de um candidato

Muitas vezes temos mais de uma versão de um candidato instalado, mas na maior parte das vezes usamos apenas uma. Nestes casos é interessante definirmos qual a instalação padrão com a qual queremos trabalhar. Isto é feito com o comando `jenv default [candidato] [versão]`.

Imagine que o Java com o qual você mais trabalhe seja o 1.6.0_45. Basta executar o comando abaixo:

```
jenv default java 1.6.0_45
```

Importante mencionar que sempre que é finalizada a instalação de um candidato pelo Jenv, ele irá lhe perguntar se esta será a versão padrão.

E como faço para desinstalar?

Você usa o comando `jenv uninstall [candidate] [versão]`.

Exemplo: para desinstalar o Gradle 2.7 execute o comando a seguir.

```
jenv uninstall gradle 2.7
```

Usando uma versão diferente do candidato

Tal como dito no início deste guia, basta usar o comando `jenv use [candidato] [versão]`. Para usar o JDK 1.8.0_75, portanto, bastaria executar o comando a seguir:

```
jenv use java 1.8.0_75
```

Lidando com repositórios

O Jenv armazena localmente a lista de todas as versões dos candidatos disponíveis em seu repositório central. Muitas vezes novas versões são lançadas, incluídas neste repositório e não aparecem na sua listagem local.

Este problema pode ser facilmente resolvido executando-se o comando abaixo:

```
jenv repo update
```

O índice do seu repositório local estará atualizado e agora aquela nova versão estará disponível para você.

Atualizando o próprio Jenv

Muitas vezes é interessante atualizar o próprio Jenv. Para tal, basta executar o comando abaixo:

```
jenv selfupdate
```

Caso uma nova versão do Jenv exista, esta será instalada em seu computador.

Hackeando o Jenv

Todas as configurações do Jenv encontram-se armazenadas no diretório `${user.home}/.jenv` no Linux ou, no caso do Windows, `${user.home}/jenv` ou `C:\JENV`. Chamaremos este diretório de `JENV_HOME`.

Nesta pasta se encontram todos os candidates que você mandou instalar usando o comando `jenv install`. Nesta seção você aprenderá como tirar proveito deste diretório em duas situações bastante comuns que são as seguintes:

- A versão que preciso do candidate não se encontra disponível no repositório do Jenv
- Já possuo a versão do candidate instalada em meu computador e não quero baixá-la novamente, mas gostaria que o Jenv a gerenciasse para mim.

Na pasta `JENV_HOME` há um subdiretório chamado `candidates`. Nele é que se encontram instalados os candidates que você precisa. No meu computador existem quatro subdiretórios em seu interior:

- ant
- gradle
- grails
- java

Se você precisar, por exemplo, usar a versão 1.1 do Grails, que não se encontra disponível no repositório central do Jenv, o que você faz? Simples: crie um link simbólico para a pasta na qual se encontra sua instalação deste candidato dentro da pasta `JENV_HOME/candidates/grails`.

Antes de fazer isto, verifique o conteúdo da pasta `grails` (ou do seu candidate alvo). Você notará que em seu interior há algumas pastas nomeadas de acordo com o número da versão do candidato, tal como no exemplo abaixo em meu computador:

- 1.3.7
- 1.3.9
- 2.5.3
- 3.0.9
- 3.0.14

Criando um link simbólico no Linux

Use o comando `ln` com a sintaxe abaixo:

```
ln -s [diretório de origem] [nome do Link]
```

Usando o exemplo acima, poderia ser executado o seguinte comando:

```
ln -s /home/kico/grails/1.1 1.1
```

Criando um link simbólico no Windows

No caso do Windows, é necessário executar o prompt de comando no modo administrador e, em seguida, usar o comando **mklink** com a sintaxe abaixo:

```
mklink /D [diretório de origem] [arquivo de link a ser criado]
```

Usando o mesmo exemplo, poderia ser executado o seguinte comando:

```
mklink /D c:\grails\1.1 c:\jenv\candidates\grails\1.1
```

Link simbólico ou cópia?

A vantagem do link simbólico é a economia de espaço em seu disco, pois você não irá duplicar informações em seu interior, mas se preferir simplesmente copiar a versão do seu candidate para a pasta **JENV_HOME/candidates/[nome do candidate]/[versão]**, sintá-se à vontade. Funciona também.

Jenv ou SDKMan?

Uma alternativa interessante ao Jenv é o SDKMan (<http://sdkman.io>), que é muito popular entre desenvolvedores Groovy e Grails.

Na itexto (<http://www.itexto.com.br>) optamos por usar mais o Jenv pelas seguintes razões:

- Possui um número muito maior de candidatos
- Até a escrita deste guia, nos permite trocar a versão do JDK, o que é fundamental em nosso dia a dia
- Em nossa experiência se mostrou bem mais fácil de ser instalado em ambientes Windows