nuts, the Java Package Manager

https://github.com/thevpc/nuts (git repo)

https://thevpc.github.io/nuts (website)

thevpc, 2021-12-18

Plan

1. Why a package manager
2. `nuts` features
3. Demo

1. Why a Package Manager

- Popularity of a language is proportional to popularity of its PM
  - Javascript: npm/npx/yarn
  - Python: pip, conda
  - Ruby: rubygems
- Newcomer languages already include a PM
  - golang package manager (modules)
- Java ecosystem already have more that 7M packages deployed

1.1. Java Package Manager?

- `maven`, `gradle`
  - Build tools
  - Dependency-management tools
  - Poor package/deployment management (`maven` 's `deploy` is a build time stage)
  - Lack of deployment lifecycle (install/uninstall/update)

1.2. Example

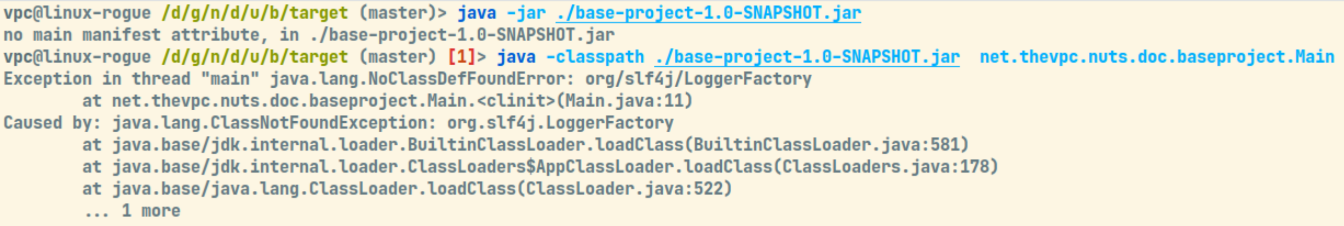
```java
package net.thevpc.nuts.doc.baseproject;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

public class Main {
private static final Logger LOG = LoggerFactory.getLogger(Main.class);
    public static void main(String[] args) {
        LOG.debug("A simple app with dependencies. Won't work out of the box!,
unless...");
    }
}
```

1.3. pom.xml

```xml
<?xml version="1.0" encoding="UTF-8"?><project xmlns="..."><modelVersion>
4.0.0</modelVersion>
<groupId>net.thevpc.nuts.doc</groupId><artifactId>base-project</artifactId>
<version>1.0-SNAPSHOT</version>
<dependencies><dependency>
        <groupId>org.apache.logging.log4j</groupId>
        <artifactId>log4j-api</artifactId><version>2.7</version>
    </dependency>
    <dependency>
        <groupId>org.apache.logging.log4j</groupId>
        <artifactId>log4j-core</artifactId><version>2.7</version>
    </dependency>
    <dependency>
        <groupId>org.apache.logging.log4j</groupId>
        <artifactId>log4j-slf4j-impl</artifactId><version>2.7</version>
    </dependency></dependencies>
</project>
```

1.4. Example

- With a minimal `pom.xml` we cannot execute unless we add the `dependencies` to the classpath
- We also need to adjust the pom.xml to include the main class too!



```
vpc@linux-rogue /d/g/n/d/u/b/target (master)> java -jar ./base-project-1.0-SNAPSHOT.jar
no main manifest attribute, in ./base-project-1.0-SNAPSHOT.jar
vpc@linux-rogue /d/g/n/d/u/b/target (master) [1]> java -classpath ./base-project-1.0-SNAPSHOT.jar  net.thevpc.nuts.doc.baseproject.Main
Exception in thread "main" java.lang.NoClassDefFoundError: org/slf4j/LoggerFactory
        at net.thevpc.nuts.doc.baseproject.Main.<clinit>(Main.java:11)
Caused by: java.lang.ClassNotFoundException: org.slf4j.LoggerFactory
        at java.base/jdk.internal.loader.BuiltinClassLoader.loadClass(BuiltinClassLoader.java:581)
        at java.base/jdk.internal.loader.ClassLoaders$AppClassLoader.loadClass(ClassLoaders.java:178)
        at java.base/java.lang.ClassLoader.loadClass(ClassLoader.java:522)
        ... 1 more
```

1.5. Alternatives for deployment

- Java Web Start
- System PM / Installers
- Portable Installers
- Custom Deployments
- Build time Processors (Fat Jars)

1.6. Java Web start

- Run Remote App using `jnlp` file (with all of it dependencies)
- Special packaging
- Execution Sandbox
- Deprecated!! since Java9
- No Shared Dependencies / Centralize Dep Mgt
- What about `trivrost`, `OpenJNLP` ?

1.7. System PM / Installers

- `rpm`, `deb`, `dmg`, `msi`
  - Native integration with OS/Env
  - Centralized management
  - Automatable (cmdline)
  - Not portable
  - Multiple deployment packages
  - Problem with installing multiple versions of the same package

1.8. Portable Installers

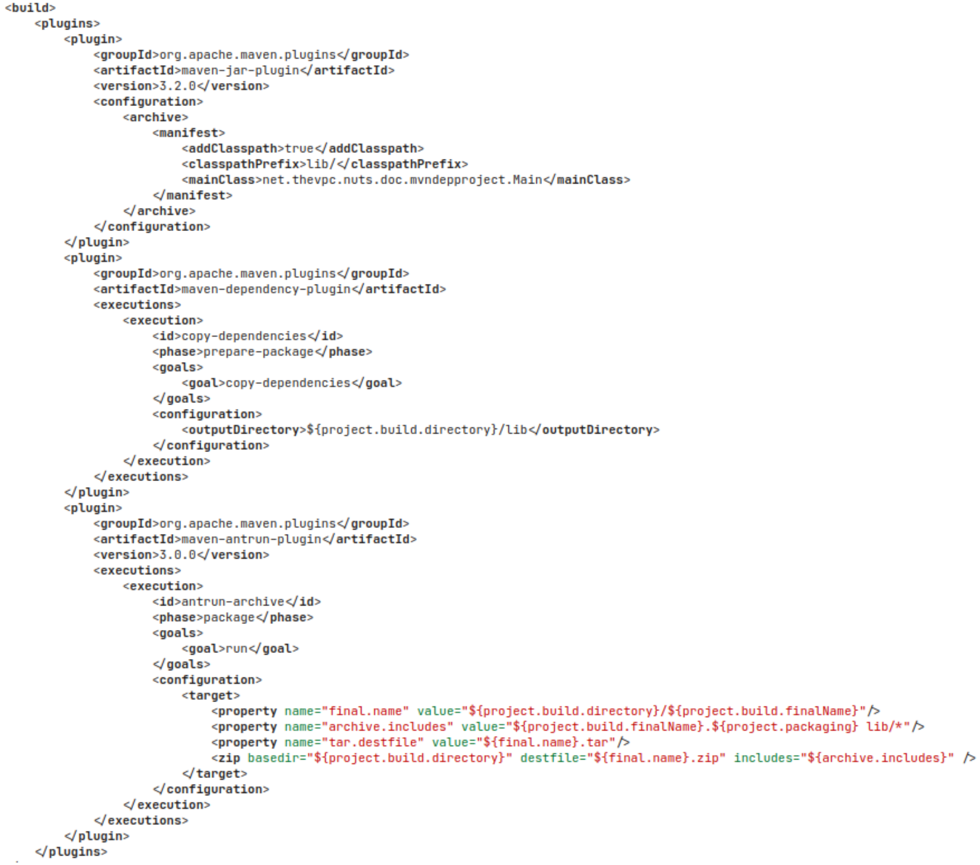
- `InstallAnywhere`, `GetDown`, `IzPack`, BitRock `InstallBuilder`
  - Good integration with OS/Env
  - No centralized management
  - Disk and network overload of dependencies
  - Graphical! not suitable for automation
  - Still Manual

1.9. Custom Deployers

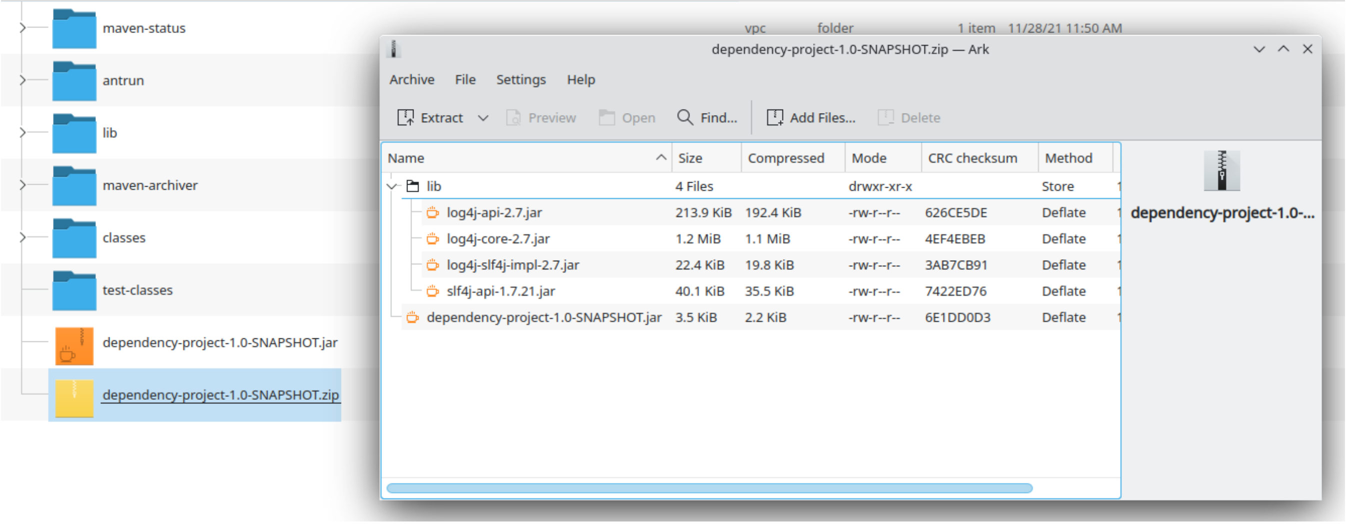
- Custom (tomcat, netbeans) with multiple formats (tarball, zip)
  - Manual
  - No centralized management
  - Difficult to automate
  - Lack of integration with environment
  - Disk and network overload of dependencies

1.10. Fat Packages: maven-dependency-plugin

- `maven-dependency-plugin`
  - Maven plugin
  - Jars included in the "lib" folder
  - Still need to bundle the jar and the lib folder (zip with `maven-antrun-plugin`)



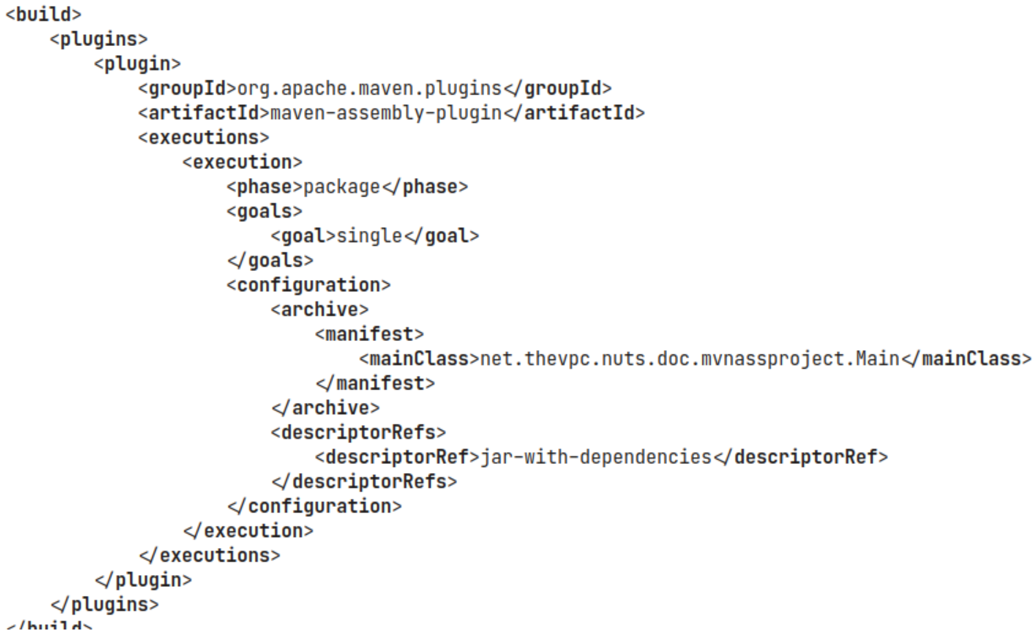
```
<build>
    <plugins>
        <plugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-jar-plugin</artifactId>
            <version>3.2.0</version>
            <configuration>
                <archive>
                    <manifest>
                        <addClasspath>true</addClasspath>
                        <classpathPrefix>lib/</classpathPrefix>
                        <mainClass>net.thevpc.nuts.doc.mvndepproject.Main</mainClass>
                    </manifest>
                </archive>
            </configuration>
        </plugin>
        <plugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-dependency-plugin</artifactId>
            <executions>
                <execution>
                    <id>copy-dependencies</id>
                    <phase>prepare-package</phase>
                    <goals>
                        <goal>copy-dependencies</goal>
                    </goals>
                    <configuration>
                        <outputDirectory>${project.build.directory}/lib</outputDirectory>
                    </configuration>
                </execution>
            </executions>
        </plugin>
        <plugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-antrun-plugin</artifactId>
            <version>3.0.0</version>
            <executions>
                <execution>
                    <id>antrun-archive</id>
                    <phase>package</phase>
                    <goals>
                        <goal>run</goal>
                    </goals>
                    <configuration>
                        <target>
                            <property name="final.name" value="${project.build.directory}/${project.build.finalName}"/>
                            <property name="archive.includes" value="${project.build.finalName}.${project.packaging} lib/*"/>
                            <property name="tar.destfile" value="${final.name}.tar"/>
                            <zip basedir="${project.build.directory}" destfile="${final.name}.zip" includes="${archive.includes}" />
                        </target>
                    </configuration>
                </execution>
            </executions>
        </plugin>
    </plugins>
```



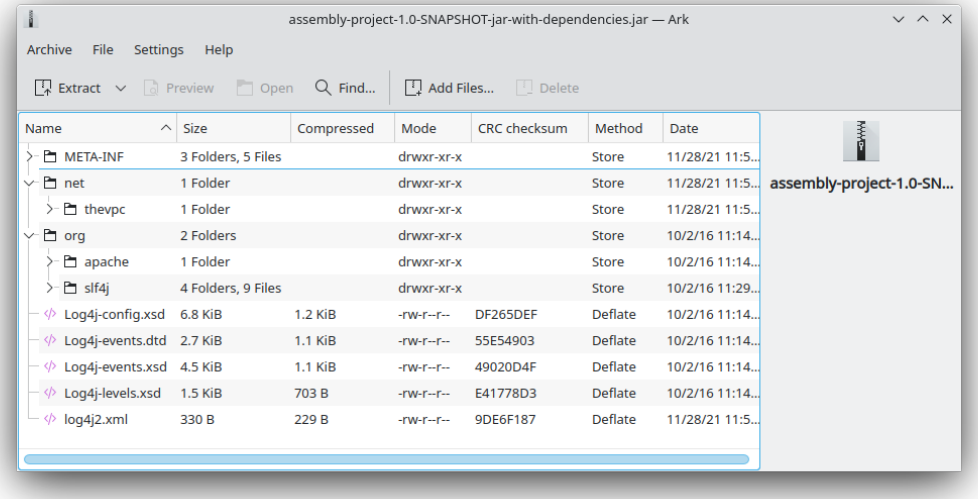
Name	Size	Compressed	Mode	CRC checksum	Method
lib	4 Files		drwxr-xr-x		Store
log4j-api-2.7.jar	213.9 KiB	192.4 KiB	-rw-r--r--	626CE5DE	Deflate
log4j-core-2.7.jar	1.2 MiB	1.1 MiB	-rw-r--r--	4EF4EBEB	Deflate
log4j-slf4j-impl-2.7.jar	22.4 KiB	19.8 KiB	-rw-r--r--	3AB7CB91	Deflate
slf4j-api-1.7.21.jar	40.1 KiB	35.5 KiB	-rw-r--r--	7422ED76	Deflate
dependency-project-1.0-SNAPSHOT.jar	3.5 KiB	2.2 KiB	-rw-r--r--	6E1DD0D3	Deflate

1.11. Fat Jars : Uber Jar

- `maven-assembly-plugin`
  - Jars deflated into the same jar
  - Can rewrite classes/resources
- `maven-shade-plugin`
  - Jars deflated into thesame jar
  - Rewrites classes/resources
  - Simpler than `maven-assembly-plugin`



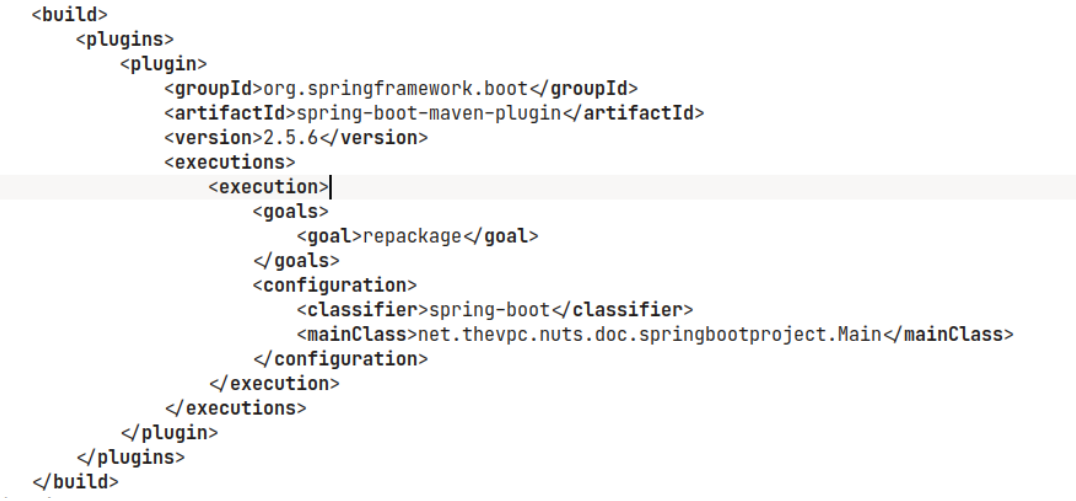
```
<build>
    <plugins>
        <plugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-assembly-plugin</artifactId>
            <executions>
                <execution>
                    <phase>package</phase>
                    <goals>
                        <goal>single</goal>
                    </goals>
                    <configuration>
                        <archive>
                            <manifest>
                                <mainClass>net.thevpc.nuts.doc.mvnassproject.Main</mainClass>
                            </manifest>
                        </archive>
                        <descriptorRefs>
                            <descriptorRef>jar-with-dependencies</descriptorRef>
                        </descriptorRefs>
                    </configuration>
                </execution>
            </executions>
        </plugin>
    </plugins>
</build>
```


assembly-project-1.0-SNAPSHOT-jar-with-dependencies.jar — Ark
Archive
File
Settings
Help
Extract
Preview
Open
Find...
Add Files...
Delete
assembly-project-1.0-SN...

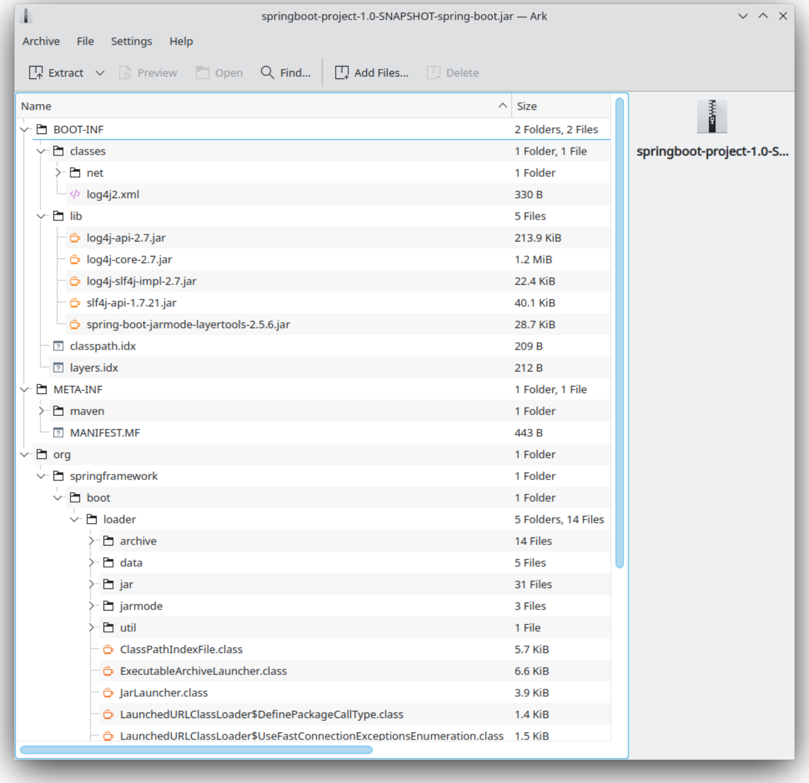
Name	Size	Compressed	Mode	CRC checksum	Method	Date
META-INF	3 Folders, 5 Files		drwxr-xr-x		Store	11/28/21 11:5...
net	1 Folder		drwxr-xr-x		Store	11/28/21 11:5...
thevpc	1 Folder		drwxr-xr-x		Store	11/28/21 11:5...
org	2 Folders		drwxr-xr-x		Store	10/2/16 11:14...
apache	1 Folder		drwxr-xr-x		Store	10/2/16 11:14...
slf4j	4 Folders, 9 Files		drwxr-xr-x		Store	10/2/16 11:29...
Log4j-config.xsd	6.8 KiB	1.2 KiB	-rw-r--r--	DF265DEF	Deflate	10/2/16 11:14...
Log4j-events.dtd	2.7 KiB	1.1 KiB	-rw-r--r--	55E54903	Deflate	10/2/16 11:14...
Log4j-events.xsd	4.5 KiB	1.1 KiB	-rw-r--r--	49020D4F	Deflate	10/2/16 11:14...
Log4j-levels.xsd	1.5 KiB	703 B	-rw-r--r--	E41778D3	Deflate	10/2/16 11:14...
log4j2.xml	330 B	229 B	-rw-r--r--	9DE6F187	Deflate	11/28/21 11:5...

1.12. Fat Jars : Jar Jar

- `onejar-maven-plugin`
  - Rewrites jar to include dependencies as jars!
  - Adds bootstrap classes
  - Changes classloader
- `spring-boot-maven-plugin`
  - Rewrites jar to include dependencies as jars!
  - Adds bootstrap classes
  - Changes classloader



```
<build>
    <plugins>
        <plugin>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-maven-plugin</artifactId>
            <version>2.5.6</version>
            <executions>
                <execution>
                    <goals>
                        <goal>repackage</goal>
                    </goals>
                    <configuration>
                        <classifier>spring-boot</classifier>
                        <mainClass>net.thevpc.nuts.doc.springbootproject.Main</mainClass>
                    </configuration>
                </execution>
            </executions>
        </plugin>
    </plugins>
</build>
```


springboot-project-1.0-SNAPSHOT-spring-boot.jar — Ark
Archive
File
Settings
Help
Extract
Preview
Open
Find...
Add Files...
Delete
springboot-project-1.0-S...

```
Name                                                              Size
BOOT-INF                                                          2 Folders, 2 Files
  classes                                                         1 Folder, 1 File
    net                                                           1 Folder
    log4j2.xml                                                    330 B
  lib                                                             5 Files
    log4j-api-2.7.jar                                             213.9 KiB
    log4j-core-2.7.jar                                            1.2 MiB
    log4j-slf4j-impl-2.7.jar                                      22.4 KiB
    slf4j-api-1.7.21.jar                                          40.1 KiB
    spring-boot-jarmode-layertools-2.5.6.jar                      28.7 KiB
  classpath.idx                                                   209 B
  layers.idx                                                      212 B
META-INF                                                          1 Folder, 1 File
  maven                                                           1 Folder
  MANIFEST.MF                                                     443 B
org                                                               1 Folder
  springframework                                                 1 Folder
    boot                                                          1 Folder
      loader                                                      5 Folders, 14 Files
        archive                                                   14 Files
        data                                                      5 Files
        jar                                                       31 Files
        jarmode                                                   3 Files
        util                                                      1 File
        ClassPathIndexFile.class                                  5.7 KiB
        ExecutableArchiveLauncher.class                           6.6 KiB
        JarLauncher.class                                         3.9 KiB
        LaunchedURLClassLoader$DefinePackageCallType.class        1.4 KiB
        LaunchedURLClassLoader$UseFastConnectionExceptionsEnumeration.class  1.5 KiB
```

1.13. So…

- All alternatives are poor and/or ugly
- `pom.xml` polluted with +16-20 lines of code
- ~~Why do we need a package manager for `Java`~~
- Why don’t we already have a package manager for `Java`!

2. nuts Package Manager for Java

Main Idea:

- Little to no Intrusion and Backward compatibility to support existing apps and repos
- Good Integration with Java ecosystem and popular build/deploy/devops tools
- Solid enough to support multiple platforms
- Simple but extensible
- Open Source

2.1. nuts: A Package Manager for Java

- Centralized package manager for Java Apps and Libs (not only)
  - `install`, `uninstall`, `update`, `search` and `exec` for packages
  - Optimized dependency resolution solver
  - Cache for dependencies across installed apps
- Automation/devops friendly commandline tool
- Portable across Architectures, OSes, OS Distibs, Desktop Environments, Platforms (Java versions)
- Libre and Open Source, developed in java

2.2. nuts: A Package Manager for Java

Is Not:

- a replacement for `maven`, `gradle` or any build tool (used at deploy time)
- a plugin for `maven`, `gradle` or any build tool (do not change the build process)
- a replacement for `spring` framework or any other framework
- a replacement for `IzPack` or `InstallAnywhere` (but can do pretty much of it)
- a replacement for `ansible` or `chef` (but is conceptually driven by automation)
- a mere download tool

2.3. nuts: Maven & Gradle

- Integrates seamlessly with `maven`
  - No required modification of the build process
  - Does not alter/rewrite the package
  - No special `maven`/`gradle` plugin needed
- Supports local Jars, public packages (maven central), and private packages (local .m2, nexus repos,...)
- Solves at runtime what `maven`/`gradle` solve at build time
  - Supports `maven` and `gradle` dependency resolution algorithms, scopes, ...

2.4. nuts: Dependency Optimization

- Downloads, Caches and Installs only relevant dependencies according to
  - `arch` (hardware architecture: x86, x64, relevant for native dependencies)
  - `os` (operating system: Win/Linux/Mac, relevant for specific tasks)
  - `osDist` (operating distribution : Ubuntu/OpenSuse,...)
  - `desktop` (desktop environment, relevant for icon/shortcut creation and environment integration)
  - `platform` (java SE versions installed to know what dependencies to use)

2.5. nuts: Integration

- Solid integration with environments
  - Uses OS’s File System Layouts (XDG for Linux, …)
    - separate folders per app
    - separate folders for log, config, lib, cache, etc.
    - portable across OSes (~/.config versus ~/AppData)
  - Supports cmdline and gui apps (installs scripts, icons, menus, …)
  - Supports `jar` and `zip` based apps

2.6. nuts: Toolbox

- Terminal Coloring on Linux/Windows
- Supports Windows `cmd`/`PowerShell` and *NIX `sh`, `bash`, `csh`, `zsh` and `fish` and their relative `rcfiles`
- Bundles a `bash`/`GNU binutils` compatible (still incomplete) but enhanced java implementations
  - ls, cp, touch, mkdir, rmdir, ...
  - works on windows
  - adds some extra goodies (ssh, json, support ...)

2.7. nuts: Existing Apps

- Supports out of the box
  - `maven` 's repos (including central, spring, google, …), more than 7M dependencies
  - Apache repos (`netbeans`, `tomcat`, `derby`, etc…)

2.8. nuts: Automation

- Powerful toolbox with customizable output formats
  - `props`
  - `xml`
  - `json`
  - `yaml`
  - `table`
  - `tree`

2.9. nuts: Unique features

- Is statically built and has (almost) no dependencies
- Can be used as a library to support transitive classPath resolver
- Has a clean and rich API

2.10. nuts: Stability

- Tested:
  - over 140 regression tests with 3500+ lines of test-code in the repository.
  - `opensuse`, `ubuntu`, `docker`, `windows7`, `windows10`
  - `sh`, `bash`, `csh`, `zsh`, `fish`

2.11. 'nuts'… really?

- `N` etwork `U` pdatable `T` hings `S` ervices
- The `nuts` (fool) companion for the `maven` (sage)

3. Demonstration

3.1. Install Nuts

1. Download `nuts.jar`
2. run `java -jar nuts.jar -Zy`
3. restart your terminal

3.2. Install Nuts (Linux)

- Install for Preview/Evaluation, most `recent`

```
$ wget https://thevpc.net/nuts.jar -o nuts.jar
$ java -jar nuts.jar -Zy -r=+dev
$ exit
```

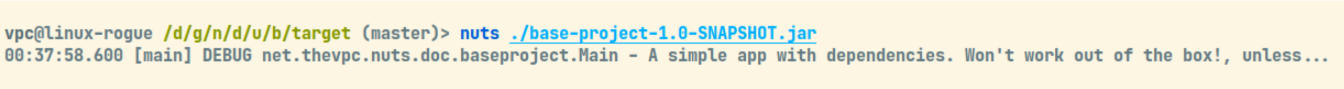
- Install for Production, most `stable`

```
$ wget https://repo.maven.apache.org/maven2/net/thevpc/nuts/nuts/0.8.3/nuts-0.8.3.jar
-O nuts.jar
$ java -jar nuts.jar -Zy
$ exit
```

- In all cases, do not forget to `restart` your terminal

3.3. Run the app

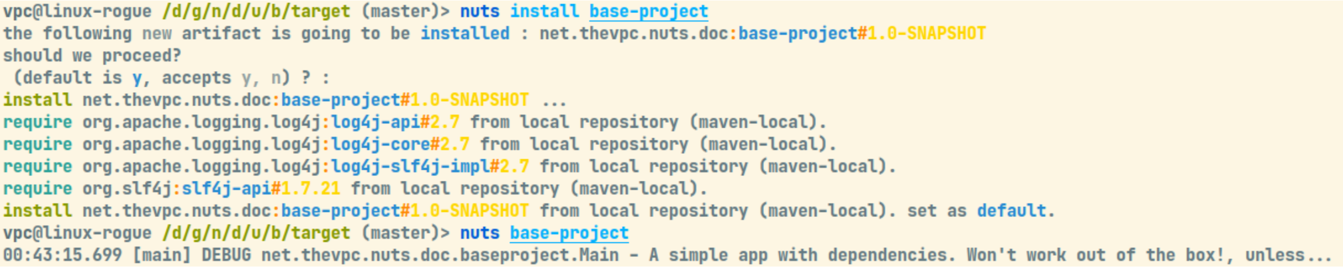
- We just run the app, with no modification
- We use the built (by maven) jar



```
vpc@linux-rogue /d/g/n/d/u/b/target (master)> nuts ./base-project-1.0-SNAPSHOT.jar
00:37:58.600 [main] DEBUG net.thevpc.nuts.doc.baseproject.Main - A simple app with dependencies. Won't work out of the box!, unless...
```

3.4. Demonstration : Install Application

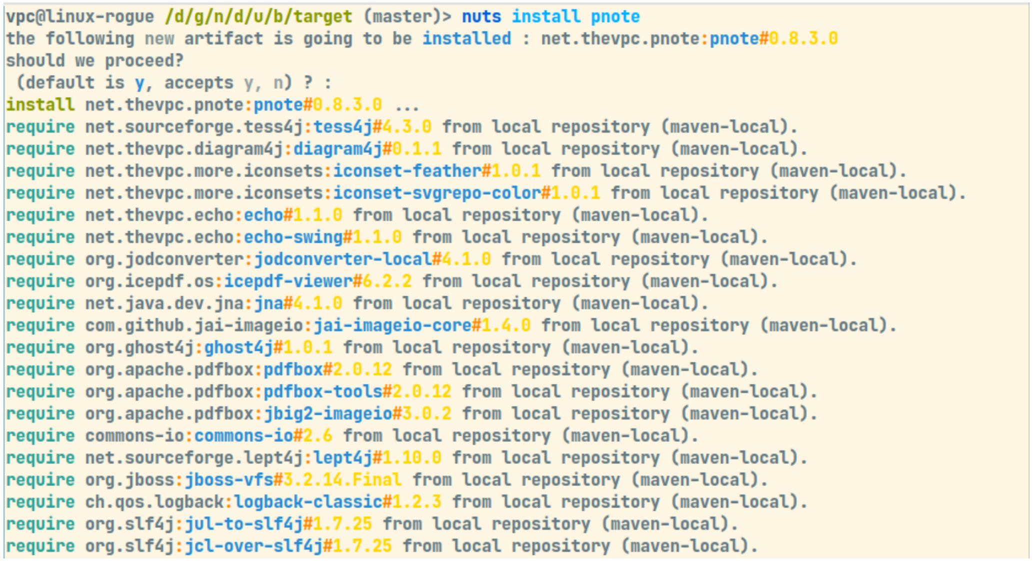
- Or we can install the app (and its `required` dependencies)
- And then we run it



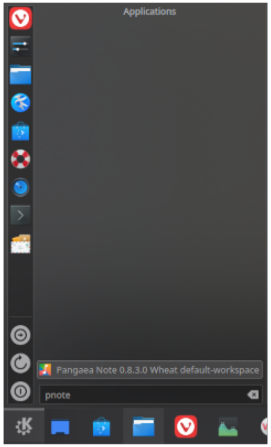
```
vpc@linux-rogue /d/g/n/d/u/b/target (master)> nuts install base-project
the following new artifact is going to be installed : net.thevpc.nuts.doc:base-project#1.0-SNAPSHOT
should we proceed?
 (default is y, accepts y, n) ? :
install net.thevpc.nuts.doc:base-project#1.0-SNAPSHOT ...
require org.apache.logging.log4j:log4j-api#2.7 from local repository (maven-local).
require org.apache.logging.log4j:log4j-core#2.7 from local repository (maven-local).
require org.apache.logging.log4j:log4j-slf4j-impl#2.7 from local repository (maven-local).
require org.slf4j:slf4j-api#1.7.21 from local repository (maven-local).
install net.thevpc.nuts.doc:base-project#1.0-SNAPSHOT from local repository (maven-local). set as default.
vpc@linux-rogue /d/g/n/d/u/b/target (master)> nuts base-project
00:43:15.699 [main] DEBUG net.thevpc.nuts.doc.baseproject.Main - A simple app with dependencies. Won't work out of the box!, unless...
```

3.5. Install Gui App

- We can run a gui app of course, and create a shortcut for it:

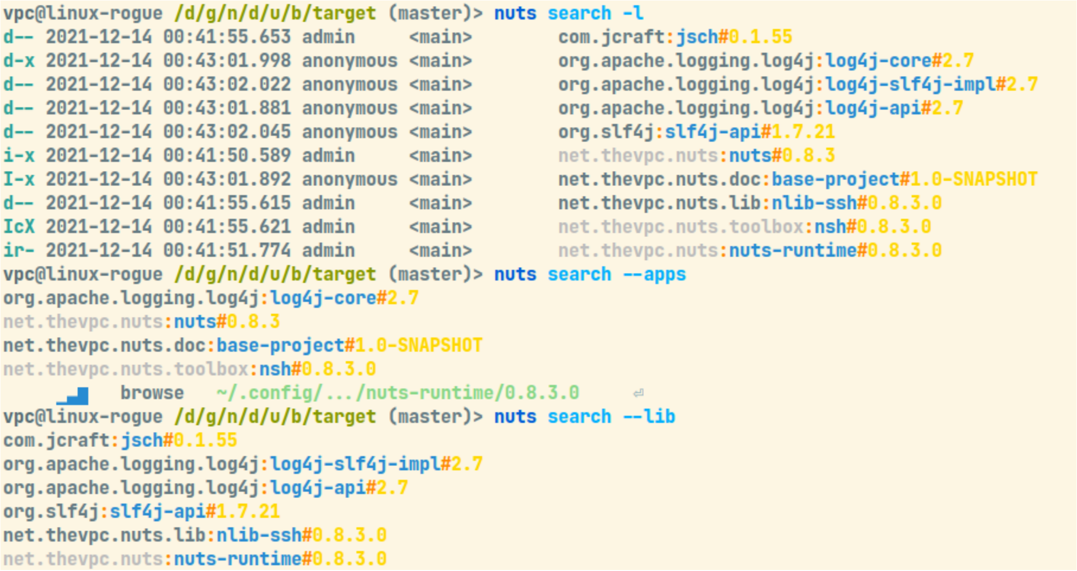


```
vpc@linux-rogue /d/g/n/d/u/b/target (master)> nuts install pnote
the following new artifact is going to be installed : net.thevpc.pnote:pnote#0.8.3.0
should we proceed?
 (default is y, accepts y, n) ? :
install net.thevpc.pnote:pnote#0.8.3.0 ...
require net.sourceforge.tess4j:tess4j#4.3.0 from local repository (maven-local).
require net.thevpc.diagram4j:diagram4j#0.1.1 from local repository (maven-local).
require net.thevpc.more.iconsets:iconset-feather#1.0.1 from local repository (maven-local).
require net.thevpc.more.iconsets:iconset-svgrepo-color#1.0.1 from local repository (maven-local).
require net.thevpc.echo:echo#1.1.0 from local repository (maven-local).
require net.thevpc.echo:echo-swing#1.1.0 from local repository (maven-local).
require org.jodconverter:jodconverter-local#4.1.0 from local repository (maven-local).
require org.icepdf.os:icepdf-viewer#6.2.2 from local repository (maven-local).
require net.java.dev.jna:jna#4.1.0 from local repository (maven-local).
require com.github.jai-imageio:jai-imageio-core#1.4.0 from local repository (maven-local).
require org.ghost4j:ghost4j#1.0.1 from local repository (maven-local).
require org.apache.pdfbox:pdfbox#2.0.12 from local repository (maven-local).
require org.apache.pdfbox:pdfbox-tools#2.0.12 from local repository (maven-local).
require org.apache.pdfbox:jbig2-imageio#3.0.2 from local repository (maven-local).
require commons-io:commons-io#2.6 from local repository (maven-local).
require net.sourceforge.lept4j:lept4j#1.10.0 from local repository (maven-local).
require org.jboss:jboss-vfs#3.2.14.Final from local repository (maven-local).
require ch.qos.logback:logback-classic#1.2.3 from local repository (maven-local).
require org.slf4j:jul-to-slf4j#1.7.25 from local repository (maven-local).
require org.slf4j:jcl-over-slf4j#1.7.25 from local repository (maven-local).
```



3.6. Search for available applications

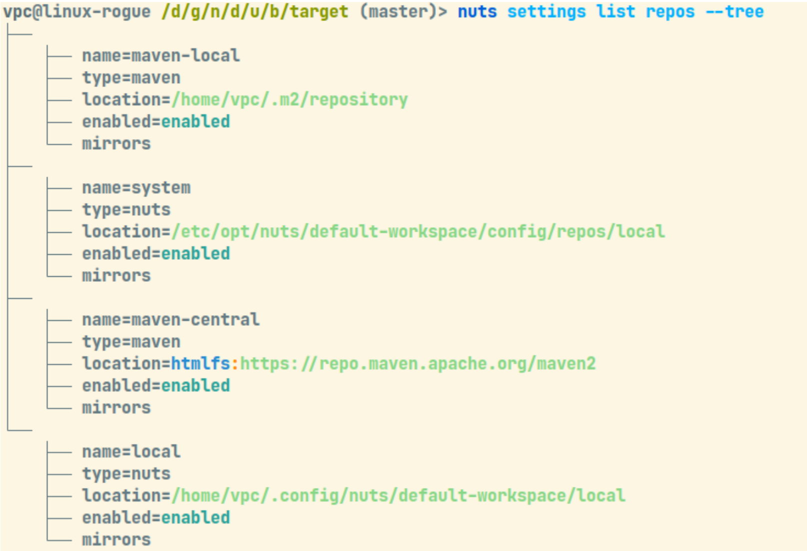
- We can search for installed or available (local/remote) apps
- We can search for apps and/or libs



```
vpc@linux-rogue /d/g/n/d/u/b/target (master)> nuts search -l
d-- 2021-12-14 00:41:55.653 admin     <main>      com.jcraft:jsch#0.1.55
d-x 2021-12-14 00:43:01.998 anonymous <main>      org.apache.logging.log4j:log4j-core#2.7
d-- 2021-12-14 00:43:02.022 anonymous <main>      org.apache.logging.log4j:log4j-slf4j-impl#2.7
d-- 2021-12-14 00:43:01.881 anonymous <main>      org.apache.logging.log4j:log4j-api#2.7
d-- 2021-12-14 00:43:02.045 anonymous <main>      org.slf4j:slf4j-api#1.7.21
i-x 2021-12-14 00:41:50.589 admin     <main>      net.thevpc.nuts:nuts#0.8.3
I-x 2021-12-14 00:43:01.892 anonymous <main>      net.thevpc.nuts.doc:base-project#1.0-SNAPSHOT
d-- 2021-12-14 00:41:55.615 admin     <main>      net.thevpc.nuts.lib:nlib-ssh#0.8.3.0
IcX 2021-12-14 00:41:55.621 admin     <main>      net.thevpc.nuts.toolbox:nsh#0.8.3.0
ir- 2021-12-14 00:41:51.774 admin     <main>      net.thevpc.nuts:nuts-runtime#0.8.3.0
vpc@linux-rogue /d/g/n/d/u/b/target (master)> nuts search --apps
org.apache.logging.log4j:log4j-core#2.7
net.thevpc.nuts:nuts#0.8.3
net.thevpc.nuts.doc:base-project#1.0-SNAPSHOT
net.thevpc.nuts.toolbox:nsh#0.8.3.0
          browse   ~/.config/.../nuts-runtime/0.8.3.0     ⏎
vpc@linux-rogue /d/g/n/d/u/b/target (master)> nuts search --lib
com.jcraft:jsch#0.1.55
org.apache.logging.log4j:log4j-slf4j-impl#2.7
org.apache.logging.log4j:log4j-api#2.7
org.slf4j:slf4j-api#1.7.21
net.thevpc.nuts.lib:nlib-ssh#0.8.3.0
net.thevpc.nuts:nuts-runtime#0.8.3.0
```

3.7. Repositories

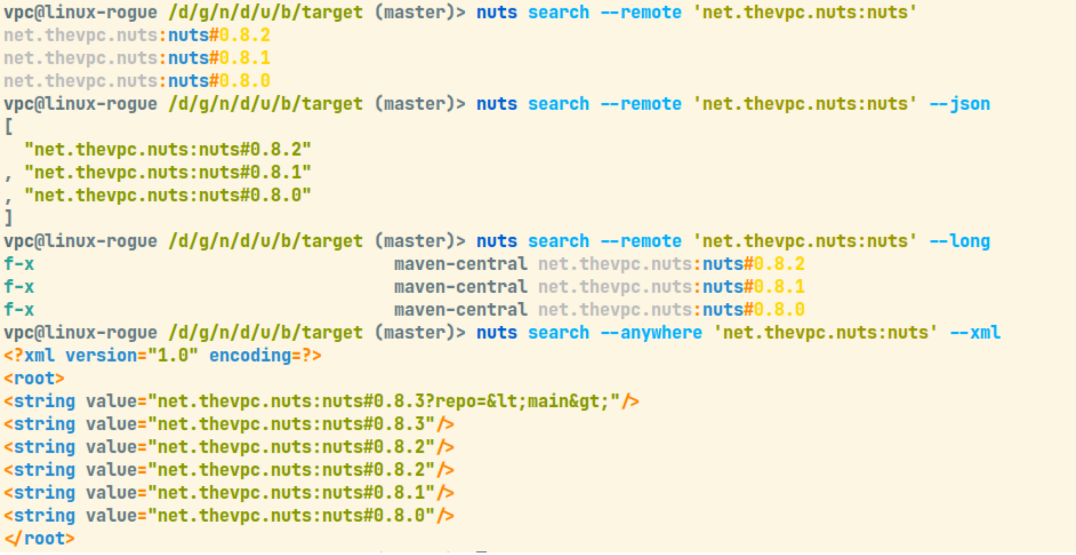
- We can configure Repositories used to install/update packages
- We can list Repositories used to install/update packages



```
vpc@linux-rogue /d/g/n/d/u/b/target (master)> nuts settings list repos --tree
├──
│   ├── name=maven-local
│   ├── type=maven
│   ├── location=/home/vpc/.m2/repository
│   ├── enabled=enabled
│   └── mirrors
├──
│   ├── name=system
│   ├── type=nuts
│   ├── location=/etc/opt/nuts/default-workspace/config/repos/local
│   ├── enabled=enabled
│   └── mirrors
├──
│   ├── name=maven-central
│   ├── type=maven
│   ├── location=htmlfs:https://repo.maven.apache.org/maven2
│   ├── enabled=enabled
│   └── mirrors
└──
    ├── name=local
    ├── type=nuts
    ├── location=/home/vpc/.config/nuts/default-workspace/local
    ├── enabled=enabled
    └── mirrors
```

3.8. Integration and Formats

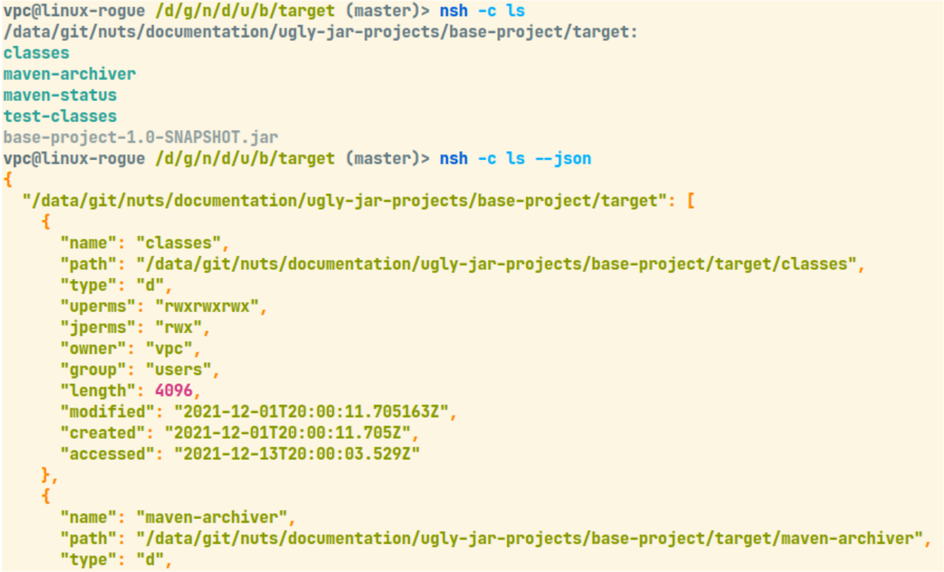
- using `--json`, `--xml`, `--props`, `--yaml` , `--table` , `--tree` for any command will customize output



```
vpc@linux-rogue /d/g/n/d/u/b/target (master)> nuts search --remote 'net.thevpc.nuts:nuts'
net.thevpc.nuts:nuts#0.8.2
net.thevpc.nuts:nuts#0.8.1
net.thevpc.nuts:nuts#0.8.0
vpc@linux-rogue /d/g/n/d/u/b/target (master)> nuts search --remote 'net.thevpc.nuts:nuts' --json
[
  "net.thevpc.nuts:nuts#0.8.2"
, "net.thevpc.nuts:nuts#0.8.1"
, "net.thevpc.nuts:nuts#0.8.0"
]
vpc@linux-rogue /d/g/n/d/u/b/target (master)> nuts search --remote 'net.thevpc.nuts:nuts' --long
f-x                                    maven-central net.thevpc.nuts:nuts#0.8.2
f-x                                    maven-central net.thevpc.nuts:nuts#0.8.1
f-x                                    maven-central net.thevpc.nuts:nuts#0.8.0
vpc@linux-rogue /d/g/n/d/u/b/target (master)> nuts search --anywhere 'net.thevpc.nuts:nuts' --xml
<?xml version="1.0" encoding=?>
<root>
<string value="net.thevpc.nuts:nuts#0.8.3?repo=&lt;main&gt;"/>
<string value="net.thevpc.nuts:nuts#0.8.3"/>
<string value="net.thevpc.nuts:nuts#0.8.2"/>
<string value="net.thevpc.nuts:nuts#0.8.2"/>
<string value="net.thevpc.nuts:nuts#0.8.1"/>
<string value="net.thevpc.nuts:nuts#0.8.0"/>
</root>
```

3.9. Companions

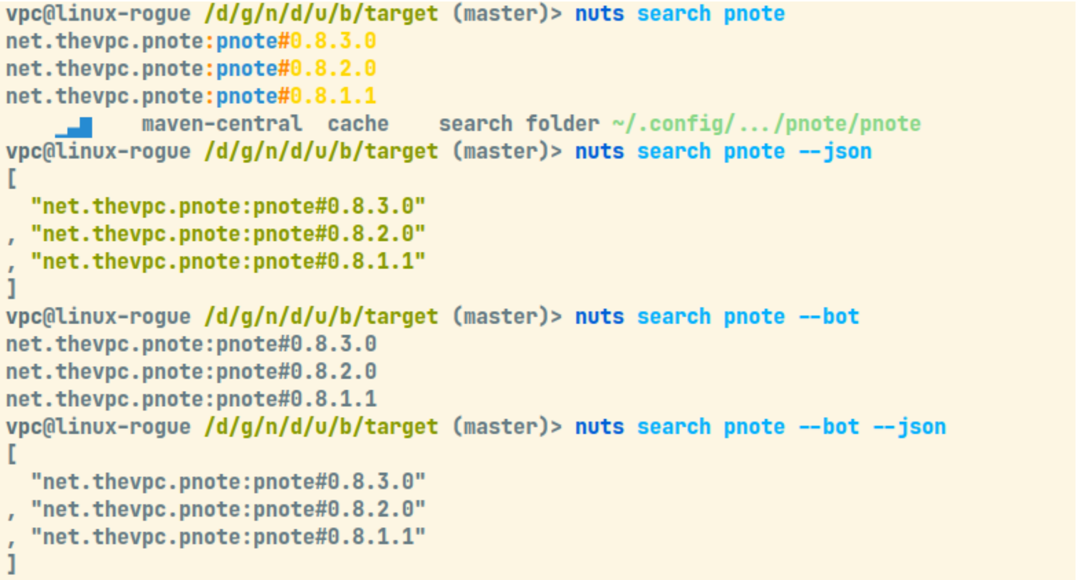
- We can use nsh instead of bash / cmd and make usage of json support out of the box



```
vpc@linux-rogue /d/g/n/d/u/b/target (master)> nsh -c ls
/data/git/nuts/documentation/ugly-jar-projects/base-project/target:
classes
maven-archiver
maven-status
test-classes
base-project-1.0-SNAPSHOT.jar
vpc@linux-rogue /d/g/n/d/u/b/target (master)> nsh -c ls --json
{
  "/data/git/nuts/documentation/ugly-jar-projects/base-project/target": [
    {
      "name": "classes",
      "path": "/data/git/nuts/documentation/ugly-jar-projects/base-project/target/classes",
      "type": "d",
      "uperms": "rwxrwxrwx",
      "jperms": "rwx",
      "owner": "vpc",
      "group": "users",
      "length": 4096,
      "modified": "2021-12-01T20:00:11.705163Z",
      "created": "2021-12-01T20:00:11.705Z",
      "accessed": "2021-12-13T20:00:03.529Z"
    },
    {
      "name": "maven-archiver",
      "path": "/data/git/nuts/documentation/ugly-jar-projects/base-project/target/maven-archiver",
      "type": "d",
```

3.10. Bot Mode

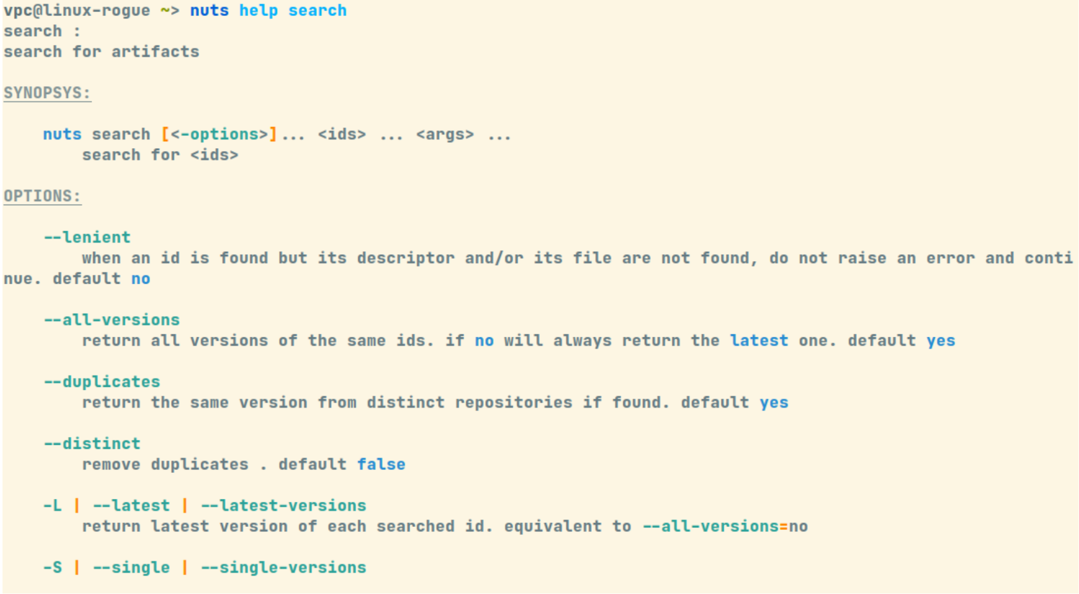
- Running with `--bot` will disable all interaction and terminal coloring



```
vpc@linux-rogue /d/g/n/d/u/b/target (master)> nuts search pnote
net.thevpc.pnote:pnote#0.8.3.0
net.thevpc.pnote:pnote#0.8.2.0
net.thevpc.pnote:pnote#0.8.1.1
           maven-central  cache    search folder ~/.config/.../pnote/pnote
vpc@linux-rogue /d/g/n/d/u/b/target (master)> nuts search pnote --json
[
  "net.thevpc.pnote:pnote#0.8.3.0"
, "net.thevpc.pnote:pnote#0.8.2.0"
, "net.thevpc.pnote:pnote#0.8.1.1"
]
vpc@linux-rogue /d/g/n/d/u/b/target (master)> nuts search pnote --bot
net.thevpc.pnote:pnote#0.8.3.0
net.thevpc.pnote:pnote#0.8.2.0
net.thevpc.pnote:pnote#0.8.1.1
vpc@linux-rogue /d/g/n/d/u/b/target (master)> nuts search pnote --bot --json
[
  "net.thevpc.pnote:pnote#0.8.3.0"
, "net.thevpc.pnote:pnote#0.8.2.0"
, "net.thevpc.pnote:pnote#0.8.1.1"
]
```

3.11. Help

- An extensive help is available from within the command line



```
vpc@linux-rogue ~> nuts help search
search :
search for artifacts

SYNOPSYS:

    nuts search [<-options>]... <ids> ... <args> ...
        search for <ids>

OPTIONS:

    --lenient
        when an id is found but its descriptor and/or its file are not found, do not raise an error and conti
nue. default no

    --all-versions
        return all versions of the same ids. if no will always return the latest one. default yes

    --duplicates
        return the same version from distinct repositories if found. default yes

    --distinct
        remove duplicates . default false

    -L | --latest | --latest-versions
        return latest version of each searched id. equivalent to --all-versions=no

    -S | --single | --single-versions
```

3.12. Conclusion

- `nuts` tries to be for `java` what `npm` is for javascript
- `nuts` is a versatile toolbox
- `nuts` is `2800+` classes, `600ko+` boot jar
- I invite you to
  - Take a shot, try to use it and give feedback
  - `Star(*)` the repository https://github.com/thevpc/nuts
  - Spread the word
  - Join the Core Team to enhance `nuts`

Thank you

https://github.com/thevpc/nuts (git repo)

https://thevpc.github.io/nuts (website)