



nuts, the Java Package Manager



~~~~~  
<https://thevpc.github.io/nuts>

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

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

[nuts.packagemanager@gmail.com](mailto:nuts.packagemanager@gmail.com)

thevpc, 2022-01-08

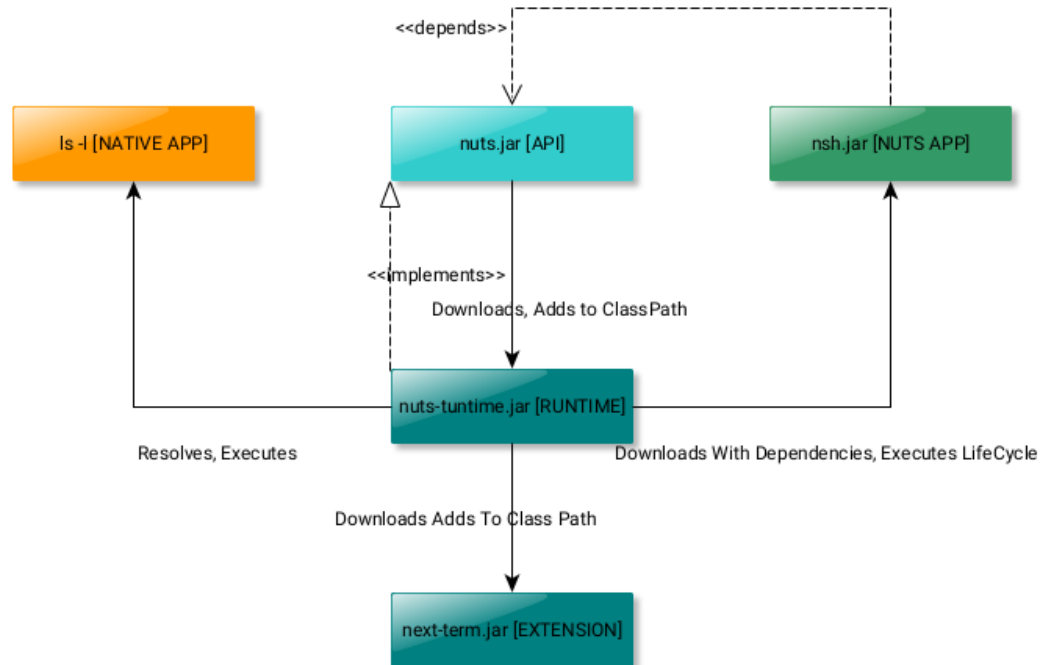


## Plan

1. API
2. Nuts as Library
3. Nuts as a Framework
4. Spring Integration



## 1. Main Components





## 2. Nuts as A Library

- Simply add nuts to your dependencies
- Compatible with java 1.8+

```
<dependency>  
  <groupId>net.thevpc.nuts</groupId>  
  <artifactId>nuts</artifactId>  
  <version>0.8.3</version>  
</dependency>
```



### 3. Nuts as A Library

- You can add runtime to force the runtime version

```
<dependency>  
  <groupId>net.thevpc.nuts</groupId>  
  <artifactId>nuts-runtime</artifactId>  
  <version>0.8.3.1</version>  
</dependency>
```



## 3.1. Session API

```
NutsSession session=Nuts.openWorkspace("-Z", "-S", "y", "--json");  
session.setConfirm(NutsConfirmationMode.ASK);  
session.setOutputFormat(NutsContentType.XML);  
  
session.out().println("Hello");  
session.out().printlnf("Hello");  
  
session.out().println(Arrays.asList("Hello"));  
session.out().printlnf("Hello");
```



## 3.2. Std In/Out/Err API

```
NutsSession session=Nuts.openWorkspace("-Z", "-S", "y", "--json");  
  
session.out().println("Hello");  
session.out().printlnf("Hello");  
  
session.out().println(Arrays.asList("Hello"));  
session.out().printlnf("Hello");  
  
session.err().....;  
session.in().....;
```



### 3.3. Find API

```
NutsSession session=...;
NutsResult<NutsId> ids=session.search()
    .addId("org.jedit:jedit")
    .setLatest(true)
    .setDistinct(true).getResultIds();
for(NutsId id:ids){
    ...
}
NutsResult<NutsDefinition> defs=session.search()
    .addId("org.jedit:jedit")
    .setLatest(true)
    .setDistinct(true).getResultDefnintions();
for(NutsDefinition d:defs){
    session.out().println(d.getInstallInformation()
        .getInstallFolder());
}
```





## 3.4. ClassPath API

```
NutsSession session=...;  
ClassLoader loader=session.search()  
    .addId("org.jedit:jedit")  
    .addId("org.springframework:spring-context")  
    .setLatest(true)  
    .setDistinct(true).getResultClassLoader();
```



## 3.5. NTF API

```
NutsSession session=...;  
session.out().println("#Hello1# ##Hello2## ##:_:Hello3## ");  
session.out().println("```java public static class MyClass {}```");  
session.out().println("```js public static class MyClass {}```");  
session.out().println("```xml <a>hello</a>```");  
session.out().println("```json {a:'hello'}```");
```



## 3.6. Format API

```
NutsSession session=...;
class Customer{String id;String name;}
Customer customer1,customer2,customer3; ...
//
session.setOutputFormat(NutsContentType.JSON).out().println(Arrays.
asList(customer1,customer2,customer3))
session.setOutputFormat(NutsContentType.TREE).out().println(Arrays.
asList(customer1,customer2,customer3))
session.setOutputFormat(NutsContentType.PLAIN).out().println(Arrays.
asList(customer1,customer2,customer3))
session.setOutputFormat(NutsContentType.XML).out().println(Arrays.
asList(customer1,customer2,customer3))
session.setOutputFormat(NutsContentType.PROPS).out().println(Arrays.
asList(customer1,customer2,customer3))
session.out().println(Arrays.asList(customer1,customer2,customer3))
```



## 3.7. Format API

```
NutsSession session=...;  
Object a,b,c,d; ...  
NutsMutableTableModel m = NutsMutableTableModel.of(session);  
m.newRow().addCells(a,b,c,d);  
session.out().printlnf(m);
```



## 3.8. Exec API

```
NutsSession session=Nuts.openWorkspace("-Z", "-S");  
int code=session.exec().addCommand("ls", "-l").getResult();  
String out=session.exec().addCommand("nsh", "ls", "--table")  
    .grabOutputString()  
    .getOutputString();
```



## 3.9. IO API

```
NutsSession session=...;
NutsCp.of(session)
    .from("http://my-server.com/file.pdf")
    .to("/home/my-file")
    .setProgressMonitor(true)
    .setValidator((in)->checkSHA1Hash(in))
    .run();

NutsPs ps=NutsPs.of(session)
if(ps.isSupportedKillProcess()){
    ps.killProcess("1234");
}
```



## 4. Nuts as a Framework

- Nuts Application Framework
  - Add support for Base Directory API
    - API to manage per application directories (log, cache, config,...)
  - Add support for Base Commandline API
    - standardized commandline options
    - inherit common options (--table, --json, ...)



## 5. Nuts as a Framework

- Add support for Application Lifecycle (Hooks for install, update, uninstall)
- Add support for auto update
- Add support for isolated input/output (via session in/out)
- Add support for Desktop Integration
  - Add Shortcuts, Menus
  - Add Aliases





## 6. Nuts Application Framework

- Implement NutsApplication
- Add Description Properties in pom.xml



## 7. NAF Example

```
public class Main implements NutsApplication {
    public static void main(String[] args) {
        new Main().runAndExit(args);
    }
    @Override
    public void run(NutsApplicationContext applicationContext) {
        NutsCommandLine cmd=applicationContext.commandLine();
        ...
    }
}
```



## 8. NAF Example

```
public class Main implements NutsApplication {
    public static void main(String[] args) {new Main().runAndExit(args);}
    @Override
    public void run(NutsApplicationContext applicationContext) {
        NutsCommandLine cmd=applicationContext.commandLine();
        ...
    }
    @Override
    public void onInstallApplication(NutsApplicationContext applicationContext) {}
    @Override
    public void onUpdateApplication(NutsApplicationContext applicationContext) {}
    @Override
    public void onUninstallApplication(NutsApplicationContext applicationContext) {}
}
```



## 9. NAF + Spring

```
@SpringBootApplication
@Import(NutsSpringBootConfig.class)
public class AppExample implements NutsApplication {
    public static void main(String[] args) {
        SpringApplication.run(AppExample.class, args);
    }

    @Override
    public void run(NutsApplicationContext context) {
        NutsPrintStream out = context.getSession().out();
        out.println("Hello ##World##");
    }
}
```



```
<dependency>
  <groupId>net.thevpc.nuts</groupId>
  <artifactId>nuts</artifactId>
  <version>0.8.3</version>
</dependency>
<dependency>
  <groupId>net.thevpc.nuts.lib</groupId>
  <artifactId>nlib-spring-boot</artifactId>
  <version>0.8.3.1</version>
</dependency>
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter</artifactId>
</dependency>
```

## 9.1. Conclusion

- **nuts** can be used as a library or as a framework



- Using **nuts** provides many valuable features
- 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

please support us by starring our repo at

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

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

[nuts.packagemanager@gmail.com](mailto:nuts.packagemanager@gmail.com)