

nuts, the Java Package Manager

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

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

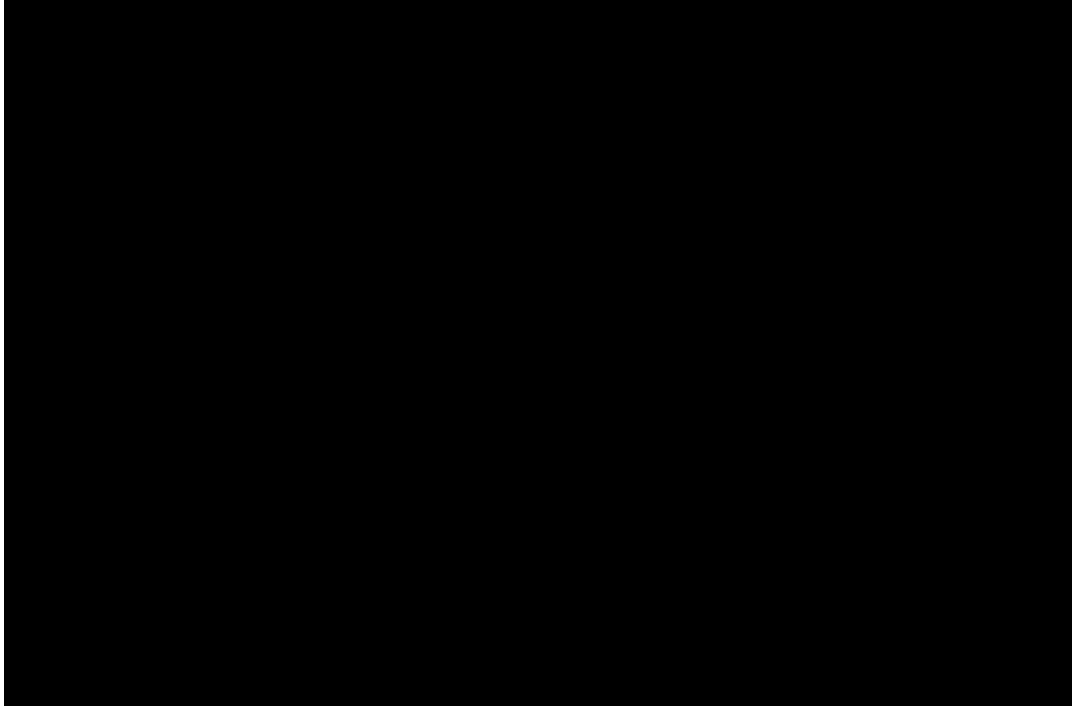
nuts.packagemanager@gmail.com

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>  
  Ê <groupId>net. thevpc. nuts</groupId>  
  Ê <artifactId>nuts</artifactId>  
  Ê <version>0. 8. 3</version>  
</dependency>
```

3. Nuts as A Library

¥ You can add runtime to force the runtime version

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

3.1. Session API

```
Ê  NutsSession session=Nuts.openWorkspace("-Z", "-S", "y", "--json");
Ê  session.setConfirm(NutsConfirmationMode.ASK);
Ê  session.setOutputFormat(NutsContentType.XML);

Ê  session.out().println("Hello");
Ê  session.out().printlnf("Hello");

Ê  session.out().println(Arrays.asList("Hello"));
Ê  session.out().printlnf("Hello");
```

3.2. Std In/Out/Err API

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

3.3. Find API

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


3.4. ClassPath API

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

3.5. NTF API

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

3.6. Format API

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

3.7. Format API

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

3.8. Exec API

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

3.9. IO API

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

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

4. Nuts as a Framework

¥ Nuts Application Framework

- ! Add support for Base Directory API

- " API to manage per application directories (log, cache, config,É)

- ! Add support for Base Commandline API

- " standardized commandline options

- " inherit common options (--table, --json, É)

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##");
    }
}
```

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