IT'S TIME FOR THE RESOLUTION

TIM VERBELEN
Senior Researcher
imec – Ghent University
! could not resolve the bundles:
[that.awesome.library.i.found-1.0.0
org.osgi.framework.BundleException: Unable to resolve that.awesome.library.i.found [1](R 1.0): missing requirement [that.awesome.library.i.found [1](R 1.0)] osgi.wiring.package;
(&(osgi.wiring.package=some.exotic.package)(version>=1.0.0)
(!(version>=2.0.0)))
Unresolved requirements: [[that.awesome.library.i.found
[1](R 1.0)] osgi.wiring.package;
(&(osgi.wiring.package=some.exotic.package)(version>=1.0.0)
(!(version>=2.0.0))]]
WHEN DO YOU RESOLVE?
WHEN DO YOU RESOLVE?

AT RUNTIME?
WHEN DO YOU RESOLVE?

AT RUNTIME?

AT LAUNCH TIME?
WHEN DO YOU RESOLVE?

AT RUNTIME?

AT LAUNCH TIME?

AT DEVELOPMENT TIME?
RESOLVING IMPORT-PACKAGES

foo.bar v1.0.0
Export-Package
  foo.bar.api; version=1.0.0

Private-Package
  foo.bar.impl

Import-Package
  foo.bar.util; version=[1.0.0, 1.1.0)
OSGi BUNDLES AND PACKAGES

RESOLVING IMPORT-PACKAGES

foo.bar v1.0.0
Export-Package
foo.bar.api; version=1.0.0
Private-Package
foo.bar.impl
Import-Package
foo.bar.util; version=[1.0.0, 1.1.0)
RESOLVING IMPORT-PACKAGES

foo.bar util v1.0.0
Export-Package
foo.bar.util; version=1.0.0

foo.bar v1.0.0
Export-Package
foo.bar.api; version=1.0.0
Private-Package
foo.bar.impl
Import-Package
foo.bar.util; version=[1.0.0, 1.1.0)
RESOLVING AGAINST A REPOSITORY

Resolver

resolve foo.bar v1.0.0

Repository

- foo.bar v1.0.0
  - foo.bar.api; version=1.0.0
  - foo.bar.util; version=[1.0.0, 1.1.0)
- foo.bar.util v2.0.0
  - foo.bar.util; version=2.0.0
- com.car v1.0.0
  - com.car.api; version=1.0.0
- foo.bar.util v1.0.0
  - foo.bar.util; version=1.0.0

...
RESOLVING AGAINST A REPOSITORY

Resolver

resolve

foo.bar v1.0.0

Repository

foo.bar v1.0.0
  foo.bar.api; version=1.0.0
  foo.bar.util; version=[1.0.0, 1.1.0)

foo.bar.util v2.0.0
  foo.bar.util; version=2.0.0

com.car v1.0.0
  com.car.api; version=1.0.0

foo.bar.util v1.0.0
  foo.bar.util; version=1.0.0

...
RESOLVING PACKAGES
OSGI EXTENDERS
DO STUFF ON BEHALF OF ANOTHER BUNDLE

**foo.bar v1.0.0**
Service-Component:
OSGI-INF/foo.bar.BarImpl.xml

**felix.scr**

**OSGi framework**

**Resolver**
DO STUFF ON BEHALF OF ANOTHER BUNDLE

foo.bar v1.0.0

OSGI-INF/configurator/config.json
REQUIREMENTS AND CAPABILITIES

Requirement

- Something the bundle needs
  - Namespace
  - Directives (i.e. a filter)

Capability

- Something the bundle offers
  - Namespace
  - Attributes (i.e. key-value pairs)
**REQUIREMENTS AND CAPABILITIES**

**Requirement**

- Something the bundle needs
  - Namespace
  - Directives (i.e. a filter)

```
osgi.wiring.package;
filter:="(&(osgi.wiring.package=foo.bar.util)(version>=1.0.0)!(version>=2.0.0))"
```

```
osgi.extender;
filter:="(&(osgi.extender=osgi.component)(version>=1.4.0)!(version>=2.0.0))"
```

**Capability**

- Something the bundle offers
  - Namespace
  - Attributes (i.e. key-value pairs)

```
osgi.wiring.package;
osgi.wiring.package="foo.bar.util"
version="1.0.0"
```

```
osgi.extender;
osgi.extender="osgi.component"
```
RESOLVING CAPABILITIES AND REQUIREMENTS
OSGI SERVICES
REQUIRE AN OSGI SERVICE

foo.bar.api v1.0.0

foo.bar.api; version=1.0.0
OSGI SERVICES
REQUIRE AN OSGI SERVICE

foo.bar.api v1.0.0
foo.bar.api; version=1.0.0

import

foo.bar.impl v1.0.0
foo.bar.api.BarService
OSGI SERVICES
REQUIRE AN OSGI SERVICE

foo.bar.api v1.0.0
foo.bar.api; version=1.0.0

import

foo.bar.consumer v1.0.0

import

foo.bar.impl v1.0.0

foo.bar.api.BarService
OSGI SERVICES
REQUIRE AN OSGI SERVICE

---

Requiring an OSGI Service

```
foo.bar.api v1.0.0
foo.bar.api; version=1.0.0

import foo.bar.api.BarService

Provide-Capability
osgi.service;
objectClass:List<String>="foo.bar.api.BarService"

effective:=active
```

---

Importing OSGI Services

```
import foo.bar.impl

import foo.bar.consumer

Require-Capability
osgi.service;
filter:="(objectClass=foo.bar.api.BarService)"

effective:=active
```

---

Graphical Representation:

- foo.bar.api v1.0.0
- foo.bar.consumer v1.0.0
- foo.bar.impl v1.0.0
- foo.bar.api.BarService

Connections:

1. foo.bar.api v1.0.0 -> import foo.bar.impl
2. foo.bar.consumer v1.0.0 -> foo.bar.api.BarService
3. foo.bar.impl v1.0.0 -> foo.bar.api.BarService
REQUIRE AN OSGI SERVICE

foo.bar.api v1.0.0
foo.bar.api; version=1.0.0

foo.bar.impl v1.0.0

foo.bar.consumer v1.0.0

import

foo.bar.api.BarService

Provide-Capability
osgi.service;
objectClass:List<String>="foo.bar.api.BarService"

effective:=active

Require-Capability
osgi.service;
filter:="(objectClass=foo.bar.api.BarService)"

effective:=active
RESOLVING SERVICES
DECLARING REQUIREMENTS AND CAPABILITIES

USE BUNDLE ANNOTATIONS

```java
@Capability (namespace="your.name.space",
             name="foo",  version="1.0.0")

@Requirement (namespace="your.name.space",
               filter="(name=foo)")
```
DECLARING REQUIREMENTS AND CAPABILITIES WITH BUNDLE ANNOTATIONS
SO WHAT ABOUT INSTALLING AT RUNTIME?
USE THE REPOSITORY AND RESOLVER SERVICES
SO WHAT ABOUT INSTALLING AT RUNTIME?
USE THE REPOSITORY AND RESOLVER SERVICES
SO WHAT ABOUT INSTALLING AT RUNTIME?
USE THE REPOSITORY AND RESOLVER SERVICES

Deployment Agent

Repository

foo.bar v1.0.0
foo.bar.api; version=1.0.0
foo.bar.util; version=[1.0.0, 1.1.0)

foo.bar.util v2.0.0
foo.bar.util; version=2.0.0

com.car v1.0.0
com.car.api; version=1.0.0

foo.bar.util v1.0.0
foo.bar.util; version=1.0.0

...
RESOLVER SERVICE

- Resolver
  Map<Resource, List<Wire>> resolve(ResolveContext context) throws ResolutionException

- ResolveContext
  Collection<Resource> getMandatoryResources()
  what to resolve?

  List<Capability> findProviders(Requirement requirement)
  where to find capabilities to resolve it?
RESOLVING AT RUNTIME
CONCLUSION

- Resolving is a gain, not a pain
- Resolve early and often
- Curate your repositories
- Use the (BND) tooling
embracing a better life