



FREE eBook

LEARNING

osgi

Free unaffiliated eBook created from
Stack Overflow contributors.

#osgi

Table of Contents

| | |
|--|----------|
| About | 1 |
| Chapter 1: Getting started with osgi | 2 |
| Remarks..... | 2 |
| Versions..... | 2 |
| Examples..... | 3 |
| Downloading and Using Equinox..... | 3 |
| Using Apache Felix..... | 4 |
| Chapter 2: Using Pax Exam for integration testing OSGi applications | 6 |
| Introduction..... | 6 |
| Examples..... | 6 |
| Getting Started..... | 6 |
| Credits | 8 |

About

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [osgi](#)

It is an unofficial and free osgi ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official osgi.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Chapter 1: Getting started with osgi

Remarks

OSGi is a dynamic module system specification for the JVM. It has specifications with different versions and audiences. There are multiple implementations too:

- [Equinox](#),
- [Felix](#),
- [Knopflerfish](#)

Eclipse from its version 3 is based on OSGi. NetBeans is also capable to handle OSGi modules since 6.9. Liferay portal also started to use OSGi as its modularization foundation since its version 7.

Versions

| Version | Release date |
|-------------------|--------------|
| SGR1 | 2000-05-01 |
| SPR2 | 2001-10-01 |
| SR3 | 2003-03-01 |
| SR4 | 2005-11-01 |
| SR4.1 | 2007-05-01 |
| SR4.2 Core | 2009-09-01 |
| SR4.2 Compendium | 2009-09-01 |
| SR4.2 Enterprise | 2010-03-01 |
| SR4.3 Core | 2011-04-01 |
| SR4.3 Compendium | 2012-05-01 |
| SR4.3 Residential | 2012-05-01 |
| SR5 Core | 2012-06-01 |
| SR5 Enterprise | 2012-06-01 |
| SR6 Core | 2014-06-01 |
| SR6 Compendium | 2014-06-01 |

| Version | Release date |
|-----------------|--------------|
| SR6 Enterprise | 2015-08-01 |
| SR6 Residential | 2015-08-01 |

Note: <https://www.osgi.org/developer/specifications/> does not mention exact dates, so used first day of the month in the table.

Examples

Downloading and Using Equinox

Download and extract the OSGi starter kit for your platform from [Equinox download page for Neon release](#).

Start the framework from the `rt/plugins` folder with the following command (or your platform's `rt` executable from the `rt` folder):

```
rt/plugins$ java -jar org.eclipse.equinox.launcher_1.3.200.v20160318-1642.jar -console
osgi>
```

In the OSGi console prompt, you can manage the bundles, for example:

```
osgi>ss
"Framework is launched."

id      State      Bundle
0       ACTIVE    org.eclipse.osgi_3.11.0.v20160603-1336
        Fragments=1
1       RESOLVED  org.eclipse.osgi.compatibility.state_1.0.200.v20160504-1419
        Master=0
2       RESOLVED  org.apache.commons.codec_1.6.0.v201305230611
3       RESOLVED  org.apache.commons.logging_1.1.1.v201101211721
4       ACTIVE    org.apache.felix.gogo.command_0.10.0.v201209301215
5       ACTIVE    org.apache.felix.gogo.runtime_0.10.0.v201209301036
6       ACTIVE    org.apache.felix.gogo.shell_0.10.0.v201212101605
7       RESOLVED  org.apache.httpcomponents.httpClient_4.3.6.v201511171540
8       RESOLVED  org.apache.httpcomponents.httpcore_4.3.3.v201411290715
9       STARTING  org.eclipse.core.jobs_3.8.0.v20160509-0411
10      STARTING  org.eclipse.ecf_3.8.0.v20160405-1820
        Fragments=17
11      STARTING  org.eclipse.ecf.filetransfer_5.0.0.v20160405-1820
12      STARTING  org.eclipse.ecf.identity_3.7.0.v20160405-1820
13      STARTING  org.eclipse.ecf.provider.filetransfer_3.2.200.v20160405-1820
        Fragments=16
14      STARTING  org.eclipse.ecf.provider.filetransfer.httpClient4_1.1.100.v20160405-1820
        Fragments=15
15      RESOLVED  org.eclipse.ecf.provider.filetransfer.httpClient4.ssl_1.1.0.v20160405-1820
        Master=14
16      RESOLVED  org.eclipse.ecf.provider.filetransfer.ssl_1.0.0.v20160405-1820
        Master=13
17      RESOLVED  org.eclipse.ecf.ssl_1.2.0.v20160405-1820
        Master=10
```

```

18     STARTING    org.eclipse.equinox.app_1.3.400.v20150715-1528
19     ACTIVE      org.eclipse.equinox.common_3.8.0.v20160509-1230
20     STARTING    org.eclipse.equinox.concurrent_1.1.0.v20130327-1442
21     ACTIVE      org.eclipse.equinox.console_1.1.200.v20150929-1405
22     RESOLVED    org.eclipse.equinox.ds_1.4.400.v20160226-2036
23     STARTING    org.eclipse.equinox.frameworkadmin_2.0.300.v20160504-1450
24     ACTIVE      org.eclipse.equinox.frameworkadmin.equinox_1.0.700.v20160102-2223
25     RESOLVED    org.eclipse.equinox.launcher_1.3.200.v20160318-1642
                Fragments=26
26     RESOLVED    org.eclipse.equinox.launcher.win32.win32.x86_64_1.1.400.v20160518-1444
                Master=25
27     STARTING    org.eclipse.equinox.p2.artifact.repository_1.1.500.v20160419-0834
28     ACTIVE      org.eclipse.equinox.p2.console_1.0.500.v20160504-1450
29     ACTIVE      org.eclipse.equinox.p2.core_2.4.100.v20160419-0834
30     STARTING    org.eclipse.equinox.p2.director_2.3.300.v20160504-1450
31     STARTING    org.eclipse.equinox.p2.engine_2.4.100.v20160419-0834
32     STARTING    org.eclipse.equinox.p2.garbagecollector_1.0.300.v20160504-1450
33     RESOLVED    org.eclipse.equinox.p2.jarprocessor_1.0.500.v20160504-1450
34     ACTIVE      org.eclipse.equinox.p2.metadata_2.3.100.v20160427-2220
35     STARTING    org.eclipse.equinox.p2.metadata.repository_1.2.300.v20160419-0834
36     STARTING    org.eclipse.equinox.p2.operations_2.4.200.v20160504-1450
37     ACTIVE      org.eclipse.equinox.p2.repository_2.3.200.v20160421-0324
38     STARTING    org.eclipse.equinox.p2.touchpoint.eclipse_2.1.400.v20160419-0834
39     STARTING    org.eclipse.equinox.p2.touchpoint.natives_1.2.100.v20160419-0834
40     STARTING    org.eclipse.equinox.p2.transport.ecf_1.1.200.v20160606-1311
41     STARTING    org.eclipse.equinox.preferences_3.6.0.v20160120-1756
42     ACTIVE      org.eclipse.equinox.registry_3.6.100.v20160223-2218
43     STARTING    org.eclipse.equinox.security_1.2.200.v20150715-1528
                Fragments=44
44     RESOLVED    org.eclipse.equinox.security.win32.x86_64_1.0.100.v20130327-1442
                Master=43
45     STARTING    org.eclipse.equinox.simpleconfigurator_1.1.200.v20160504-1450
46     ACTIVE      org.eclipse.equinox.simpleconfigurator.manipulator_2.0.200.v20160504-1450
47     STARTING    org.eclipse.equinox.util_1.0.500.v20130404-1337
48     RESOLVED    org.eclipse.osgi.services_3.5.100.v20160504-1419
49     RESOLVED    org.sat4j.core_2.3.5.v201308161310
50     RESOLVED    org.sat4j.pb_2.3.5.v201404071733
51     RESOLVED    org.tukaani.xz_1.3.0.v201308270617

```

Using Apache Felix

Download the [Apache Felix Framework Distribution](#) and extract it into a directory:

```

$ tar xf org.apache.felix.main.distribution-5.4.0.tar.gz
$ cd felix-framework-5.4.0

```

And then start the framework with the following command:

```

$ java -jar bin/felix.jar
-----
Welcome to Apache Felix Gogo

g!

```

By default, Felix uses [Apache Felix Gogo](#) as its interactive shell. The default command prompt, `g!`, is shown above.

The Gogo shell provides some simple builtin commands to interact with both the Gogo runtime environment, as well as with the OSGi framework itself. Type the `help` command at the `g!` prompt to see a list of builtin commands.

One example is the `lb` (short for *list bundles*) command which outputs a listing of the OSGi bundles that are currently installed in the framework:

```
g! lb
START LEVEL 1
  ID|State      |Level|Name
  0|Active      |    0|System Bundle (5.4.0)|5.4.0
  1|Active      |    1|Apache Felix Bundle Repository (2.0.6)|2.0.6
  2|Active      |    1|Apache Felix Gogo Command (0.16.0)|0.16.0
  3|Active      |    1|Apache Felix Gogo Runtime (0.16.2)|0.16.2
  4|Active      |    1|Apache Felix Gogo Shell (0.10.0)|0.10.0
```

Read [Getting started with osgi online](https://riptutorial.com/osgi/topic/2422/getting-started-with-osgi): <https://riptutorial.com/osgi/topic/2422/getting-started-with-osgi>

Chapter 2: Using Pax Exam for integration testing OSGi applications

Introduction

Pax Exam allows testing of bundles within an OSGi container (e.g. AEM, Apache Karaf). Pax Exam is usually used in conjunction with JUnit.

Examples

Getting Started

Here is an example of a test using Pax Exam.

```
package com.example.project.test;

import static org.junit.Assert.*;
import static org.ops4j.pax.exam.CoreOptions.*;

import javax.inject.Inject;

import org.junit.Test;
import org.junit.runner.RunWith;
import org.ops4j.pax.exam.Configuration;
import org.ops4j.pax.exam.Option;
import org.ops4j.pax.exam.junit.PaxExam;
import org.ops4j.pax.exam.regression.pde.HelloService;
import org.ops4j.pax.exam.spi.reactors.*;

@RunWith(PaxExam.class)
@ExamReactorStrategy(PerMethod.class)
public class SampleTest {

    @Inject
    private HelloService helloService;

    @Configuration
    public Option[] config() {

        return options(
            mavenBundle("com.example.myproject", "myproject-api", "1.0.0-SNAPSHOT"),
            bundle("http://www.example.com/repository/foo-1.2.3.jar"),
            junitBundles()
        );
    }

    @Test
    public void getHelloService() {
        assertNotNull(helloService);
        assertEquals("Hello Pax!", helloService.getMessage());
    }
}
```


Code is from [Pax Exam 4 Page](#)

Read [Using Pax Exam for integration testing OSGi applications](#) online:

<https://riptutorial.com/osgi/topic/9682/using-pax-exam-for-integration-testing-osgi-applications>

Credits

| S. No | Chapters | Contributors |
|-------|--|---|
| 1 | Getting started with osgi | Achim Nierbeck , Community , Gábor Bakos , Kerry , khakiout , Sean Bright , vorburger |
| 2 | Using Pax Exam for integration testing OSGi applications | khakiout |