



FREE eBook

LEARNING wildfly

Free unaffiliated eBook created from
Stack Overflow contributors.

#wildfly

Table of Contents

About	1
Chapter 1: Getting started with wildfly	2
Remarks.....	2
Versions.....	2
Examples.....	2
Installation.....	2
Running it via Docker.....	3
Starting the server.....	3
Chapter 2: Adding modules to wildfly	6
Examples.....	6
Adding module com.stackoverflow.....	6
Adding module com.stackoverflow with versioning.....	6
Multiple modules directories.....	6
Chapter 3: Command line interface (CLI)	8
Examples.....	8
Connecting to the local Wildfly server.....	8
Credits	9

About

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

It is an unofficial and free wildfly 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 wildfly.

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 wildfly

Remarks



Wildfly is a Java EE compliant application server. As an application server, its main purpose is to provide a set of tools that Java enterprise applications usually need, such as support for EJBs, JPA, Servlets, JAX-RS, Batch, Security, Transactions, ...

Wildfly is the upstream project used by the commercial offering JBoss Enterprise Application Platform (EAP) by Red Hat.

Versions

Version	Release Date
10.1.0.Final	2016-08-19
10.0.0.Final	2016-01-29
9.0.2.Final	2015-10-26
9.0.1.Final	2015-07-23
8.2.1.Final	2015-07-23
9.0.0.Final	2015-07-02
8.2.0.Final	2014-11-20
8.1.0.Final	2014-05-30
8.0.0.Final	2014-02-11

Examples

Installation

Installing Wildfly is just a matter of unzipping the distribution into your local machine. Wildfly can be downloaded from its [official website](#).

Once it is unzipped go in to bin directory of installation and run `standalone.sh` for Linux systems or `standalone.bat` for Windows systems to start your WildFly instance in default configurations. Once

you see something like

```
13:16:12,503 INFO [org.jboss.as] (Controller Boot Thread) WFLYSRV0025: WildFly Full
10.1.0.Final (WildFly Core 2.2.0.Final) started in 18909ms - Started 331 of 577 services (393
services are lazy, passive or on-demand)
```

then your brand new WildFly instance waiting to welcome you at : <http://localhost:8080/>

Some Linux distributions, such as Fedora, have Wildfly on its repositories and can be installed via YUM/DNF: `dnf install wildfly`. This, however, is not really recommended, as it tends to use slightly different versions of the libraries than the official distribution, which might cause problems that are hard to diagnose/fix.

Running it via Docker

Wildfly, part of the JBoss umbrella of projects, can also be executed via Docker. On a machine with Docker properly configured, run:

```
$ docker run -it jboss/wildfly
```

Once the image is pulled, the container starts and the following line can be seen:

```
09:44:49,225 INFO [org.jboss.as] (Controller Boot Thread) WFLYSRV0025: WildFly Full
10.0.0.Final (WildFly Core 2.0.10.Final) started in 5644ms - Started 267 of 553 services (371
services are lazy, passive or on-demand)
```

This is an "empty" Wildfly server. On real world projects, the base image is meant to be extended so that your application in WAR/EAR packaging format, is added to it, as well as the necessary configuration changes to `standalone/configuration/standalone.xml`.

Starting the server

Once Wildfly is installed by unzipping the distribution, it can be started by running the `standalone.sh` script on the `bin` directory:

```
$ ./bin/standalone.sh
=====

JBoss Bootstrap Environment

JBOSS_HOME: /mnt/storage/tools/servers/wildfly-10.0.0.Final

JAVA: java

JAVA_OPTS: -server -Xms64m -Xmx512m -XX:MetaspaceSize=96M -XX:MaxMetaspaceSize=256m -
Djava.net.preferIPv4Stack=true -Djboss.modules.system.pkgs=org.jboss.byteman -
Djava.awt.headless=true

=====

11:54:33,781 INFO [org.jboss.modules] (main) JBoss Modules version 1.5.1.Final
11:54:34,096 INFO [org.jboss.msc] (main) JBoss MSC version 1.2.6.Final
```

```

11:54:34,193 INFO [org.jboss.as] (MSC service thread 1-6) WFLYSRV0049: WildFly Full
10.0.0.Final (WildFly Core 2.0.10.Final) starting
...
...
11:54:37,653 INFO [org.jboss.as] (Controller Boot Thread) WFLYSRV0025: WildFly Full
10.0.0.Final (WildFly Core 2.0.10.Final) started in 4357ms - Started 273 of 559 services (374
services are lazy, passive or on-demand)

```

With no arguments, the default configuration is used. To override the default configuration you can providing arguments on the command line.

<code>--admin-only</code>	Set the server's running type to ADMIN_ONLY causing it to open administrative interfaces and accept management requests but not start other runtime services or accept end user requests.
<code>-b <value>, -b=<value></code>	Set system property <code>jboss.bind.address</code> to the given value
<code>-b<interface>=<value></code>	Set system property <code>jboss.bind.address.<interface></code> to the given value
<code>-c <config>, -c=<config></code>	Name of the server configuration file to use (default is "standalone.xml") (Same as <code>--server-config</code>)
<code>--debug [<port>]</code>	Activate debug mode with an optional argument to specify the port. Only works if the launch script supports it.
<code>-D<name>[=<value>]</code>	Set a system property
<code>-h, --help</code>	Display this message and exit
<code>--read-only-server-config=<config></code>	Name of the server configuration file to use. This differs from <code>'--server-config'</code> and <code>'-c'</code> in that the original file is never overwritten.
<code>-P <url>, -P=<url>, --properties=<url></code>	Load system properties from the given url
<code>-S<name>[=<value>]</code>	Set a security property
<code>--server-config=<config></code>	Name of the server configuration file to use (default is "standalone.xml") (Same as <code>-c</code>)

```
-u <value>, -u=<value>          Set system property  
                                jboss.default.multicast.address to the  
                                given value  
  
-v, -V, --version              Print version and exit  
  
-secmgr                        Runs the server with a security manager  
                                installed.
```

Read **Getting started with wildfly online**: <https://riptutorial.com/wildfly/topic/1942/getting-started-with-wildfly>

Chapter 2: Adding modules to wildfly

Examples

Adding module com.stackoverflow

- Pack your com.stackoverflow.`{whatever}` classes on a jar called `stackoverflow.jar`
- Create folder `com/stackoverflow/main` on `${WILDFLY_HOME}/modules`
- Put the jar on the last dir
- Create a file called `module.xml` with the following content:

```
<?xml version="1.0" encoding="UTF-8"?>
<module xmlns="urn:jboss:module:1.1" name="com.stackoverflow">
  <resources>
    <resource-root path="stackoverflow.jar"/>
  </resources>
</module>
```

And now you have you module available to applications

Adding module com.stackoverflow with versioning

For some reason you develop a new version of com.stackoverflow (say version 1.1), then you should:

- Create folder `com/stackoverflow/1.1` on `${WILDFLY_HOME}/modules`
- Put the new jar on last dir
- Create file `module.xml` on last dir with the following content

```
<?xml version="1.0" encoding="UTF-8"?>
<module xmlns="urn:jboss:module:1.1" name="com.stackoverflow" slot="1.1">
  <resources>
    <resource-root path="stackoverflow.jar"/>
  </resources>
</module>
```

Note the `slot="1.1"` modification on this example

Multiple modules directories

By default modules are placed on `${WILDFLY}/modules` directory but you can have more directories with modules, just edit your `standalone.conf` (or `standalone.conf.bat` if you are on Microsoft Windows) and properly set the variable `JBOSS_MODULEPATH`

For example in Unix/Linux/MacOSX:

```
JBOSS_MODULEPATH="${JBOSS_HOME}/modules:${JBOSS_HOME}/myownmodules1:${JBOSS_HOME}/myownmodules2"
```


Or in Windows:

```
set  
"JBOSS_MODULEPATH=%JBOSS_HOME%\modules;%JBOSS_HOME%\myownmodules1;%JBOSS_HOME%\myownmodules2"
```

Read [Adding modules to wildfly online](https://riptutorial.com/wildfly/topic/9502/adding-modules-to-wildfly): <https://riptutorial.com/wildfly/topic/9502/adding-modules-to-wildfly>

Chapter 3: Command line interface (CLI)

Examples

Connecting to the local Wildfly server

To connect to a local Wildfly server via the command line, the tool `bin/jboss-cli.sh` can be used:

```
$ ./bin/jboss-cli.sh --connect
[standalone@localhost:9990 /]
```

To connect to a remote Wildfly server, use the `--controller` option:

```
$ ./bin/jboss-cli.sh --connect --controller=localhost:9990
[standalone@localhost:9990 /]
```

Note that the port `9990` is used for the administrative operations, and as such, should *not* be opened to the public internet. By default, Wildfly binds the administrative port to `localhost`, which means that it cannot be accessed from remote hosts. To overcome this, start Wildfly with the `-bmanagement` option, specifying the IP it should bind to:

```
$ ./bin/standalone.sh -bmanagement 0.0.0.0
```

Read Command line interface (CLI) online: <https://riptutorial.com/wildfly/topic/2418/command-line-interface--cli->

Credits

S. No	Chapters	Contributors
1	Getting started with wildfly	cache , Community , Ebuzer Taha KANAT , Federico Sierra , jpkrohling , Maverick
2	Adding modules to wildfly	alphamikevictor
3	Command line interface (CLI)	jpkrohling