

Instalación de Maven

Comenzamos actualizando los repositorios:

`sudo apt update`

```
usuario@david:~$ sudo apt update
Hit:1 http://es.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:3 http://es.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:4 http://es.archive.ubuntu.com/ubuntu noble-backports InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
32 packages can be upgraded. Run 'apt list --upgradable' to see them.
usuario@david:~$
```

E instalamos Maven:

`sudo apt install maven`

```
usuario@david:~$ sudo apt install maven
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  alsa-topology-conf alsa-ucm-conf ca-certificates-java default-jre-headless java-common libaopalliance-java
  libapache-pom-java libasound2-data libasound2t64 libatinject-jsr330-api-java libavahi-client3
  libavahi-common-data libavahi-common3 libcdi-api-java libcommons-cli-java libcommons-io-java
  libcommons-lang3-java libcommons-parent-java libcupst64 liberror-prone-java libgeronimo-annotation-1.3-spec-java
  libgeronimo-interceptor-3.0-spec-java libgraphite2-3 libguava-java libguice-java libharfbuzz0b libjansi-java
  libjsr305-java liblcms2-2 libmaven-parent-java libmaven-resolver-java libmaven-shared-utils-java
  libmaven3-core-java libpcsclite1 libplexus-cipher-java libplexus-classworlds-java
  libplexus-component-annotations-java libplexus-interpolation-java libplexus-sec-dispatcher-java
  libplexus-utils2-java libsisu-inject-java libsisu-plexus-java libslf4j-java libwagon-file-java
  libwagon-http-shaded-java libwagon-provider-api-java openjdk-21-jre-headless
Suggested packages:
  default-jre alsa-utils libasound2-plugins libatinject-jsr330-api-java-doc libel-api-java libcommons-io-java-doc
  cups-common libasm-java libcglib-java libjsr305-java-doc liblcms2-utils libmaven-shared-utils-java-doc
  liblogback-java pcsd libplexus-utils2-java-doc junit4 testng libcommons-logging-java liblog4j1.2-java
  libss-mdns fonts-dejavu-extra fonts-ipafont-gothic fonts-ipafont-mincho fonts-wqy-microhei-l fonts-wqy-zenhei
usuario@david:~$
```

Para comprobar que todo ha ido correctamente, podemos ver la versión instalada de Maven: `mvn --v`

```
usuario@david:~$ mvn --v
Apache Maven 3.8.7
Maven home: /usr/share/maven
Java version: 21.0.8, vendor: Ubuntu, runtime: /usr/lib/jvm/java-21-openjdk-amd64
Default locale: en_US, platform encoding: UTF-8
OS name: "linux", version: "6.8.0-86-generic", arch: "amd64", family: "unix"
usuario@david:~$
```

Instalación del servidor de aplicaciones WildFly

Instalar Java Development Kit (JDK)

sudo apt update

sudo apt -y install default-jdk

```
Preparing to unpack .../081-libxcb-shape0_1.15-1ubuntu2_amd64.deb ...
Unpacking libxcb-shape0:amd64 (1.15-1ubuntu2) ...
Selecting previously unselected package libxft2:amd64.
Preparing to unpack .../082-libxft2_2.3.6-1build1_amd64.deb ...
Unpacking libxft2:amd64 (2.3.6-1build1) ...
Selecting previously unselected package libxkbfile1:amd64.
Preparing to unpack .../083-libxkbfile1_1%3a1.1.0-1build4_amd64.deb ...
Unpacking libxkbfile1:amd64 (1:1.1.0-1build4) ...
Selecting previously unselected package libxv1:amd64.
Preparing to unpack .../084-libxv1_2%3a1.0.11-1.1build1_amd64.deb ...
Unpacking libxv1:amd64 (2:1.0.11-1.1build1) ...
Selecting previously unselected package libxxf86dga1:amd64.
Preparing to unpack .../085-libxxf86dga1_2%3a1.1.5-1build1_amd64.deb ...
Unpacking libxxf86dga1:amd64 (2:1.1.5-1build1) ...
Selecting previously unselected package x11-utils.
Preparing to unpack .../086-x11-utils_7.7+6build2_amd64.deb ...
Unpacking x11-utils (7.7+6build2) ...
Selecting previously unselected package libatk-wrapper-java.
Preparing to unpack .../087-libatk-wrapper-java_0.40.0-3build2_all.deb ...
Unpacking libatk-wrapper-java (0.40.0-3build2) ...
Selecting previously unselected package libatk-wrapper-java-jni:amd64.
Preparing to unpack .../088-libatk-wrapper-java-jni_0.40.0-3build2_amd64.deb ...
Unpacking libatk-wrapper-java-jni:amd64 (0.40.0-3build2) ...
Selecting previously unselected package libgdk-pixbuf2.0-bin.
Preparing to unpack .../089-libgdk-pixbuf2.0-bin_2.42.10+dfsg-3ubuntu3.2_amd64.deb ...
Unpacking libgdk-pixbuf2.0-bin (2.42.10+dfsg-3ubuntu3.2) ...
Selecting previously unselected package libgtk-3-bin.
Preparing to unpack .../090-libgtk-3-bin_3.24.41-4ubuntu1.3_amd64.deb ...
Unpacking libgtk-3-bin (3.24.41-4ubuntu1.3) ...
Progress: [ 42%] [#####.....]
```

WildFly soporta Java 17 y Java 11, podéis confirmar vuestra versión instalada así:

java --version

```
usuario@david:~$ java --version
openjdk 21.0.8 2025-07-15
OpenJDK Runtime Environment (build 21.0.8+9-Ubuntu-0ubuntu124.04.1)
OpenJDK 64-Bit Server VM (build 21.0.8+9-Ubuntu-0ubuntu124.04.1, mixed mode, sharing)
usuario@david:~$
```

Descargar el tarball (archivo comprimido) de WildFly

utilizaremos las herramientas *curl* y *wget*. Si no las tuviéramos instaladas, primero lo haremos:

sudo apt install curl wget

```
usuario@david:~$ sudo apt install curl wget
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
curl is already the newest version (8.5.0-2ubuntu10.6).
curl set to manually installed.
wget is already the newest version (1.21.4-1ubuntu4.1).
wget set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 32 not upgraded.
usuario@david:~$
```

```
WILDFLY_RELEASE=$(curl -s https://api.github.com/repos/wildfly/wildfly/releases/latest|grep
tag_name|cut -d '"' -f 4)
```

[https://github.com/Wildfly/wildfly/releases/download/\\${WILDFLY_RELEASE}/wildfly-\\${WILDFLY_RELEASE}.tar.gz](https://github.com/Wildfly/wildfly/releases/download/${WILDFLY_RELEASE}/wildfly-${WILDFLY_RELEASE}.tar.gz)

Descomprimimos:

```
wildfly-38.0.0.Final/modules/system/layers/base/com/fasterxml/jackson/dataformat/jackson-dataformat-yaml/main/jackson-
dataformat-yaml-2.18.4.jar
wildfly-38.0.0.Final/modules/system/layers/base/com/fasterxml/jackson/dataformat/jackson-dataformat-yaml/main/module.
xml
wildfly-38.0.0.Final/domain/configuration/application-roles.properties
wildfly-38.0.0.Final/domain/configuration/domain.xml
wildfly-38.0.0.Final/domain/configuration/logging.properties
wildfly-38.0.0.Final/domain/configuration/host.xml
wildfly-38.0.0.Final/domain/configuration/mgmt-groups.properties
wildfly-38.0.0.Final/domain/configuration/default-server-logging.properties
wildfly-38.0.0.Final/domain/configuration/application-users.properties
wildfly-38.0.0.Final/domain/configuration/host-primary.xml
wildfly-38.0.0.Final/domain/configuration/host-secondary.xml
wildfly-38.0.0.Final/domain/configuration/mgmt-users.properties
usuario@david:~$
```

Movemos el contenido descomprimido al directorio /opt:

```
sudo mv wildfly-${WILDFLY_RELEASE} /opt/wildfly
```

```
usuario@david:~$ sudo mv wildfly-${WILDFLY_RELEASE} /opt/wildfly
usuario@david:~$
```

Configurar WildFly como un servicio más de Systemd

Para empezar añadido el usuario de sistema wildfly que es el que se encargará del servicio:

```
sudo groupadd --system wildfly
```

```
sudo useradd -s /sbin/nologin --system -d /opt/wildfly -g wildfly wildfly
```

```
usuario@david:~$ sudo groupadd --system wildfly
usuario@david:~$ sudo useradd -s /sbin/nologin --system -d /opt/wildfly -g wildfly wildfly
usuario@david:~$
```

Creamos el directorio que contendrá las configuraciones de WildFly:

```
sudo mkdir /etc/wildfly
```

```
usuario@david:~$ sudo mkdir /etc/wildfly
usuario@david:~$
```

Copiamos el archivo del servicio así como los scripts de inicio al directorio

```
sudo cp /opt/wildfly/docs/contrib/scripts/systemd/wildfly.conf /etc/wildfly/
```

```
sudo cp /opt/wildfly/docs/contrib/scripts/systemd/wildfly.service /etc/systemd/system/
```

```
sudo cp /opt/wildfly/docs/contrib/scripts/systemd/launch.sh /opt/wildfly/bin/
```

```
sudo chmod +x /opt/wildfly/bin/launch.sh
```

```
usuario@david:~$ sudo cp /opt/wildfly/docs/contrib/scripts/systemd/wildfly.conf /etc/wildfly/
usuario@david:~$ sudo cp /opt/wildfly/docs/contrib/scripts/systemd/wildfly.service /etc/systemd/system/
usuario@david:~$ sudo cp /opt/wildfly/docs/contrib/scripts/systemd/launch.sh /opt/wildfly/bin/
usuario@david:~$ sudo chmod +x /opt/wildfly/bin/launch.sh
usuario@david:~$
```

Le damos los permisos adecuados al directorio /opt/wildfly:

```
sudo chown -R wildfly:wildfly /opt/wildfly
```

```
usuario@david:~$ sudo chown -R wildfly:wildfly /opt/wildfly
usuario@david:~$
```

Recargamos los servicios de systemd (sistema que se encarga de manejar los servicios en nuestra Debian):

```
sudo systemctl daemon-reload
```

```
usuario@david:~$ sudo systemctl daemon-reload
usuario@david:~$
```

Iniciamos el servicio WildFly y lo habilitamos para que se inicie automáticamente al arrancar la máquina:

```
sudo systemctl start wildfly
```

```
usuario@david:~$ sudo systemctl start wildfly
usuario@david:~$
```

```
sudo systemctl enable wildfly
```

```
usuario@david:~$ sudo systemctl enable wildfly
Created symlink /etc/systemd/system/multi-user.target.wants/wildfly.service → /etc/systemd/system/wildfly.service.
usuario@david:~$
```

Comprobamos que se está ejecutando:

```
sudo systemctl status wildfly
```

```
usuario@david:~$ sudo systemctl status wildfly
● wildfly.service - The WildFly Application Server
   Loaded: loaded (/etc/systemd/system/wildfly.service; enabled; preset: enabled)
   Active: active (running) since Fri 2025-10-31 08:12:10 UTC; 1min 2s ago
     Main PID: 6886 (launch.sh)
       Tasks: 64 (limit: 11882)
    Memory: 274.4M (peak: 284.3M)
       CPU: 9.970s
    CGroup: /system.slice/wildfly.service
            └─6886 /bin/bash /opt/wildfly/bin/launch.sh standalone standalone.xml 0.0.0.0
              └─6887 /bin/sh /opt/wildfly/bin/standalone.sh -c standalone.xml -b 0.0.0.0
                └─7008 java "-D[Standalone]" "-Djdk.serialFilter=maxbytes=10485760;maxdepth=128;maxarray=100000;maxrefs=
Oct 31 08:12:10 david systemd[1]: Started wildfly.service - The WildFly Application Server.
lines 1-13/13 (END)
```

WildFly nos proporciona un archivo para administrar los usuarios, así que en primer lugar y para crear un usuario:

```
sudo /opt/wildfly/bin/add-user.sh
```

```
usuario@david:~$ sudo /opt/wildfly/bin/add-user.sh

What type of user do you wish to add?
  a) Management User (mgmt-users.properties)
  b) Application User (application-users.properties)
(a): a
```

Ponemos el nombre de usuario

```
Enter the details of the new user to add.
Using realm 'ManagementRealm' as discovered from the existing property files.
Username : david
```

Ponemos la contraseña y le damos enter y luego yes

```
Password recommendations are listed below. To modify these restrictions edit the add-user.properties configuration file.
- The password should be different from the username
- The password should not be one of the following restricted values {root, admin, administrator}
- The password should contain at least 8 characters, 1 alphabetic character(s), 1 digit(s), 1 non-alphanumeric symbol(s)
Password :
WFLYDM0099; Password should have at least 8 characters!
Are you sure you want to use the password entered yes/no? y
Re-enter Password :
What groups do you want this user to belong to? (Please enter a comma separated list, or leave blank for none)[ ]:
```

Vemos que se ha creado correctamente

```
Added user 'david' to file '/opt/wildfly/standalone/configuration/mgmt-users.properties'
Added user 'david' to file '/opt/wildfly/domain/configuration/mgmt-users.properties'
Added user 'david' with groups to file '/opt/wildfly/standalone/configuration/mgmt-groups.properties'
Added user 'david' with groups to file '/opt/wildfly/domain/configuration/mgmt-groups.properties'
usuario@david:~$
```

Para ser capaces de ejecutar los scripts de WildFly desde nuestra sesión actual, añadimos el directorio /opt/wildfly/bin/ a nuestro PATH:

```
cat >> ~/.bashrc <<EOF
```

```
export WILDFLY_BIN="/opt/wildfly/bin/"
```

```
export PATH=$PATH:$WILDFLY_BIN
```

```
EOF
```

Y para no tener que reiniciar nuestra sesión actual SSH, aplicamos los cambios en la sesión actual así:

```
source ~/.bashrc
```

```
usuario@david:~$ cat >> ~/.bashrc <<EOF
N="/"> export WILDFLY_BIN="/opt/wildfly/bin/"
> export PATH=$PATH:$WILDFLY_BIN
> EOF
usuario@david:~$ source ~/.bashrc
usuario@david:~$
```

Y validamos que nos podemos conectar sin problemas:

```
jboss-cli.sh --connect
```

```
usuario@david:~$ jboss-cli.sh --connect
Authenticating against security realm: ManagementRealm
Username: david
Password:
[standalone@localhost:9990 /] exit
```

Accediendo a la consola de administración desde la interfaz web

Debemos modificar el archivo /opt/wildfly/bin/launch.sh/opt/wildfly/bin/launch.sh y dejarlo así:

```
GNU nano 7.2 /opt/wildfly/bin/launch.sh *
#!/bin/bash

if [ "$wildfly_HOME" = "x" ]; then
    wildfly_HOME="/opt/wildfly"
fi

if [[ "$1" == "domain" ]]; then
    $wildfly_HOME/bin/domain.sh -c $2 -b $3
else
    $wildfly_HOME/bin/standalone.sh -c $2 -b $3 -bmanagement=0.0.0.0
fi
```

Reiniciamos el servicio tras el cambio:

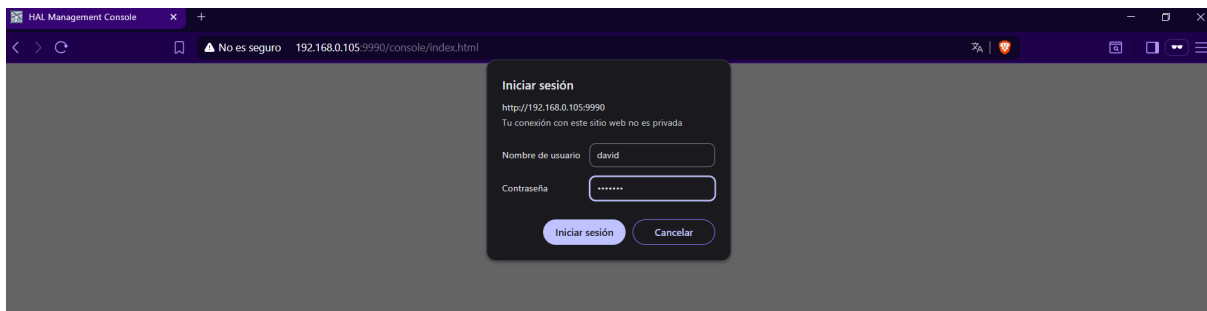
```
sudo systemctl restart wildfly
```

Comprobamos su estado:

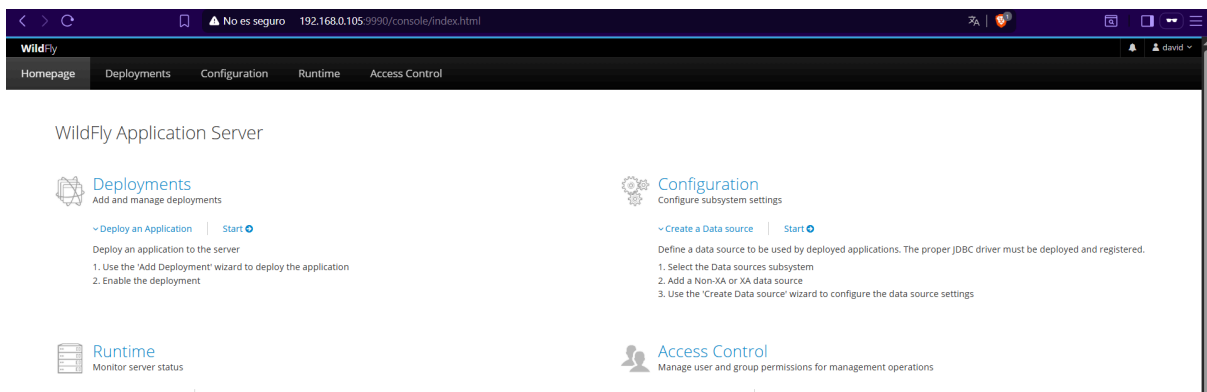
```
systemctl status wildfly
```

```
usuario@david:~$ sudo systemctl restart wildfly
usuario@david:~$ sudo systemctl status wildfly
● wildfly.service - The WildFly Application Server
   Loaded: loaded (/etc/systemd/system/wildfly.service; enabled; preset: enabled)
   Active: active (running) since Fri 2025-10-31 08:23:20 UTC; 15s ago
     Main PID: 7453 (launch.sh)
        Tasks: 139 (limit: 11882)
      Memory: 275.5M (peak: 276.0M)
         CPU: 9.793s
       CGroup: /system.slice/wildfly.service
              └─7453 /bin/bash /opt/wildfly/bin/launch.sh standalone standalone.xml 0.0.0.0
                 └─7454 /bin/sh /opt/wildfly/bin/standalone.sh -c standalone.xml -b 0.0.0.0 -bmanagement=0.0.0.0
                    └─7582 java "-D[Standalone]" "-Djdk.serialFilter=maxbytes=10485760;maxdepth=128;maxarray=100000;maxrefs=
Oct 31 08:23:20 david systemd[1]: Started wildfly.service - The WildFly Application Server.
lines 1-13/13 (END)
```


Ponemos `http://Nuestra_ip:9990`



Aparecerá esto



Clonamos el repositorio

sudo git clone <https://github.com/raul-profesor/practica-jakarta-wildfly>

```
usuario@david:~$ sudo git clone https://github.com/raul-profesor/practica-jakarta-wildfly
Cloning into 'practica-jakarta-wildfly'...
remote: Enumerating objects: 54, done.
remote: Counting objects: 100% (54/54), done.
remote: Compressing objects: 100% (36/36), done.
remote: Total 54 (delta 11), reused 37 (delta 2), pack-reused 0 (from 0)
Receiving objects: 100% (54/54), 9.46 KiB | 3.15 MiB/s, done.
Resolving deltas: 100% (11/11), done.
usuario@david:~$
```


Nos movemos aquí y creamos el pom.xml

```
usuario@david:~/practica-jakarta-wildfly$
```

```
GNU nano 7.2 pom.xml
<?xml version="1.0" encoding="UTF-8" ?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">

  <modelVersion>4.0.0</modelVersion>

  <groupId>com.mycompany.myproject</groupId>
  <artifactId>modulename.backend</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <packaging>war</packaging>

  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <java.version>21</java.version>
    <maven.compiler.source>${java.version}</maven.compiler.source>
    <maven.compiler.target>${java.version}</maven.compiler.target>
    <failOnMissingWebXml>false</failOnMissingWebXml>
  </properties>

  <dependencies>

    <dependency>
      <groupId>jakarta.platform</groupId>
      <artifactId>jakarta.jakartaee-api</artifactId>
      <version>10.0.0</version>
      <scope>provided</scope>
    </dependency>

  </dependencies>

  <build>

    <finalName>${project.artifactId}</finalName>

    <plugins>

      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-compiler-plugin</artifactId>
        <version>3.12.1</version>
        <configuration>
          <release>${java.version}</release>
        </configuration>
      </plugin>

      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-war-plugin</artifactId>
        <version>3.4.0</version>
      </plugin>

      <plugin>
        <groupId>org.wildfly.plugins</groupId>
        <artifactId>wildfly-maven-plugin</artifactId>
        <version>4.2.2.Final</version>
      </plugin>

    </plugins>

  </build>

</project>
```

Despliegue

Build

Para decirle a Maven que haga el *build* del archivo *war*, usaremos el siguiente comando, para limpiar restos de compilaciones anteriores y realizar la nueva de forma "limpia":

`mvn clean package`

```
usuario@david:~/practica-jakarta-wildfly$ mvn clean package
[INFO] Packaging webapp
[INFO] Assembling webapp [modulename.backend] in [/home/usuario/practica-jakarta-wildfly/target/modulename.backend]
[INFO] Processing war project
[INFO] Copying webapp resources [/home/usuario/practica-jakarta-wildfly/src/main/webapp]
[INFO] Building war: /home/usuario/practica-jakarta-wildfly/target/modulename.backend.war
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 32.933 s
[INFO] Finished at: 2025-10-31T08:40:35Z
[INFO] -----
usuario@david:~/practica-jakarta-wildfly$
```

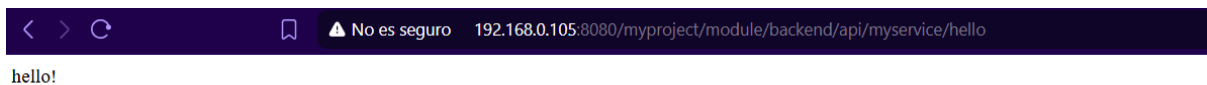
Despliegue en WildFly

Ya tenemos WildFly corriendo desde el principio de este proceso. Para realizar el despliegue, basta con meter el siguiente comando:

```
mvn wildfly::deploy
```

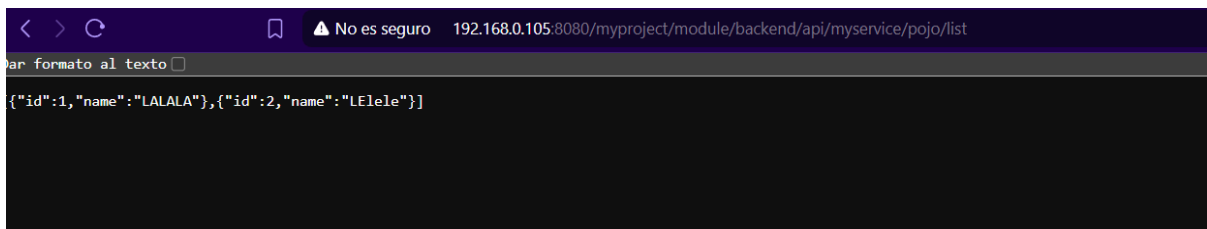
```
pero-metadata-1.1.5.Final.jar (15 kB at 5.3 kB/s)
[INFO] JBoss Threads version 2.4.0.Final
[INFO] JBoss Remoting version 5.0.27.Final
[INFO] XNIO version 3.8.9.Final
[INFO] XNIO NIO Implementation Version 3.8.9.Final
[INFO] ELY00001: WildFly Elytron version 2.2.1.Final
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 26.717 s
[INFO] Finished at: 2025-10-31T08:48:10Z
[INFO] -----
usuario@david:~/practica-jakarta-wildfly$
```

Ponemos `http://192.168.0.105:8080/myproject/module/backend/api/myservice/hello`



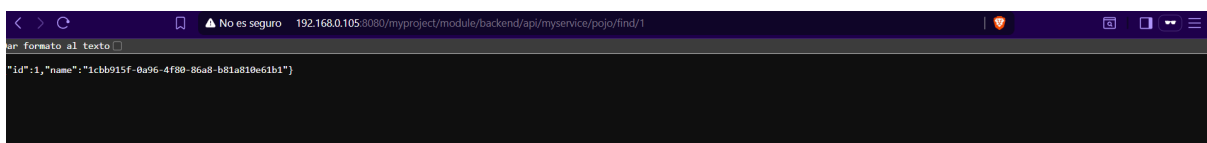
A screenshot of a web browser window. The address bar shows the URL `192.168.0.105:8080/myproject/module/backend/api/myservice/hello`. The page content displays the text `hello!`.

`http://192.168.0.105:8080/myproject/module/backend/api/myservice/pojo/list`



A screenshot of a web browser window. The address bar shows the URL `192.168.0.105:8080/myproject/module/backend/api/myservice/pojo/list`. The page content displays a JSON array: `[{"id":1,"name":"LALALA"}, {"id":2,"name":"LElele"}]`.

Comprobad que ocurre lo mismo con `/pojo/find`.



A screenshot of a web browser window. The address bar shows the URL `192.168.0.105:8080/myproject/module/backend/api/myservice/pojo/find/1`. The page content displays a JSON object: `{"id":1,"name":"1cbb915f-0a96-4f80-86a8-b81a810e61b1"}`.

Ejecuta el siguiente comando para poder ver a tiempo real el log de Wildfly:

`tail -f /opt/wildfly/standalone/log/server.log`

```
usuario@david:~/practica-jakarta-wildfly$ tail -f /opt/wildfly/standalone/log/server.log
2025-10-31 09:03:12,459 INFO [org.jboss.as.connector.subsystems.datasources] (MSC service thread 1-5) WFLYJCA0001: Bound data source [java:jboss/datasources/ExampleDS]
2025-10-31 09:03:12,511 INFO [org.jboss.ws.common.management] (MSC service thread 1-2) JBWS022052: Starting JBossWS 7.3.5.Final (Apache CXF 4.0.9.jbossorg-1)
2025-10-31 09:03:13,466 INFO [org.jboss.resteasy.resteasy_jaxrs.i18n] (ServerService Thread Pool -- 80) RESTEASY00225: Deploying jakarta.ws.rs.core.Application: class com.mycountry.myproject.module.RestApplication
2025-10-31 09:03:13,510 INFO [org.hibernate.validator.internal.util.Version] (ServerService Thread Pool -- 80) HV000001: Hibernate Validator 8.0.3.Final
2025-10-31 09:03:13,537 INFO [org.wildfly.extension.undertow] (ServerService Thread Pool -- 80) WFLYUT0021: Registered web context: '/myproject/module/backend' for server 'default-server'
2025-10-31 09:03:13,596 INFO [org.jboss.as.server] (Controller Boot Thread) WFLYSRV0010: Deployed "module-name-backend.war" (runtime-name: "module-name-backend.war")
2025-10-31 09:03:13,649 INFO [org.jboss.as.server] (Controller Boot Thread) WFLYSRV0212: Resuming server
2025-10-31 09:03:13,655 INFO [org.jboss.as] (Controller Boot Thread) WFLYSRV0060: Http management interface listening on http://0.0.0.0:9990/management
2025-10-31 09:03:13,655 INFO [org.jboss.as] (Controller Boot Thread) WFLYSRV0051: Admin console listening on http://0.0.0.0:9990
2025-10-31 09:03:13,656 INFO [org.jboss.as] (Controller Boot Thread) WFLYSRV0025: WildFly 38.0.0.Final (WildFly Core 38.0.0.Final) started in 3693ms - Started 384 of 598 services (335 services are lazy, passive or on-demand) - Server configuration file in use: standalone.xml - Minimum feature stability level: community
```

Crear nueva entrada (pon tu nombre, no el que viene en este comando):

```
curl -d '{"id": "2023", "name": "Despliegue"}' -H "Content-Type: application/json" -X POST
http://localhost:8080/myproject/module/backend/api/myservice/pojo/new
```

Actualizar una entrada (pon tu nombre, no el mío):

```
curl -d '{"id": "55", "name": "Raúl"}' -H "Content-Type: application/json" -X PUT
http://localhost:8080/myproject/module/backend/api/myservice/pojo/update
```

Eliminar una entrada

```
curl -X DELETE
http://localhost:8080/myproject/module/backend/api/myservice/pojo/remove?id=3
```

```
usuario@david:~/practica-jakarta-wildfly$ sudo curl -d '{"id": "2023", "name": "Despliegue"}' -H "Content-Type: application/json" -X POST http://localhost:8080/myproject/module/backend/api/myservice/pojo/new
usuario@david:~/practica-jakarta-wildfly$ sudo curl -d '{"id": "55", "name": "Raúl"}' -H "Content-Type: application/json" -X PUT http://localhost:8080/myproject/module/backend/api/myservice/pojo/update
usuario@david:~/practica-jakarta-wildfly$ sudo curl -X DELETE http://localhost:8080/myproject/module/backend/api/myservice/pojo/remove?id=3
```

Muestra las entradas de los logs que se corresponden con estas peticiones

```
usuario@david:~/practica-jakarta-wildfly$ tail -f /opt/wildfly/standalone/log/server.log
2025-10-31 09:03:13,510 INFO [org.hibernate.validator.internal.util.Version] (ServerService Thread Pool -- 80) HV000001: Hibernate Validator 8.0.3.Final
2025-10-31 09:03:13,537 INFO [org.wildfly.extension.undertow] (ServerService Thread Pool -- 80) WFLYUT0021: Registered web context: '/myproject/module/backend' for server 'default-server'
2025-10-31 09:03:13,596 INFO [org.jboss.as.server] (Controller Boot Thread) WFLYSRV0010: Deployed "module-name-backend.war" (runtime-name: "module-name-backend.war")
2025-10-31 09:03:13,649 INFO [org.jboss.as.server] (Controller Boot Thread) WFLYSRV0212: Resuming server
2025-10-31 09:03:13,655 INFO [org.jboss.as] (Controller Boot Thread) WFLYSRV0060: Http management interface listening on http://0.0.0.0:9990/management
2025-10-31 09:03:13,655 INFO [org.jboss.as] (Controller Boot Thread) WFLYSRV0051: Admin console listening on http://0.0.0.0:9990
2025-10-31 09:03:13,656 INFO [org.jboss.as] (Controller Boot Thread) WFLYSRV0025: WildFly 38.0.0.Final (WildFly Core 38.0.0.Final) started in 3693ms - Started 384 of 598 services (335 services are lazy, passive or on-demand) - Server configuration file in use: standalone.xml - Minimum feature stability level: community
2025-10-31 09:09:58,345 INFO [stdout] (default task-2) Creating new Pojo: pojo( id: 2023, name: Despliegue )
2025-10-31 09:10:20,847 INFO [stdout] (default task-2) Updating the Pojo: pojo( id: 55, name: Raúl )
2025-10-31 09:12:02,630 INFO [stdout] (default task-2) Removing pojo with id: 3
```

