



entando

11/8/17

GETTING STARTED



ENTANDO EXECUTIVE SUMMARY

This document describes the basics of how to get up and run a web application based on Entando v4.3.1.

TARGET AUDIENCE

This document is targeted to users who wish to explore the capabilities and functionality of the Entando Platform.

DOCUMENT DETAILS

Document Version	Product Version	Date	Description
V 1.1	4.3.1	25 October 2017	Updated to Entando 4.3.1 version

SUMMARY

- 1 Introduction To Entando 4
- 2 Environment Setup..... 5
 - 2.1 Hardware configuration 5
 - 2.2 Software Configuration 5
- 3 Run the Entando Application 6
- 4 Java Environment Setup..... 7
 - 4.1 Setup Environment with OS X 10.9 or newer 7
 - 4.2 Java Environment Setup with Linux 7
 - 4.3 Setup Environment with Windows 9

1 INTRODUCTION TO ENTANDO

Entando is the lightest, open source Digital Experience Platform (DXP) for modern applications. Entando harmonizes customer experience across the omnichannel applying the techniques of modern software practices to enterprise applications. Entando can be used to modernize UI/UX layers on top of existing applications or to build new generation applications aligned to UI/UX best practices, across different industries and use cases.

2 ENVIRONMENT SETUP

To run a web application based on Entando v.4.3.1 you need a system that meets the following hardware and software requirements.

2.1 HARDWARE CONFIGURATION

The minimum hardware requirements are

- 4 GB of RAM
- Dual core CPU ~ 2.2 Ghz
- ~ 10 Gb of free space on disk (the vast majority of the disk space is for Maven dependencies)

2.2 SOFTWARE CONFIGURATION

Entando is based on Java, so you can run it on any operating system, platform or architecture that supports a Java environment.

Here below what you need to have installed on your system to run an Entando web application

Java Oracle JDK, version ≥ 1.8 (Entando has been tested with Oracle JDK up to jdk-8u151)¹

You can easily check your installed JDK version issuing the command

```
java -version
```

Maven, version $\geq 3.5.0$ (Entando has been tested with Maven up to version v3.5.0)²

Issue this command to check your version

```
mvn -version
```

Ant, version up to 1.10.1 (if you need a development ready environment)³

NOTE: Follow “Java Environment Setup” section, if you need any help on how to get a Java environment correctly set up, depending on your operating system.

¹ <http://www.oracle.com/technetwork/java/javase/downloads/index.html>

² <http://maven.apache.org/download.cgi>

³ <http://ant.apache.org/bindownload.cgi>

3 RUN THE ENTANDO APPLICATION

With JDK and Maven installed let's start creating a new web application based on Entando bootstrap archetype. First create a new directory which will host the Entando project, move into that folder and issue the command

```
mvn archetype:generate -Dfilter=entando-archetype-portal-bootstrap
```

you will be prompt for

- **archetype:** a list of archetypes corresponding to the supplied filter;
- **version:** a list of versions, with the last one set as the default;
- **groupId:** this is generally unique amongst an organization or a project (e.g. *mycompany*);
- **artifactId:** this is generally the name of the project you are going to create (e.g. *mysample*);
- **version:** this is the version of the project that is being built;
- **package:** this is the package name where to place your project's source code.

For the archetype you will, generally, be presented with several projects' choices, you can choose any of those typing the corresponding number; for the `version`, a list will be proposed to you, just choose one, typing its number or accept the default; for the others parameters you can insert bogus data.

Maven will create a new folder named as your `artifactId` parameter entry.

To start the project using the *Jetty* server, get into the newly created folder, where the `.pom` file resides and launch it issuing the command

```
mvn clean jetty:run
```

Open your browser and point it to <http://localhost:8080/>

NOTE: Ignore the error and click on the link shown on that page.

Log into the web application with **username:** *admin* and **password:** *adminadmin*

4 JAVA ENVIRONMENT SETUP

The Java Environment Setup may vary depending on your operating system, here some tips to help you in that task.

4.1 SETUP ENVIRONMENT WITH OS X 10.9 OR NEWER

- Download the Java JDK >= 8 from the Oracle website and install the provided package
- Set the JAVA_HOME environment variable

```
$ echo "export JAVA_HOME=~/.usr/libexec/java_home" | tee -a ~/.bash_profile
```

- Install XCode from the App Store
- Install Homebrew⁴ or Macports⁵ (choose what you prefer)
- Run brew doctor and fix any warning you get from it
- Install Maven and Ant with Homebrew:
 - `$ brew install maven`
 - `$ brew install ant`

4.2 JAVA ENVIRONMENT SETUP WITH LINUX

The procedure is supposed to work with all major Linux distributions, although your mileage may vary depending on your specific distribution or settings.

Let's assume that Oracle Java is not installed, or you have OpenJDK installed on your system.

- Download `jdk-8u151-linux-x64.tar.gz` from Oracle site
- Download Apache maven `apache-maven-3.5.0-bin.tar.gz` from Apache site
- Download Apache Ant `apache-ant-1.10.1-bin.tar.gz` from Apache site

We are going to install Oracle JDK, Maven and Ant into `/opt` directory.

- copy the `jdk tarball` into it

```
sudo cp jdk-8u151-linux-x64.tar.gz /opt/
```

⁴ <https://brew.sh/>

⁵ <https://www.macports.org/>

- move into /opt and issue the following command to *untar* the packed JDK you have already copied there

```
sudo tar xvzf jdk-8u151-linux-x64.tar.gz
```

- copy the apache-maven-3.5.0-bin.tar.gz tarball into /opt and untar it
- copy the apache-ant-1.10.1-bin.tar.gz tarball into /opt and untar it

```
sudo cp apache-maven-3.5.0-bin.tar.gz /opt/
```

```
sudo cp apache-ant-1.10.1-bin.tar.gz /opt/
```

```
cd /opt/
```

```
sudo tar xvfz apache-maven-3.5.0-bin.tar.gz
```

```
sudo tar apache-ant-1.10.1-bin.tar.gz
```

- Now let's set the PATH for Java, Maven and Ant and some Environment variables, globally, we will do that adding a new file in /etc/profile.d directory.
- Create a file named java_env.sh in /etc/profile.d, with the editor of your choice and with the following content

```
PATH=$PATH:/opt/jdk1.8.0_151/bin:/opt/jdk1.8.0_151/db/bin:/opt/jdk1.8.0_151/jre/bin:/opt/apache-maven-3.5.0/bin:/opt/apache-ant-1.10.1/bin
```

```
J2SDKDIR=/opt/jdk1.8.0_151
```

```
J2REDIR=/opt/jdk1.8.0_151/jre
```

```
JAVA_HOME=/opt/jdk1.8.0_151
```

```
DERBY_HOME=/opt/jdk1.8.0_151/db
```

```
export PATH;
```

```
export J2SDKDIR;
```

```
export J2REDIR;
```

```
export JAVA_HOME;
```

```
export DERBY_HOME;
```

- Save the file, log out and log in again.

- Enter the following commands to inform the system about Java's location

```
sudo update-alternatives --install "/usr/bin/java" "java"  
"/opt/jdk1.8.0_151/bin/java" 0
```

```
sudo update-alternatives --install "/usr/bin/javac" "javac"  
"/opt/jdk1.8.0_151/bin/javac" 0
```

```
sudo update-alternatives --set java /opt/jdk1.8.0_151/bin/java
```

```
sudo update-alternatives --set javac /opt/jdk1.8.0_151/bin/javac
```

- Check that java and javac point to the previous defined paths

```
update-alternatives --list java
```

```
update-alternatives --list javac
```

or

```
update-alternatives -list for RedHat derived Linux distributions.
```

- Then check the JDK with

```
java -version
```

If you get the installed Oracle java version, you have done it right and the JDK is correctly installed.

- Also check whether Maven and Ant are available issuing the commands

```
mvn -version
```

```
ant -version
```

If you get an answer Java is now correctly set up to run an Entando project or start working with it.

4.3 SETUP ENVIRONMENT WITH WINDOWS

- Install Java JDK >= 1.8

Download and install the .exe installer [jdk-8u151-windows-x64.exe](#)

- Create the environment variable

```
JAVA_HOME -> C:\Program Files\Java\jdk1.8.0_151
```

- Install Maven

`apache-maven-3.5.0-bin`

- Set the Maven environment variable

`MVN_HOME -> C:\Program Files\apache-maven-3.5.0`

- Install Ant

`apache-ant-1.10.1-bin`

- Set the Ant environment variable

`ANT_HOME -> C:\Program Files\apache-ant-1.10.1`