Installing JHipster

Installation types

We provide 5 ways of working with JHipster:

- JHipster Online (https://start.jhipster.tech/) is a simple way to generate an application without installing JHipster in the first place.
- “Local installation with Yarn” is the classical way of working with JHipster. Everything is installed on your machine, which can be a little complex to set up, but that’s how most people usually work. In case of doubt, choose this installation.
- “Local installation with NPM” is the same as classical “local installation with Yarn”, but using NPM instead of Yarn (https://yarnpkg.com/)
- The Vagrant-based “development box (https://github.com/jhipster/jhipster-devbox)”, with all tools already set up in a Ubuntu-based virtual machine.
- The “Docker (https://www.docker.io/)” container, which brings you a lightweight, virtualized container with JHipster installed.

JHipster Online (for users wanting a simplified way to run JHipster)

JHipster Online (https://start.jhipster.tech/) allows you to easily generate JHipster applications, without having to install JHipster.

This is intended for people trying JHipster for the first time, or who just want to have a look at what JHipster provides.

While it is easier to use, it is not the “full JHipster experience”, and once your application is generated you will still need to follow most of the steps from the next section (“Local installation with Yarn”), as you will still need Java (to run your application) and Yarn (to manage your front-end code).

In the future, we expect JHipster Online to provide more features, of course.

Local installation with Yarn (recommended for normal users)

1. Install Java 8 from the Oracle website (http://www.oracle.com/technetwork/java/javase/downloads/index.html).
2. (Optional) Install a Java build tool.
   - Whether you choose to use Maven (http://maven.apache.org/) or Gradle (http://www.gradle.org/), you normally don’t have to install anything, as JHipster will automatically install the Maven Wrapper (https://github.com/takari/maven-wrapper) or the Gradle Wrapper (http://gradle.org/docs/current/userguide/gradle_wrapper.html) for you.
   - If you don’t want to use those wrappers, go to the official Maven website (http://maven.apache.org/) or Gradle website (http://www.gradle.org/) to do your own installation.
3. Install Git from git-scm.com (http://git-scm.com/). We recommend you also use a tool like SourceTree (http://www.sourcetreeapp.com/) if you are starting with Git.
4. Install Node.js from the Node.js website (http://nodejs.org/) (prefer an LTS version)
5. Install Yarn from the Yarn website (https://yarnpkg.com/en/docs/install)
6. Install Yeoman: `yarn global add yo`
7. Only for AngularJS 1, install Bower: `yarn global add bower`
8. Only for AngularJS 1, install Gulp: `yarn global add gulp-cli`
9. Install JHipster: `yarn global add generator-jhipster`

Note: if you have problems to use these tools globally, be sure you have `$HOME/.config/yarn/global/node_modules/.bin` in your path.

On Mac or Linux: `export PATH="$PATH:`yarn global bin`:$HOME/.config/yarn/global/node_modules/.bin"`

JHipster uses Yeoman (http://yeoman.io/) for code generation. To find more information, tips and help, please have a look at the Yeoman “getting starting” guide (http://yeoman.io/learning/index.html) and at the Yarn documentation (https://yarnpkg.com/) before submitting a bug (https://github.com/jhipster/generator-jhipster/issues?state=open).

Now that JHipster is installed, your next step is to create an application (http://www.jhipster.tech/creating-an-app/)

Local installation with NPM (alternative to Yarn)

1. Install Java 8 from the Oracle website (http://www.oracle.com/technetwork/java/javase/downloads/index.html).
2. (Optional) Install a Java build tool.
   - Whether you choose to use Maven (http://maven.apache.org/) or Gradle (http://www.gradle.org/), you normally don’t have to install anything, as JHipster will automatically install the Maven Wrapper (https://github.com/takari/maven-wrapper) or the Gradle Wrapper (http://gradle.org/docs/current/userguide/gradle_wrapper.html) for you.
   - If you don’t want to use those wrappers, go to the official Maven website (http://maven.apache.org/) or Gradle website (http://www.gradle.org/) to do your own installation.
3. Install Git from git-scm.com (http://git-scm.com/). We recommend you also use a tool like SourceTree (http://www.sourcetreeapp.com/) if you are starting with Git.
4. Install Node.js from the Node.js website (http://nodejs.org/) (prefer an LTS version). This will also install `npm`, which is the node package manager we are using in the next commands.
5. (Recommended) Update NPM: `npm install -g npm`
6. Install Yeoman: `npm install -g yo`
7. Only for AngularJS 1, install Bower: `npm install -g bower`
8. Only for AngularJS 1, install Gulp: `npm install -g gulp-cli` (If you have previously installed a version of gulp globally, please run `npm rm -g gulp` to make sure your old version doesn’t collide with `gulp-cli`)
9. Install JHipster: `npm install -g generator-jhipster`
10. (Optional) Install Yarn: `npm install -g yarn` (If so, after generating a project, `yarn install` will be launched instead of `npm install`)

You can find here the NPM documentation (https://docs.npmjs.com/).

Vagrant box installation

The JHipster development box (https://github.com/jhipster/jhipster-devbox) project gives you a virtual machine with all the necessary tools to develop your JHipster project.

It's an easy way to get up and running very quickly with JHipster.

Besides JHipster, this virtual machine includes many development tools, as well as Docker, so you should have everything ready for working.

Please go to the JHipster development box page (https://github.com/jhipster/jhipster-devbox) for installation and configuration information.

Docker installation (for advanced users only)

Please note: this Docker image is for running the JHipster generator inside a container. It's completely different from the Docker and Docker Compose configurations (http://www.jhipster.tech/docker-compose/) that JHipster will generate, which goal is to run your generated application inside a container

Information

JHipster has a specific Dockerfile (https://github.com/jhipster/generator-jhipster/blob/master/Dockerfile), which provides a Docker (https://www.docker.io/) image.

It makes a Docker "Automated build" that is available on: https://hub.docker.com/r/jhipster/jhipster/ (https://hub.docker.com/r/jhipster/jhipster/)

This image will allow you to run JHipster inside Docker.

Prerequisites

This depends on your operating system.

1. **Linux:** Linux supports Docker out-of-box. You just need to follow the tutorial on the Docker (https://docs.docker.com/installation/#installation) website.
2. **Mac & Windows:** install the Docker Toolbox (https://www.docker.com/docker-toolbox) to get Docker installed easily.

As the generated files are in your shared folder, they will not be deleted if you stop your Docker container. However, if you don't want Docker to keep downloading all the Maven and NPM dependencies every time you start the container, you should commit its state or mount a volume.

> *Warning:* Based on your OS, your `DOCKER_HOST` will differ. On Linux, it will be simply your localhost. For Mac/Windows, you will have to obtain the IP using following command: `docker-machine ip default`

> *Tip:* Kitematic is an easy-to-use graphical interface provided with the Docker Toolbox, which will makes this installation a lot easier.

On Linux, you might need to run the `docker` command as root user if your user is not part of docker group. It's a good idea to add your user to docker group so that you can run docker commands as a non-root user. Follow the steps on http://askubuntu.com/a/477554 (http://askubuntu.com/a/477554) to do so.

Usage on Linux/Mac Windows (using Docker Toolbox)

Pull the image

Pull the latest JHipster Docker image:

```
docker image pull jhipster/jhipster
```

Pull the development JHipster Docker image:

```
docker image pull jhipster/jhipster:master
```

You can see all tags here (https://hub.docker.com/r/jhipster/jhipster/tags/)

Run the image

> *Warning:* If you are using Docker Machine on Mac or Windows, your Docker daemon has only limited access to your OS X or Windows file system. Docker Machine tries to auto-share your /Users (OS X) or C:\Users\<username> (Windows) directory. So you have to create the project folder under these directory to avoid any volume mounting issues.

Create a "jhipster" folder in your home directory:

```
mkdir ~/jhipster
```

Run the Docker image, with the following options:

- The Docker "/home/jhipster/app" folder is shared to the local "~/jhipster" folder
- Forward all ports exposed by Docker (8080 for the Java application, 9000 for BrowserSync, 3001 for the BrowserSync UI)

```
docker container run --name jhipster -v ~/jhipster:/home/jhipster/app -v ~/.m2:/home/jhipster/.m2 -p 8080:8080 -p 9000:9000 -p 3001:3001
-d -t jhipster/jhipster
```

> *Tip:* If you have already started the container once before, you do not need to run the above command, you can simply start/stop the existing container.

Check if the container is running

To check that your container is running, use the command `docker container ps` :

```
CONTAINER ID    IMAGE               COMMAND                  CREATED         STATUS          PORTS
            NAMES
4ae16c0539a3    jhipster/jhipster   "tail -f /home/jhipst"   4 seconds ago   Up 3 seconds    0.0.0.0:9000-3001->9000-3001/tcp, 0.0.0.0:8080-
>8080/tcp    jhipster
```

Common operations

- To stop the container execute: `docker container stop jhipster`
- And to start again, execute: `docker container start jhipster`

In case you update the Docker image (rebuild or pull from the Docker hub), it's better to remove the existing container, and run the container all over again. To do so, first stop the container, remove it and then run it again:

1. `docker container stop jhipster`
2. `docker container rm jhipster`
3. `docker image pull jhipster/jhipster`
4. `docker container run --name jhipster -v ~/jhipster:/home/jhipster/app -v ~/.m2:/home/jhipster/.m2 -p 8080:8080 -p 9000:9000 -p 3001:3001 -d -t jhipster/jhipster`

Accessing the container

> *Warning:* On Windows, you need to execute the Docker Quick Terminal as Administrator to be able to create symlinks during the `yarn install` step.

The easiest way to log into the running container is by executing following command:

```
docker container exec -it <container_name> bash
```

If you copy-pasted the above command to run the container, notice that you have to specify `jhipster` as the container name:

```
docker container exec -it jhipster bash
```

You will log in as the "jhipster" user.

If you want to log in as "root", as the `sudo` command isn't available in Ubuntu Xenial, you need to run:

```
docker container exec -it --user root jhipster bash
```

Your first project

You can then go to the /home/jhipster/app directory in your container, and start building your app inside Docker:

```
cd /home/jhipster/app
```

```
jhipster
```

> *Tip:* If you are having issues with Yarn, you can use `jhipster --npm` , to use NPM instead of Yarn.

Once your application is created, you can run all the normal gulp/bower/maven commands, for example:

```
./mvnw
```

Congratulations! You've launched your JHipster app inside Docker!

On your host machine, you should be able to :

- Access the running application at `http://DOCKER_HOST:8080`
- Get all the generated files inside your shared folder

> *Warning:* By default, Docker is not installed inside the `jhipster/jhipster` image.
> So you won't be able to:
> - use the docker-compose files
> - build a Docker image (Maven goal: `dockerfile:build` or Gradle task: `buildDocker`)

Team (http://www.jhipster.tech/team/)

Artwork (http://www.jhipster.tech/artwork)

Policies (http://www.jhipster.tech/policies/)

Thanks (http://www.jhipster.tech/thanks/)

Meetups (http://www.jhipster.tech/meetups/)

Archives (http://www.jhipster.tech/documentation-archive/)

(https://github.com/jhipster/generator-jhipster)

(https://gitter.im/jhipster/generator-jhipster)

(https://twitter.com/java_hipster)