

WebFX Documentation

Table of Contents

Introduction	1
Getting started	1
Prerequisite	1
Introducing the WebFX CLI	1
Installing the WebFX CLI	2
Creating your first WebFX app	3

Introduction

Getting started

Prerequisite

- Java version 13 or above
- Maven

Introducing the WebFX CLI

The WebFX CLI is an essential tool to help you to build WebFX apps. It will create and maintain your modules and build files, such as:

- pom.xml (Maven module files)
- module.gwt.xml (GWT module files)
- module-info.java (Java module files)
- and many other files (GWT super sources, GWT bundles, Glueon builds, GraalVM conf, ...)



The WebFX CLI is required to build a WebFX application, but it is not required for its execution.

Your only inputs will be centralized in the WebFX module files (named webfx.xml), and the tool will automatically generate the rest of the build chain from these webfx.xml files, combined with an analysis of your source code. This includes all your dependencies. When a feature is platform-dependent (several implementations exist for different platforms), the tool will pick up the right modules in the final executable modules (the ones matching the platform targeted by the executable module).

Installing the WebFX CLI

As we haven't published any release at this stage yet, the way to install the WebFX CLI for now is to clone the [webfx-cli](#) repository, and build it with Maven.



We will distribute the WebFX CLI in a better way later, when we will publish the first official release.

Cloning the webfx-cli repository

```
$ git clone git@github.com:webfx-project/webfx-cli.git
```

Building webfx-cli with Maven

```
$ cd webfx-cli  
$ mvn package
```



As already mentioned, WebFX CLI requires java 13 or above (you will get a compilation error if using a lower JDK version).

This generates an executable fat jar in the target folder that we can execute with java:

```
$ java -jar target/webfx-cli-0.1.0-SNAPSHOT-fat.jar --help
```

Creating a permanent alias

To easily invoke the WebFX CLI from a terminal, we need to create a *webfx* alias to it. This is done with the following command:

Linux

```
$ echo "alias webfx='java -jar $(cd "$(dirname "$1")" && pwd -P)/$(basename "$1")/target/webfx-cli-0.1.0-SNAPSHOT-fat.jar'" >> ~/.bashrc  
$ source ~/.bashrc
```

macOS >= Catalina

```
$ echo "alias webfx='java -jar $(cd "$(dirname "$1")" && pwd -P)/$(basename "$1")/target/webfx-cli-0.1.0-SNAPSHOT-fat.jar'" >> ~/.zshrc  
$ source ~/.zshrc
```

macOS < Catalina

```
$ echo "alias webfx='java -jar $(cd "$(dirname "$1")" && pwd -P)/$(basename "$1")/target/webfx-cli-0.1.0-SNAPSHOT-fat.jar'" >> ~/.bash_profile
$ source ~/.bash_profile
```

Windows Powershell

```
$ alias webfx="java -jar /absolute-path-to-target/webfx-cli-0.1.0-SNAPSHOT-fat.jar"
```

Then you should be able to invoke the CLI tool from the terminal:

```
$ webfx --help
```

Updating the WebFX CLI to the latest version

If later you want to update the WebFX CLI to the latest version, you just need to update your local repository and rebuild it with Maven. This is done through the following commands (from your webfx-cli local folder):

```
$ git pull
$ mvn package
```

Creating your first WebFX app

Creating and initializing your repository

Let's create our first WebFX application. We need to create the repository folder and ask the WebFX CLI to initialize it, passing it the groupId, artifactId and version of our application.

```
$ mkdir my-webfxapp
$ cd my-webfxapp
$ webfx init org.example my-webfxapp 1.0.0-SNAPSHOT
```

Creating your application modules

```
$ webfx create application -c org.example.mywebfxapp.MyWebFxApplication -w
```

Updating your modules

```
webfx update
```