

Simple JBang Hello.java

```
#!/usr/bin/env jbang "$0" "$@" ; exit $?
class Hello {
    public static void main(String[] args) {
        System.out.println("Hello
World! ");
    }
}
```

Run JBang script

```
$ jbang Hello.java
$ jbang --debug Hello.java
$ chmod u+x
$ ./Hello.java
```

Templates

```
$ jbang template list
$ jbang init --template=cli helloworld.java
```

ChatGPT - init script from ChatGPT

```
$ jbang init HelloCli.java "generate hello command
with picocli library"
```

You need an OPENAI_API_KEY environment variable.

Editing

```
$ jbang edit helloworld.java
$ jbang edit --open=code helloworld.java
$ jbang edit --open=idea helloworld.java
```

Assign Java version

//JAVA 8	Assigned version in Script
//JAVA 11+	Java 11 or higher
jbang --java 8 Hello.java	Assigned version from CLI
jbang jdk list --available	List available JDKs

If you don't assign Java version, and JBang will run the script with Java 17 by default.

java and javac Options

```
//JAVAC_OPTIONS -parameters
//JAVA_OPTIONS -Dnick=xxx
//JAVA_OPTIONS -Dserver.port=8081 -D
spring.application.name=appl
//PREVIEW
```

Java preview

```
#!/usr/bin/env jbang "$0" "$@" ; exit $?
//PREVIEW
//JAVA 21
void main() {
    System.out.println("Hello World");
}
```

Multiple Java source files

```
//SOURCES UserService.java
//SOURCES impl/UserServiceImpl.java
```

Multi resource files - properties/xml etc

```
//FILES application.properties
//FILES META-INF/resources/index.html= index.html
//FILES META-INF/resources/jquery.js= jquery-3.6.0.min.js
//FILES templates/HelloWorld.txt= template.txt
//FILES resources/HelloWorld.txt= template.txt
Resource import format: //FILES <module> <resource> [=
<sourcefile>]
```

Polyglot - Kotlin

```
#!/usr/bin/env jbang "$0" "$@" ; exit $?
//KOTLIN 1.8.21
//DEPS org.jetbrains.kotlin:kotlin-stdlib:1.8.21@pom
//DEPS org.jetbrains.kotlin:kotlin-stdlib-jdk8:1.8.21@pom
public fun main() {
    println("Hello World");
}
```



Polyglot - Groovy

```
///usr/bin/env jbang "$0" "$@" ; exit $?
//GROOVY 4.0.12
println("Hello World");
```

Polyglot - JShell

```
///usr/bin/env jbang "$0" "$@" ; exit $?
//DEPS net.dafny.dafny:1.9.0
import net.dafny.dafny.Faker;
Faker faker = new Faker();
```

\$ jbang demo.jsh

\$ jbang --interactive demo.jsh

JBang native

```
$ jbang --native Hello.java
// https://github.com/linux-china/jbang-ang -sp ring-native
// https://github.com/Godofrigo/gralv -m-hint
```

Dependencies

```
//DEPS systems.manifold:manifold-all:2023.1.10
//DEPS org.projectlombok:lombok:1.18.28
//DEPS org.slf4j:slf4j-simple:2.0.7
//DEPS libs/loca.jar
//DEPS eu.maveniverse.maven.plugin:box:0.1.9:cli@fatjar
```

fatjar: all its dependencies included and thus not require fetching declared dependencies.

Managed dependencies by BOM - Spring Boot

```
//DEPS org.springframework.boot:spring-boot-dependencies:2.7.11@pom
//DEPS org.springframework.boot:spring-boot-starter
```

Managed dependencies by BOM - Quarkus

```
//DEPS io.quarkus:quarkus-bom:2.9.2.Final@pom
//DEPS io.quarkus:quarkus-resteasy
```

Repositories

```
//REPOS central,spring-
release=https://repo.spring.io/release
//REPOS central,spring-milestone=https://repo.spring.io/milestone
```

Well-known repositories

```
* central - https://repo1.maven.org/maven2/
* google - https://maven.google.com/
* jitpack - https://jitpack.io/
```

Aliases

```
$ jbang alias add --name hello
https://github.com/jbangdev/jbang-examples/blob/HEAD/examples/helloworld.java
$ jbang hello
$ jbang alias list
$ jbang alias list --show -origin
```

Catalogs - jbang-catalog.json

```
{
  "aliases": {
    "hello": {
      "script-ref": "hello.java",
      "description": "Script that says hello back for each argument"
    }
  }
}
```

JSON Schema for jbang-catalog.json <https://github.com/jbangdev/jbang-idea/blob/main/src/main/resources/jbang-catalog-schema.json>

Catalogs - local jbang-catalogs.json

1. jbang-catalog.json in current directory
2. jbang-catalog.json in parent directory until the root directory
3. \$HOME/.jbang/jbang-catalog.json
4. JBangHub: <https://github.com/jbanghub>

Catalogs - implicit catalog from GitHub

jbang hello@acme	hello alias in acme/jbang-catalog/jbang-catalog.json
jbang hello@acme/mycatalog	hello alias in acme/mycatalog/jbang-catalog.json
jbang hello@acme/mycatalog/dev	hello alias in acme/mycatalog/jbang-catalog.json in dev branch
jbang hello@acme~experimental	hello alias in acme/jbang-catalog/experimental/jbang-catalog.json
jbang hello@example.com	hello alias found in https://example.com/jbang-catalog.json

Catalogs - management

```
$ jbang catalog add --name acme acme
$ jbang catalog add --name mycatalog acme/mycatalog
$ jbang catalog list
```

Tips

jbang --fresh	Use none-cached resources
jbang info tools Hello.java	Prints a json description of JBang script

Apps - setup jbang to put scripts into your PATH

```
$ jbang app install <scriptRef>
$ $ jbang app install --name app1 <scriptRef>
$ jbang app list
$ jbang app uninstall <name>
```

build.java

```
#!/usr/bin/env jbang "$@" ; exit $?
//JAVA 17
//DEPS info.picocli:picocli:4.7.3
//SOURCES src
//FILES resources
```

JBang export - generate jar or native binary

jbang export fatjar	Exports an executable fat jar with all dependencies
jbang export portable	Exports jar together with dependencies in pom.xml
jbang export native	Exports native executable
jbang export mavenrepo	Exports directory that can be used to publish as a maven repository
jbang export jlink	Export app with Java runtime embedded

JBang with ChatGPT

```
export OPENAI_API_KEY=sk-xxxxxxx-x-xxxxxx
jbang init gh.java "connect to github and fetch all repositories under jbangdev and collect how many stars they have"
```

JBang environment variables

JBANG_DEFAULT_JAVA_VERSION	11
JBANG_DOWNLOAD_SOURCES	true
JBANG_EDITOR	Code editor, such as zed, code

