The Power Of Graal

Using JavaScript, Python, Ruby and WebAssembly in your Java Apps

Java Vienna - Dominik Dorn

What is GraalVM?

A high-performance JDK with polyglot capabilities

Why Use GraalVM?

— Run JavaScript, Python, Ruby, and WebAssembly inside Java

— Call Java from these languages

— Performance optimizations with Graal compiler

— Native Image support for fast startup times

Why Use GraalVM?

— you are probably using it yet already!

— IntelliJ Idea http client uses it!

Shoutout to today's audience

— Gergö Barany - native image wizardry

— Christian Humer - Truffle Framework Lead

— Matthias N. - Security guy.

Prerequisites

- GraalVM JDK
  - https://www.graalvm.org/downloads/
- NodeJS
  - https://github.com/oracle/graaljs/releases/

Maven Settings

```
<properties>
    <maven.compiler.source>23</maven.compiler.source>
    <maven.compiler.target>23</maven.compiler.target>
    <maven.compiler.release>23</maven.compiler.release>
    <native.maven.plugin.version>0.10.4</native.maven.plugin.version>
</properties>
```

Base maven dependency

```xml
<dependency>
    <groupId>org.graalvm.polyglot</groupId>
    <artifactId>polyglot</artifactId>
    <version>24.1.2</version>
</dependency>
```

Javascript support

```
<dependency>
    <groupId>org.graalvm.polyglot</groupId>
    <artifactId>js-community</artifactId>
    <version>24.1.2</version>
    <type>pom</type>
    <scope>runtime</scope>
</dependency>
```

Python support

```
<dependency>
    <groupId>org.graalvm.polyglot</groupId>
    <artifactId>python-community</artifactId>
    <version>24.1.2</version>
    <type>pom</type>
    <scope>runtime</scope>
</dependency>
```

Ruby Support

```
<dependency>
    <groupId>org.graalvm.polyglot</groupId>
    <artifactId>ruby-community</artifactId>
    <version>24.1.2</version>
    <type>pom</type>
    <scope>runtime</scope>
</dependency>
```

WASM Support

```
<dependency>
    <groupId>org.graalvm.polyglot</groupId>
    <artifactId>wasm-community</artifactId>
    <version>24.1.2</version>
    <type>pom</type>
    <scope>runtime</scope>
</dependency>
```

LLVM Support

```
<dependency>
    <groupId>org.graalvm.polyglot</groupId>
    <artifactId>llvm-community</artifactId>
    <version>24.1.2</version>
    <type>pom</type>
    <scope>runtime</scope>
</dependency>
```

Java in Java (Espresso) Support

```
<dependency>
    <groupId>org.graalvm.polyglot</groupId>
    <artifactId>java-community</artifactId>
    <version>24.1.2</version>
    <type>pom</type>
    <scope>runtime</scope>
</dependency>
```

Java Module System any1?

```
module-info.java
java
module com.mycompany.app {
  requires org.graalvm.polyglot;
}
```

Demo usage from Java

NodeJS Integration

- https://github.com/oracle/graaljs/releases/
- download graalnodejs-jvm-24.1.2-macos-aarch64.tar.gz
- important on Mac:
  - xattr -d com.apple.Quarantine graalnodejs-jvm-24.1.2-macos-aarch64.tar.gz
  - then unarchive

NodeJS Integration

```
export JAVA_HOME=$HOME/.sdkman/candidates/java/graalnodejs-24.1.2-macos-aarch64/jvm
export PATH=$HOME/.sdkman/candidates/java/graalnodejs-24.1.2-macos-aarch64/bin:$PATH
NPM_CONFIG_USERCONFIG=`pwd`/.npmrc
```

NodeJS Integration

— am besten eigenes prefix + cache anlegen, z.b. `.npmrc`:

```
prefix=/Users/domdorn/jsug/
2025-01-27_graalvm_polyglot/.node/
prefix
cache=/Users/domdorn/jsug/
2025-01-27_graalvm_polyglot/.node/cache
```

Demo usage from JS/NodeJs

Links

- https://www.graalvm.org/jdk23/reference-manual/wasm/guides/embed-c-in-java/
- https://www.graalvm.org/jdk23/reference-manual/embed-languages/
- https://www.graalvm.org/latest/reference-manual/js/NodeJS/

Further reasearch (for me)

- how to pass a `Map<String, Object>` or `Map<String, String>` in a way serialization works in guest lang?
- migrating CLI apps
  - how to pass `InputStream`/`OutputStream` -> websockets?
- explore Micronaut + React ServerSideRendering

What's Next?

Start experimenting with GraalVM today!

Thanks

for all the 🐟 😊

Questions?

Twitter/X: x.com/domdorn
LinkedIn: linkedin.com/in/dominik-dorn/