Setting up a Flowable development environment

Setting up a Flowable development environment involves installing the required software, setting up a database, and configuring Flowable inside an IDE like **IntelliJ IDEA** or **Eclipse**. Below is a step-by-step guide to get started.

---

1. Prerequisites

Before setting up Flowable, ensure you have the following installed:

- ✅ **Java Development Kit (JDK 17 or later)**
- ✅ **Apache Maven (if using Maven projects)**
- ✅ **Gradle (if using Gradle projects)**
- ✅ **Spring Boot (if using Spring Boot integration)**
- ✅ **An IDE (IntelliJ IDEA, Eclipse, or VS Code)**
- ✅ **Docker (optional, for running Flowable UI & databases easily)**

---

2. Install Java Development Kit (JDK)

Flowable requires **JDK 17+**. You can download and install it from:

- 🔗 OpenJDK Downloads
- 🔗 Oracle JDK Downloads

Verify installation:

java -version

Expected output:

java version "17.0.2" 2023-01-17 LTS

---

3. Install Apache Maven

If you're using **Maven**, install it from Maven Downloads.

Verify installation:

mvn -version

Expected output:

Apache Maven 3.x.x

For **Gradle**, install it from Gradle Downloads and check:

gradle -v

---

4. Set Up a Database

Flowable supports multiple databases (**H2, MySQL, PostgreSQL, Oracle, MSSQL**).

For local development, use **H2 (in-memory database)**. If you want a persistent database, install **PostgreSQL or MySQL**.

Option 1: H2 (In-memory, Default)

No setup required. Flowable will use **H2** automatically when running inside Spring Boot.

Option 2: MySQL (Recommended for Development)

1. Install MySQL:

   ```
   sudo apt install mysql-server  # Linux
   brew install mysql             # Mac
   ```

2. Create a Flowable database:

   ```
   CREATE DATABASE flowable;
   CREATE USER 'flowable'@'localhost' IDENTIFIED BY 'flowable';
   GRANT ALL PRIVILEGES ON flowable.* TO 'flowable'@'localhost';
   FLUSH PRIVILEGES;
   ```

3. Add the MySQL JDBC driver to your **pom.xml**:

   ```
   <dependency>
     <groupId>mysql</groupId>
     <artifactId>mysql-connector-java</artifactId>
     <scope>runtime</scope>
   </dependency>
   ```

4. Configure **application.properties**:

   ```
   spring.datasource.url=jdbc:mysql://localhost:3306/flowable
   spring.datasource.username=flowable
   spring.datasource.password=flowable
   spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
   spring.jpa.database-platform=org.hibernate.dialect.MySQL8Dialect
   ```

---

5. Download Flowable Components

You can download Flowable's UI and Engine from:

🔗 Flowable Downloads

Download:

- **Flowable Engine**
- **Flowable UI (Modeler, IDM, Task, Admin)**
- **Flowable REST API**

Extract the files into a working directory.

---

6. Running Flowable UI with Docker (Optional)

Instead of installing Flowable manually, you can run the Flowable UI apps in **Docker**.

Run Flowable in Docker

docker run -d --name flowable-ui -p 8080:8080 flowable/all-in-one

After a few seconds, open:

- **Flowable Modeler:** http://localhost:8080/flowable-modeler
- **Flowable Task:** http://localhost:8080/flowable-task
- **Flowable Admin:** http://localhost:8080/flowable-admin

Default Credentials:

- **Username:** admin
- **Password:** test

---

7. Setting Up Flowable in a Spring Boot Project

Step 1: Create a Spring Boot Project

Use **Spring Initializr** (start.spring.io) and select:

- **Spring Boot Version:** 3.x (latest)
- **Dependencies:**
    - Spring Web
    - Spring Data JPA
    - Flowable
    - H2 Database (or MySQL/PostgreSQL)
    - Lombok (optional)

Download and extract the project.

Step 2: Add Flowable Dependencies

Modify pom.xml:

<dependencies>

```
    <!-- Spring Boot Starter Web -->
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-web</artifactId>
    </dependency>

    <!-- Flowable Spring Boot Starter -->
    <dependency>
        <groupId>org.flowable</groupId>
        <artifactId>flowable-spring-boot-starter</artifactId>
    </dependency>

    <!-- H2 Database -->
    <dependency>
        <groupId>com.h2database</groupId>
        <artifactId>h2</artifactId>
        <scope>runtime</scope>
    </dependency>
</dependencies>
```

Step 3: Configure Flowable

Modify src/main/resources/application.properties:

```
# H2 In-Memory Database
spring.datasource.url=jdbc:h2:mem:flowable
spring.datasource.driverClassName=org.h2.Driver
spring.datasource.username=sa
spring.datasource.password=
spring.jpa.database-platform=org.hibernate.dialect.H2Dialect

# Flowable Configurations
flowable.database-schema-update=true
flowable.async-executor-activate=true
```

---

8. Creating a Simple BPMN Process

Create a new BPMN process inside src/main/resources/processes/:

my-process.bpmn20.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions xmlns="http://www.omg.org/spec/BPMN/20100524/MODEL"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.omg.org/spec/BPMN/20100524/MODEL
BPMN20.xsd">

  <process id="myProcess" name="Simple Process" isExecutable="true">
    <startEvent id="startEvent" />
    <sequenceFlow id="flow1" sourceRef="startEvent" targetRef="serviceTask" />
    <serviceTask id="serviceTask" name="My Task"
flowable:class="com.example.flowable.MyServiceTask"/>
    <sequenceFlow id="flow2" sourceRef="serviceTask" targetRef="endEvent" />
    <endEvent id="endEvent" />
  </process>
</definitions>
```

---

9. Implementing a Service Task

Create a class that executes logic inside the process.

```
package com.example.flowable;

import org.flowable.engine.delegate.DelegateExecution;
import org.flowable.engine.delegate.JavaDelegate;
import org.springframework.stereotype.Component;

@Component
public class MyServiceTask implements JavaDelegate {
  @Override
  public void execute(DelegateExecution execution) {
```

```
        System.out.println("Executing Flowable Service Task...");
    }
}
```

---

10. Running the Application

Run the Spring Boot application:

```
mvn spring-boot:run
```

Test the process via REST API:

```
curl -X POST http://localhost:8080/processes/start
```