Development Documentation

This document explains how to work on the project, including details about the code, how to build it, and set up your computer for development.

Table of Contents

1. Javadocs

2. Source Code Organization

3. Building the Project

4. Dependencies Needed to Compile

5. Rules for Writing Code

6. Setting Up the Database

7. Getting the Code from GitHub

1. Javadocs

Javadocs are detailed explanations of how the code works. They help you understand what each part of the code does.

To make Javadocs:

Type this in your terminal:

```

javadoc -d docs $(find src -name "\*.java")

```

This will create a folder named `docs` with all the explanations.

2. Source Code Organization

The code is organized like this:

```

project-root/

│

├── src/

│ ├── main/

│ │ ├── java/

│ │ │ └── com/

│ │ │ └── example/

│ │ │ └── project/

│ │ │ ├── MainClass.java

│ │ │ └── utils/

│ │ │ └── Helper.java

│ │ └── resources/

│ │ └── application.properties

│ └── test/

│ └── java/

│ └── com/

│ └── example/

│ └── project/

│ ├── MainClassTest.java

│ └── utils/

│ └── HelperTest.java

│

├── build/

│

├── docs/

│

├── lib/

│

└── README.md

```

This structure shows where each part of the code is located.

3. Building the Project

Before you start, make sure you have these installed:

- Java Development Kit (JDK) 8 or newer

- Apache Maven 3.6.0 or newer

Steps to build the project:

1. Open your terminal and go to the project folder:

```

cd project-root

```

2. Type these commands to compile and package the project:

```

mvn clean compile

mvn package

```

After this, you'll see the compiled files in a folder called "target".

4. Dependencies Needed to Compile

The project uses certain tools to work correctly. These are listed in a file called `pom.xml`.

5. Rules for Writing Code

When writing code, follow these rules:

- Style: Use the Google Java Style Guide.

- Version Control: Use Git and follow the Git Flow method.

- Testing: Always test your code with JUnit.

- Documentation: Explain all public parts of your code using Javadocs.

6. Setting Up the Database

Before you start, make sure you have a database like MySQL or PostgreSQL.

Steps to set up the database:

1. Create a new database:

```

CREATE DATABASE project\_db;

```

2. Set up the database structure:

```

mysql -u username -p project\_db < schema.sql

```

3. Update the file called `application.properties` with your database information:

```

spring.datasource.url=jdbc:mysql://localhost:3306/project\_db

spring.datasource.username=db\_username

spring.datasource.password=db\_password

```

7. Getting the Code from GitHub

To get the code for your project, follow these steps:

1. Open your terminal and type:

```

git clone https://github.com/yourusername/yourproject.git

```

This will copy the project from GitHub to your computer.

2. Go into the project folder:

```

cd yourproject

```

3. To get any new changes others have made, type:

```

git pull origin main

```

By following this guide, you can set up your computer for development, understand how the code is organized, build the project, and follow the rules for writing good code.