Java Language Server and Maven Build

It sounds like you're facing two related issues in Visual Studio Code (VSCode): a complaint about needing at least Java 21, and problems with VSCode not recognizing some object definitions in your Java code, even though your Maven build compiles successfully. Let's resolve this step by step.

The issue arises because VSCode's Java extensions (specifically the Java Language Server) and your Maven build might not be aligned in terms of the Java Development Kit (JDK) version or project configuration. VSCode is flagging that it requires JDK 21 to function properly, and the parsing errors suggest it's not interpreting your code the same way Maven does. Here's how to fix it:

Step 1: Install JDK 21

Since VSCode is explicitly asking for Java 21, you'll need to ensure it's installed on your system.

- **Download JDK 21**: Get it from a trusted source like Oracle's website or Adoptium.
- **Install it**: Follow the installation instructions and note the installation path (e.g., `C:\Program Files\Java\jdk-21` on Windows or `/usr/lib/jvm/java-21-openjdk` on Linux).
- **Verify**: Open a terminal and run `java -version`. You should see something like:

  ```
  java version "21.0.1" ...
  ```

Step 2: Configure VSCode to Use JDK 21

You need to tell VSCode's Java extensions to use JDK 21. Here's how:

1. **Open VSCode Settings**:
   - Go to `File > Preferences > Settings` (or press `Ctrl + ,`).
2. **Set the JDK Path**:
   - Search for `java.jdt.ls.java.home`.
   - Enter the path to your JDK 21 installation (e.g., `C:\Program Files\Java\jdk-21`).
   - Optionally, search for `java.home` and set it to the same path for consistency.
3. **Edit `settings.json` (Alternative)**:
   - Open the Command Palette (`Ctrl + Shift + P`), type "Preferences: Open Settings (JSON)", and add:

```
{
    "java.jdt.ls.java.home": "C:\\Program Files\\Java\\jdk-21",
    "java.home": "C:\\Program Files\\Java\\jdk-21"
}
```

- Adjust the path based on your system (use forward slashes / for Linux/Mac).

This ensures the Java Language Server in VSCode uses JDK 21, meeting the minimum requirement.

Step 3: Set the Project's JDK in VSCode

To fix the parsing issues (e.g., object definitions not being found), ensure your project in VSCode uses JDK 21 as well:

- In `settings.json`, add the following to specify JDK 21 for your project:

```
"java.configuration.runtimes": [
    {
        "name": "JavaSE-21",
        "path": "C:\\Program Files\\Java\\jdk-21",
        "default": true
    }
]
```

- This aligns VSCode's project runtime with JDK 21, which should help it parse your code correctly.

Step 4: Verify Maven's JDK Configuration

Since your Maven build (`mvn compile`) works fine, it's likely using a compatible JDK. Let's confirm and align it with VSCode:

1. **Check Maven's JDK**:
    - Run `mvn -version` in your terminal. Look for the "Java version"line (e.g., `Java version: 21.0.1`).
    - If it's not using JDK 21, set the `JAVA_HOME` environment variable to your JDK 21 path:
        - **Windows**: `set JAVA_HOME=C:\Program Files\Java\jdk-21`
        - **Linux/Mac**: `export JAVA_HOME=/usr/lib/jvm/java-21-openjdk`
    - Run `mvn -version` again to confirm.
2. **Specify JDK in `pom.xml`** (optional):

- If you want to enforce JDK 21 in your Maven project, add this to your `pom.xml`:

  ```
  <properties>
      <maven.compiler.source>21</maven.compiler.source>
      <maven.compiler.target>21</maven.compiler.target>
  </properties>
  ```

Step 5: Refresh VSCode

After making these changes, VSCode needs a refresh to apply them:

- Reload VSCode:
  - Open the Command Palette (`Ctrl + Shift + P`) and select “Developer: Reload Window”.
- Alternatively, close and reopen VSCode.

Step 6: Ensure Dependencies Are Loaded

If VSCode still can’t find object definitions, it might not have loaded your Maven dependencies properly:

1. **Clean the Workspace**:
   - Open the Command Palette (`Ctrl + Shift + P`).
   - Run “Java: Clean Java Language Server Workspace”.
2. **Force Compilation**:
   - Run “Java: Force Java Compilation”from the Command Palette.
3. **Check Imports**:
   - Ensure your project’s Maven dependencies are imported by right-clicking your `pom.xml` in VSCode and selecting “Update Project Configuration”.

Troubleshooting

If the problem persists: - **Check Logs**: - Add `"java.trace.server": "verbose"` to `settings.json`. - Open the Output panel (`Ctrl + Shift + U`) and select “Java Language Server”to view detailed logs. - **Verify Code**: - Double-check that the missing object definitions are from your code or dependencies listed in `pom.xml`, not external issues.

Summary

By installing JDK 21, configuring VSCode to use it, aligning your project and Maven settings, and refreshing the environment, VSCode should stop complaining about Java 21 and correctly parse your code, matching Maven’s successful compilation. Let me know if you hit any snags!