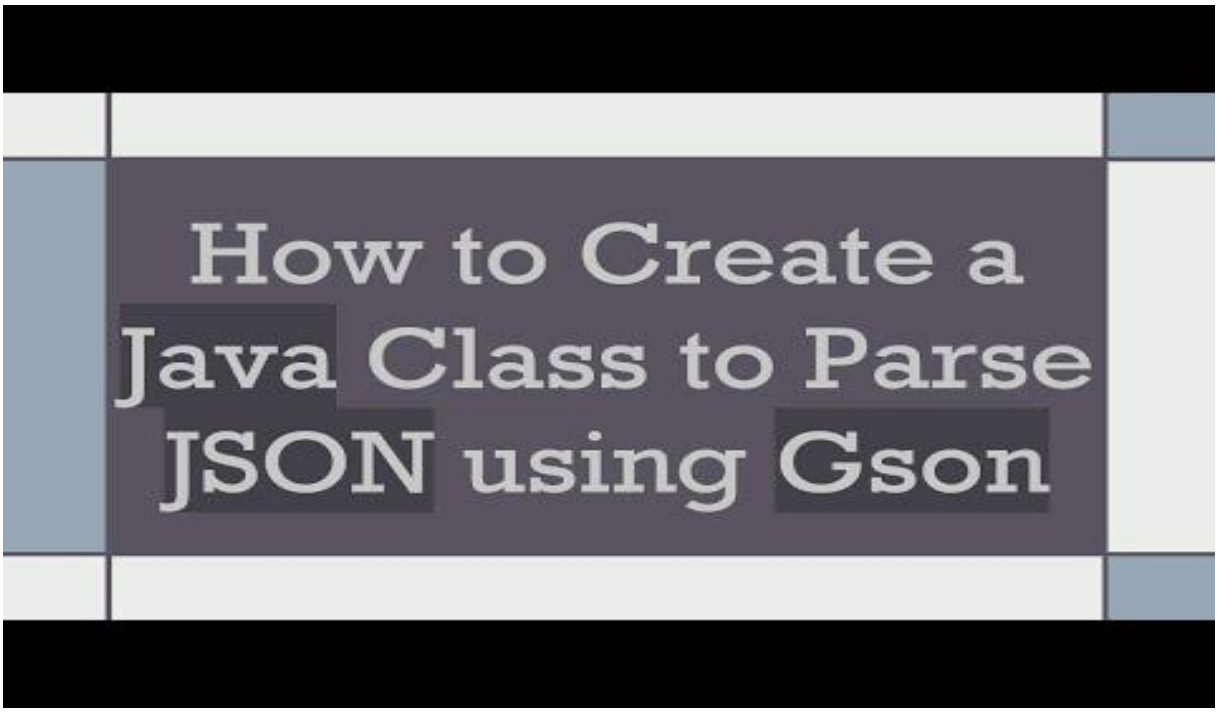
“Java Gson library and include it in your project. You might also need to look up some other examples of how to use the Gson library to parse JSON files in Java.”

Stream API in Java

- first used ArrayList and loops for filtering and sorting, but this became more complicated as the project grew. I later changed those sections using Streams to simplify the logic and improve readability.

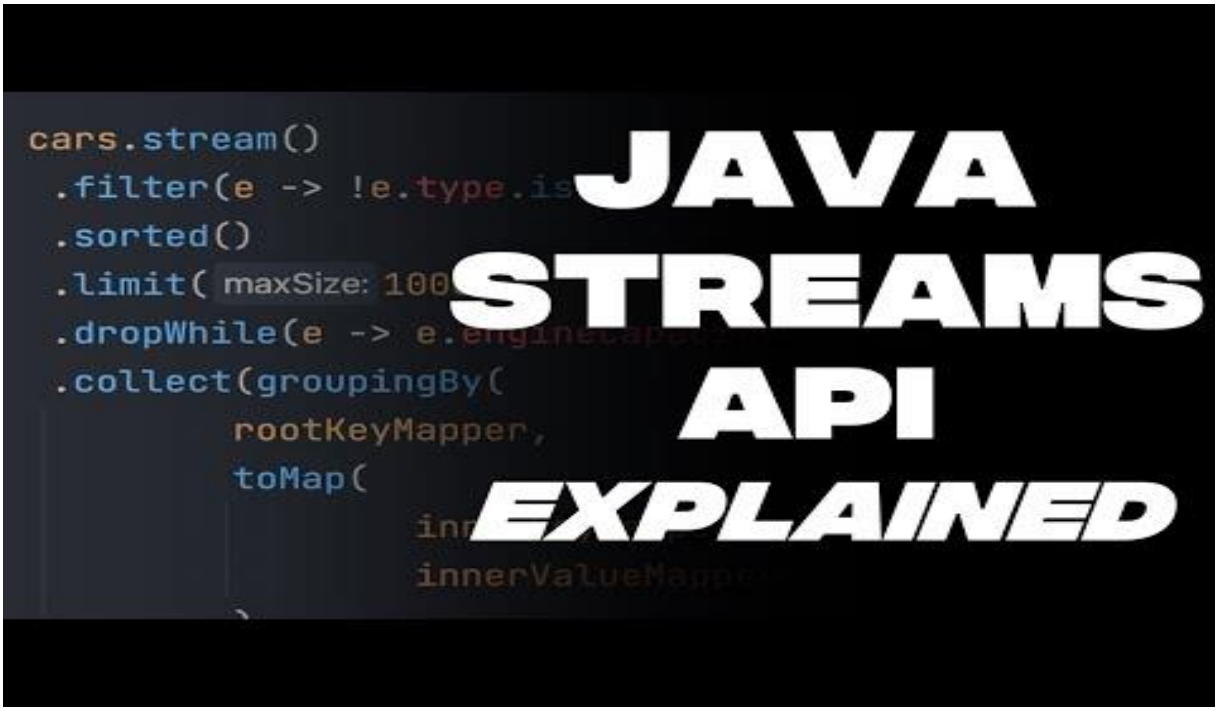
Gson Tutorial — Getting Started with Java-JSON Serialization & Deserialization

How to Create a Java Class to Parse JSON using Gson\



#99 Map Filter Reduce Sorted in Java

Java Streams API Explained (with examples) - YouTube



What I used AI for

I used AI as a troubleshooting and learning tool while setting up my Java project. I used it to:

- Understand why my code could not import the Gson library (import com.google.gson.Gson;)
- Learn how to correctly add a .jar dependency in a plain VS Code Java project
- Learn the correct way to compile and run Java files using a classpath when an external library is required
- Debug common Java/VS Code errors (missing semicolons, wrong method calls, and “red underline” issues)

The main problem I ran into (Gson not found)

At first, VS Code showed an error like:

package com.google.gson does not exist

Even though I had the correct import line.
This happened because:

- Gson is not built into Java and must be added as an external library (.jar)
- VS Code sometimes does not automatically recognize external jars unless the project is configured correctly
- I also had some syntax errors in my code at the same time, which made VS Code show extra red errors

What I did to fix it

1. I downloaded the correct Gson jar file:
    a. gson-2.10.1.jar
2. I created a lib folder inside my Final Project folder and placed the jar here:
    a. Final Project/lib/gson-2.10.1.jar
3. Even when VS Code still showed the import as red, I verified Gson was working by compiling/running from the terminal using the classpath.

How I run the program now (working method)

Because the project uses an external jar (Gson), I compile and run it using these commands from inside the Final Project folder:
Compile:
javac -cp ".;lib\gson-2.10.1.jar" *.java
Run (example main class):
java -cp ".;lib\gson-2.10.1.jar" MyMeteor
This successfully runs my program and displays the menu.

What I learned from the errors

- VS Code's “red underline” doesn't always mean the program can't run (sometimes it’s an IDE configuration issue).
- External libraries in Java require the jar to be placed correctly and included on the classpath.
- Java run commands use the class name, not the file name (example: MyMeteor, not MyMeteor.java).
- Small syntax mistakes can cause many confusing errors, so I now fix compiler errors first before assuming the library is missing.

Result

After adding Gson and running with the correct classpath, the program compiles and runs successfully, displaying the NASA Meteorite Database menu and accepting user input.