

Title: Maven + Spring Boot: Found multiple occurrences of org.json.JSONObject on the class path:

Post Body:

When I run `mvn test` I get this warning. How can I fix it?

Found multiple occurrences of `org.json.JSONObject` on the class path: `jar:file:/C:/Users/Chloe/.m2/repository/org/json/`

Here is my [pom.xml](#). The only reference to JSON is

```
<!-- https://mvnrepository.com/artifact/org.json/json -->
<dependency>
    <groupId>org.json</groupId>
    <artifactId>json</artifactId>
    <version>1.0.0</version>
</dependency>
```

Apache Maven 3.5.3

Accepted Answer:

Add under

```
<artifactId>spring-boot-starter-test</artifactId>
<scope>test</scope>
```

The following exclusion:

```
<scope>test</scope>
<exclusions>
    <exclusion>
        <groupId>com.vaadin.external.google</groupId>
        <artifactId>android-json</artifactId>
    </exclusion>
</exclusions>
```

Similarly, for Gradle projects:

```
testCompile('org.springframework.boot:spring-boot-starter-test') {
    exclude group: 'com.vaadin.external.google', module: 'android-json'
}
```

Highest Rated Answer:

**Background:** `org.json` works great, but has a license clause that some people don't like ('The Software shall be used for Good, not Evil.'). So Vaadin wanted to use the library, but couldn't be sure they wouldn't use it for evil someday. Instead, they re-implemented the interface, published `android-json` and used it as a drop in replacement for `org.json`. Others began to use `android-json` as well so that they too would not be bound by the requirement of not using their software for evil.

This is a fine solution, except that when the two libraries are on the classpath, they collide.

**Solution:** If you get this error from conflicting transitive dependencies, then your best bet is to exclude either Vaadin's `android-json` library (brought in by Spring), or exclude the `org.json` library (brought in by another dependency). Vaadin's version is meant to be an identical implementation, but there are subtle differences.

If you're using `org.json` in your code and it is conflicting with Spring's Vaadin dependency, then I would recommend trying `open-json`. It's a port of Vaadin's re-implementation of `org.json`, but they changed the packages so you won't have any conflicts with `org.json:json` or `com.vaadin.external.google:android-json`

<https://github.com/openjson/openjson>

Add gradle dependency:

```
implementation('com.github.openjson:openjson:1.0.12')
```

Or in Maven:

```
<dependency>
    <groupId>com.github.openjson</groupId>
    <artifactId>openjson</artifactId>
    <version>1.0.12</version>
</dependency>
```

Then update any imports that were being used by `org.json` classes.