Title: What is the spring-boot-configuration-processor ? Why do people exclude libraries from it? Why is it invisible in dependency tree?

Post Body:

Introduction

So I noticed the following line in the gradle file of the jhipster project:

```
annotationProcessor ('org.springframework.boot:spring-boot-configuration-processor') {         exclude group: 'com.vaadin.exte
```

https://github.com/jhipster/jhipster-sample-app-gradle/blob/9e9c3db8f3bedba4b1efd85ecb6ff3f12a5f596a/build.gradle#L230

We also used the same configuration in Maven for another project to solve the following problem: Maven transient dependency (library/jar vaadin json) is not being excluded

Questions

And now I have the following questions:

- What does the spring-boot-configuration-processor dependency do?
- Why is it necessary to sometimes exclude dependencies from the processor?
- Why doesn't the processor necessarily appear in the mvn-dependency tree?
- Why are exclusions used with processor in situations where it's very difficult to exclude a dependency?

Accepted Answer:

`spring-boot-configuration-processor` is an annotation processor that generates metadata about classes in your application that are annotated with `@ConfigurationProperties`. This metadata is used by your IDE (Eclipse, IntelliJ, or NetBeans) to provide auto-completion and documentation for the properties when editing `application.properties` and `application.yaml` files. You can learn a bit more about it in the relevant section of Spring Boot's reference documentation.

Since Spring Boot 1.5.10, the exclusion is no longer necessary as `com.vaadin.external.google:android-json` is no longer a dependency of `spring-boot-configuration-processor`.

Highest Rated Answer:

> What does the spring-boot-configuration-processor dependency do?

It scans the libraries in the build and sees what properties they use so as to inform the IDE

> Why is it necessary to sometimes exclude dependencies from the processor?

Maven libraries can clash sometimes - the one you reference was excluded by JHipster because it led to errors when on the classpath together with another library in JHipster's dependencies

> Why doesn't the processor necessarily appear in the mvn dependency:tree?

It does for me on the `jhipster-sample-app`. Presumably you're referring to the comment on the linked issue stating that the `android-json` library isn't in the tree. I've asked there about that.

> Why are exclusions used with processor in situations where it's very difficult to exclude a dependency?

This is a dependency clash issue like any other really, it just happens that the processor is bringing in the key dependency (or rather was, as @Andy Wilkinson points out `com.vaadin.external.google:android-json` is no longer used by the processor)