What is scope under dependency in pom.xml for?

What is <scope> under <dependency> in pom.xml for?

The `<scope>` element can take 6 values: *compile*, *provided*, *runtime*, *test*, *system* and *import*.

This scope is used to limit the transitivity of a dependency, and also to affect the classpath used for various build tasks.

compile

This is the default scope, used if none is specified. Compile dependencies are available in all classpaths of a project. Furthermore, those dependencies are propagated to dependent projects.

provided

This is much like compile, but indicates you expect the JDK or a container to provide the dependency at runtime. For example, when building a web application for the Java Enterprise Edition, you would set the dependency on the Servlet API and related Java EE APIs to scope provided because the web container provides those classes. This scope is only available on the compilation and test classpath, and is not transitive.

runtime

This scope indicates that the dependency is not required for compilation, but is for execution. It is in the runtime and test classpaths, but not the compile classpath.

test

This scope indicates that the dependency is not required for normal use of the application, and is only available for the test compilation and execution phases.

system

This scope is similar to provided except that you have to provide the JAR which contains it explicitly. The artifact is always available and is not looked up in a repository.

import (only available in Maven 2.0.9 or later)

This scope is only used on a dependency of type pom in the section. It indicates that the specified POM should be replaced with the dependencies in that POM's section. Since they are replaced, dependencies with a scope of import do not actually participate in limiting the transitivity of a dependency.