Section 21: Java 9 Module System

15 July 2021 19:23

Section 21: Java 9 Module System

Java 9 introduced a new program component known as a module. You can think of a Java application as collection of **Modules**. (Project Jigsaw)

A **Module** is a named collection of data and code.

Every module comes with a module descriptor file that describes the module and contains metadata about the module.
The module descriptor file is always located directly at the module root folder, and always has the name **module-info.java**.

There are 2 types of modules : **Normal Modules** and **Open Modules**.

A **Normal Module**, without the open modifier, grants access at compile time and run time to types in only those packages which are explicitly exported.

An **Open Module,** with the open modifier, grants access at compile time to types in only those packages which are explicitly exported, but grants access at run time to types in all its packages, as if all packages had been exported.

Project Jigsaw has the following primary goals :-

1) Scalable Platform
2) Security and Maintainability
3) Improved Application Performance
4) Easier Developer Experience

Project Jigsaw defines two types of modules :-
Namely :

1) **Named Modules -** don't officially exist in JDK 9 - We just use the term normal for a named module that is not an automatic module (have module descriptor file i.e. module-info.java)
2) **Unnamed Modules** - does not have a name and it is not declared.

An **Automatic Module** is created after adding a JAR file to the module path (doesn't have module descriptor file i.e. module-info.java)

Basic Modules don't officially exist in JDK 9 - We just use the term basic for a module that is not an open Module.

Aggregator Modules - These exist for convenience. Usually they have no code of their own, they just a module descriptor. They collect and export the contents of other modules, this is the reason why they are named as an aggregator.

A **Module Path** is used by the compiler to find and resolve modules. Every Module from a module path needs to have a module declaration (module-info.java).

A **Class Path** represents a sequence of JAR files.