**Explain Maven Dependencies**

Maven is a popular build automation tool used primarily for Java projects. One of the key features of Maven is its dependency management system, which allows you to easily manage and download the required libraries and dependencies for your project.

Maven dependencies are external libraries or modules that your project relies on to build, run or test successfully. These dependencies can be specified in the project's pom.xml file, which is an XML file that describes the project and its dependencies.

Maven automatically downloads the required dependencies from remote repositories such as Maven Central or your organization's internal repository. It resolves dependencies recursively, meaning it will also download any transitive dependencies required by the direct dependencies.

Maven's dependency management system simplifies the process of managing external libraries and modules, ensuring that your project has all the required dependencies to build, run, and test successfully.

**TestNG Annotation**

TestNG is a popular testing framework for Java applications that provides a variety of annotations to help you write tests and define their behavior. Here are some of the most commonly used TestNG annotations:

**@Test:**

This annotation is used to indicate that a method is a test method. TestNG will automatically execute all methods annotated with @Test.

**@BeforeTest:**

This annotation is used to indicate that a method should be run before any of the test methods in the current test class.

**@AfterTest:**

This annotation is used to indicate that a method should be run after all the test methods in the current test class have been run.

**@BeforeMethod:**

This annotation is used to indicate that a method should be run before each test method in the current test class.

**@AfterMethod:**

This annotation is used to indicate that a method should be run after each test method in the current test class.

**@BeforeClass:**

This annotation is used to indicate that a method should be run before any of the test methods in the current class, but only once.

**@AfterClass:**

This annotation is used to indicate that a method should be run after all the test methods in the current class have been run, but only once.

**@DataProvider:**

This annotation is used to indicate that a method provides data for one or more test methods. The data provider method must return an array of objects or a two-dimensional array of objects.

**@Parameters:**

This annotation is used to inject values into a test method from the TestNG XML configuration file.

**@Listeners:**

This annotation is used to specify one or more listener classes that should be notified of test events, such as when a test succeeds or fails.

These annotations are just a few of the many available in TestNG. By using these annotations, you can create powerful and flexible tests that are easy to write, read, and maintain.