Usage

Best practice is to define the version of the Surefire Plugin that you want to use in either your `pom.xml` or a parent `pom.xml`:

```
<project>
  [...]
  <build>
    <pluginManagement>
      <plugins>
        <plugin>
          <groupId>org.apache.maven.plugins</groupId>
          <artifactId>maven-surefire-plugin</artifactId>
          <version>2.18.1</version>
        </plugin>
      </plugins>
    </pluginManagement>
  </build>
  [...]
</project>
```

The Surefire Plugin can be invoked by calling the `test` phase of the build lifecycle.

```
mvn test
```

Using Different Testing Providers

Tests in your test source directory can be any combination of the following:

- TestNG
- JUnit (3.8 or 4.x)
- POJO

Which providers are available is controlled simply by the inclusion of the appropriate dependencies (i.e., `junit:junit` or `junit:junit-dep` for JUnit and `org.testng:testng` 4.7+ for TestNG). Since this is required to compile the test classes anyway, no additional configuration is required.

Note that any normal Surefire integration works identically no matter which providers are in use - so you can still produce a Cobertura report and a Surefire results report on your project web site for your TestNG tests, for example.

The POJO provider above allows you to write tests that do not depend on either of JUnit and TestNG. It behaves in the same way, running all `test*` methods that are public in the class, but the API dependency is not required. To perform assertions, the JDK 1.4 `assert` keyword can be used. See Using POJO Tests for more information.

All of the providers support the Surefire Plugin parameter configurations. However, there are additional options available if you are running TestNG tests (including if you are using TestNG to execute your JUnit tests, which occurs by default if both are present in Surefire).

See Using TestNG for more information.