

## Exercise 1: Mocking and Stubbing:

### ##ExternalApi.java

```
package org.apache.maven.archetypes.maven_archetype_quickstart;
```

```
public interface ExternalApi {  
    String getData();  
}
```

### ##MyService.java

```
package org.apache.maven.archetypes.maven_archetype_quickstart;
```

```
public class MyService {  
    private ExternalApi api;  
  
    public MyService(ExternalApi api) {  
        this.api = api;  
    }  
  
    public String fetchData() {  
        return api.getData();  
    }  
}
```

### ##MyServiceTest.java

```

package org.apache.maven.archetypes.maven_archetype_quickstart;

import static org.mockito.Mockito.*;

import static org.junit.jupiter.api.Assertions.*;

import org.junit.jupiter.api.Test;

public class MyServiceTest {

    @Test

    public void testExternalApi() {

        // Create mock object

        ExternalApi mockApi = mock(ExternalApi.class);

        // Stub method to return "Mock Data"

        when(mockApi.getData()).thenReturn("Mock Data");

        // Use mock in service

        MyService service = new MyService(mockApi);

        String result = service.fetchData();

        // Verify result

        assertEquals("Mock Data", result);

    }

}

```

###pom.xml

```

<?xml version="1.0" encoding="UTF-8"?>

<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

```

```
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
```

```
<modelVersion>4.0.0</modelVersion>
```

```
<groupId>org.apache.maven.archetypes </groupId>
```

```
<artifactId>maven-archetype-quickstart</artifactId>
```

```
<version>0.0.1-SNAPSHOT</version>
```

```
<name>maven-archetype-quickstart</name>
```

```
<!-- FIXME change it to the project's website -->
```

```
<url>http://www.example.com</url>
```

```
<properties>
```

```
<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
```

```
<maven.compiler.release>17</maven.compiler.release>
```

```
</properties>
```

```
<dependencies>
```

```
<!-- JUnit Jupiter API -->
```

```
<dependency>
```

```
<groupId>org.junit.jupiter</groupId>
```

```
<artifactId>junit-jupiter-api</artifactId>
```

```
<version>5.9.3</version>
```

```
<scope>test</scope>
```

```
</dependency>
```

<!-- JUnit Jupiter Engine -->

<dependency>

<groupId>org.junit.jupiter</groupId>

<artifactId>junit-jupiter-engine</artifactId>

<version>5.9.3</version>

<scope>test</scope>

</dependency>

<!-- Mockito Core -->

<dependency>

<groupId>org.mockito</groupId>

<artifactId>mockito-core</artifactId>

<version>5.6.0</version>

<scope>test</scope>

</dependency>

<!-- JUnit 4 dependency -->

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.13.2</version>

<scope>test</scope>

</dependency>

<dependency>

<groupId>org.slf4j</groupId>

<artifactId>slf4j-api</artifactId>

```

    <version>1.7.36</version>
  </dependency>

  <!-- Use only slf4j-simple for logging -->
  <dependency>
    <groupId>org.slf4j</groupId>
    <artifactId>slf4j-simple</artifactId>
    <version>1.7.36</version>
  </dependency>

</dependencies>

<build>
  <pluginManagement><!-- lock down plugins versions to avoid using Maven defaults (may
be moved to parent pom) -->
    <plugins>
      <!-- clean lifecycle, see https://maven.apache.org/ref/current/maven-
core/lifecycles.html#clean\_Lifecycle -->
      <plugin>
        <artifactId>maven-clean-plugin</artifactId>
        <version>3.4.0</version>
      </plugin>
      <!-- default lifecycle, jar packaging: see https://maven.apache.org/ref/current/maven-
core/default-bindings.html#Plugin\_bindings\_for\_jar\_packaging -->
      <plugin>
        <artifactId>maven-resources-plugin</artifactId>
        <version>3.3.1</version>
      </plugin>
      <plugin>

```

```
<artifactId>maven-compiler-plugin</artifactId>
<version>3.13.0</version>
</plugin>
<plugin>
  <artifactId>maven-surefire-plugin</artifactId>
  <version>3.3.0</version>
</plugin>
<plugin>
  <artifactId>maven-jar-plugin</artifactId>
  <version>3.4.2</version>
</plugin>
<plugin>
  <artifactId>maven-install-plugin</artifactId>
  <version>3.1.2</version>
</plugin>
<plugin>
  <artifactId>maven-deploy-plugin</artifactId>
  <version>3.1.2</version>
</plugin>
<!-- site lifecycle, see https://maven.apache.org/ref/current/maven-core/lifecycles.html#site\_Lifecycle -->
<plugin>
  <artifactId>maven-site-plugin</artifactId>
  <version>3.12.1</version>
</plugin>
<plugin>
  <artifactId>maven-project-info-reports-plugin</artifactId>
  <version>3.6.1</version>
</plugin>
```

```
</plugins>

</pluginManagement>

</build>

</project>
```

## OUTPUT:

```
[INFO] -----
[INFO] T E S T S
[INFO] -----
[INFO] Running org.apache.maven.archetypes.maven_archetype_quickstart.MyServiceTest
WARNING: A Java agent has been loaded dynamically (C:\Users\Dinesh Reddy\.m2\repository\net\bytebuddy\byte-buddy-agent\1.14.8\byte-buddy-agent-1.14.8.jar)
WARNING: If a serviceability tool is in use, please run with -XX:+EnableDynamicAgentLoading to hide this warning
WARNING: If a serviceability tool is not in use, please run with -Djdk.instrument.traceUsage for more information
WARNING: Dynamic loading of agents will be disallowed by default in a future release
OpenJDK 64-Bit Server VM warning: Sharing is only supported for boot loader classes because bootstrap classpath has been appended
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 4.689 s -- in org.apache.maven.archetypes.maven_archetype_quickstart.MyServiceTest
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 15.405 s
[INFO] Finished at: 2025-06-29T12:55:54+05:30
[INFO] -----
```

Activate Windows  
Go to Settings to activate Windows.