EasyMock

Mock Testing

What is Mock Testing?

- Mock testing is a technique used in software development to simulate the behavior of real objects or components.
- It helps isolate units of code for testing by replacing dependencies with mock objects.
- Mock testing enables developers to verify the interactions and behaviors of the tested code without relying on real implementations.

Why Use Mock Objects?

- Allows testing of individual components in isolation.
- Reduces dependencies on external systems or services during testing.
- Enables testing of error conditions and edge cases.
- Improves test reliability and speed.
- Facilitates parallel development and continuous integration.

Introducing EasyMock

- EasyMock is a popular open-source Java library for creating mock objects.
- It provides a simple and intuitive API for defining and manipulating mock objects.
- EasyMock is widely used for unit testing and integration testing in Java applications.
- It integrates seamlessly with popular test frameworks like JUnit and TestNG.

Key Features of EasyMock

- Easy-to-use API for creating mock objects.
- Record and replay approach for defining mock behaviors.
- Support for strict and nice mocks with varying degrees of behavior verification.
- Argument matchers for flexible parameter matching.
- Integration with popular test frameworks.
- Support for partial mocks and mocking of collaborators.

Other Tools

Mockito

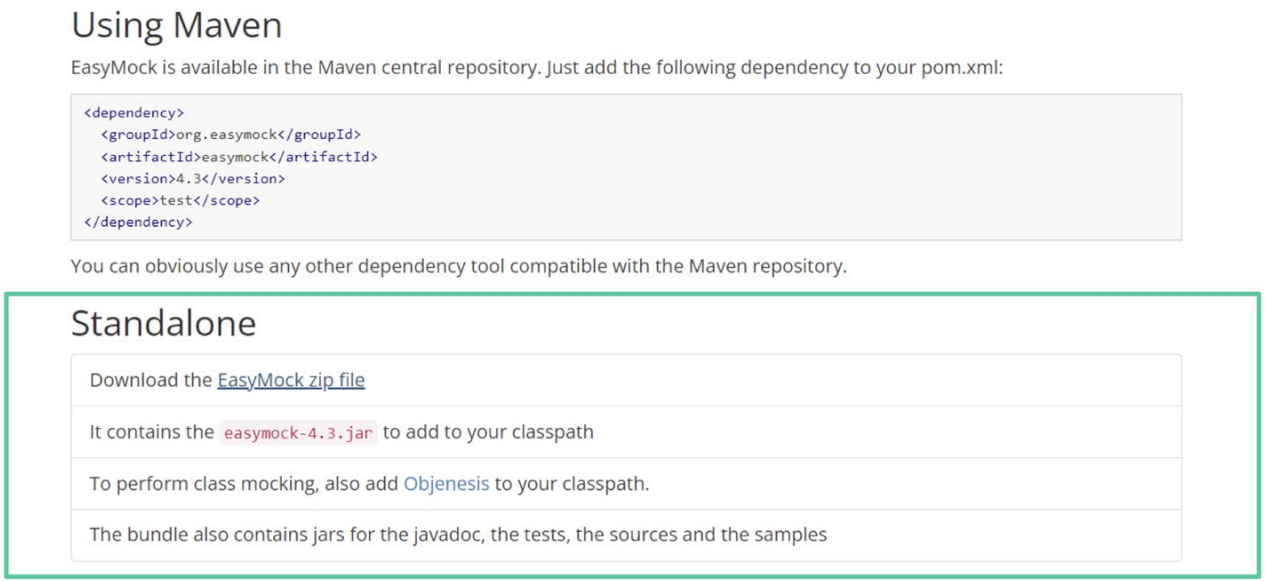
MoQ

JustMock etc.

You can use other tools but explanation will be required on what is the difference between EasyMock and the tool you are planning to use.

Tutorial Link(s)

- https://easymock.org/user-guide.html



Using Maven

EasyMock is available in the Maven central repository. Just add the following dependency to your pom.xml:

```
<dependency>
  <groupId>org.easymock</groupId>
  <artifactId>easymock</artifactId>
  <version>4.3</version>
  <scope>test</scope>
</dependency>
```

You can obviously use any other dependency tool compatible with the Maven repository.

Standalone

Download the EasyMock zip file

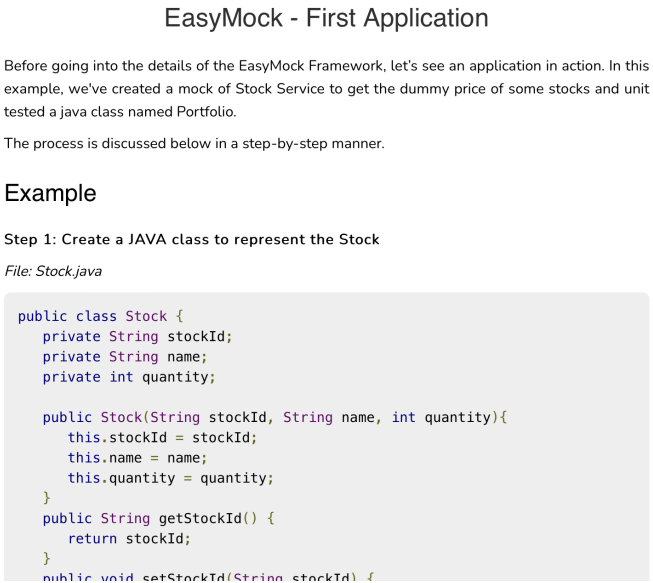
It contains the `easymock-4.3.jar` to add to your classpath

To perform class mocking, also add Objenesis to your classpath.

The bundle also contains jars for the javadoc, the tests, the sources and the samples

Tutorial Link(s)

- https://www.tutorialspoint.com/easymock/easymock_quick_guide.htm



EasyMock - First Application

Before going into the details of the EasyMock Framework, let's see an application in action. In this example, we've created a mock of Stock Service to get the dummy price of some stocks and unit tested a java class named Portfolio.

The process is discussed below in a step-by-step manner.

Example

Step 1: Create a JAVA class to represent the Stock

File: Stock.java

```
public class Stock {
   private String stockId;
   private String name;
   private int quantity;

   public Stock(String stockId, String name, int quantity){
      this.stockId = stockId;
      this.name = name;
      this.quantity = quantity;
   }
   public String getStockId() {
      return stockId;
   }
   public void setStockId(String stockId) {
```