

Answering Calls



The Basics of Answering Calls

The Answer Interface

```
Answer<String> exampleAnswer = invocation -> {  
    Object arg0 = invocation.getArgument(0);  
    // ... Perform operations based on the arguments  
    return "Processed " + arg0;  
};
```

Using Answer in Mockito

```
when(mockedList.get(anyInt())).thenAnswer(invocation -> {  
    Integer index = invocation.getArgument(0);  
    if (index % 2 == 0) {  
        return "Even Index";  
    }  
    return "Odd Index";  
});
```

An Illustrative Example

```
when(mockUserService.getUserName(anyInt())).thenAnswer(invocation -> {  
    Integer userId = invocation.getArgument(0);  
    return "User" + userId;  
});
```

Dynamic Behavior with Answer

Conditional Logic

```
when(mockedObject.calculateRiskFactor(any(InsuranceQuote.class))).thenAnswer(invocation -> {  
    InsuranceQuote quote = invocation.getArgument(0);  
    if (quote.getAmount() > 1000000) {  
        throw new HighRiskException("Quote too high!");  
    }  
    return new RiskAssessment(LOW_RISK);  
});
```

Responding to Multiple Calls

```
when(mockedObject.isServiceAvailable()).thenAnswer(new Answer<Boolean>() {  
    private int count = 0;  
  
    @Override  
    public Boolean answer(InvocationOnMock invocation) throws Throwable {  
        return count++ < 3;  
    }  
});
```


Emulating Complex Logic

```
when(mockedRepository.findItemsByType(anyString()))
    .thenAnswer(invocation -> {
        String itemType = invocation.getArgument(0);
        if ("electronics".equals(itemType)) {
            return Stream.of("TV", "Radio");
        } else if ("books".equals(itemType)) {
            return Stream.of("Novel", "Science Fiction");
        } else {
            return Stream.empty();
        }
    })
);
```

Advanced Answer Techniques

Using AdditionalAnswers

//AdditionalAnswer approach

```
Mockito.when(mockObject.calculateStringLength(Mockito.anyString()))  
    .then(AdditionalAnswers.answer(i -> i.toString().length()));
```

// Traditional Answer approach

```
Mockito.when(mockObject.calculateStringLength(Mockito.anyString()))  
    .thenAnswer(invocation -> {  
        String arg = invocation.getArgument(0);  
        return arg.length();  
    });
```

Testing Hypotheses with Answers

```
Mockito.when(mockObject.calculate(Mockito.anyInt(),
Mockito.anyInt()))
    .thenAnswer(invocation -> {
        int a = invocation.getArgument(0);
        int b = invocation.getArgument(1);
        if (a == 0) {
            return b; // Simulating a specific case
        } else {
            return a + b; // Default behavior
        }
    });
```

Performance Investigation with Answers

```
Mockito.when(mockObject.timeConsumingOperation(Mockito.any()))
    .thenAnswer(invocation -> {
        long startTime = System.currentTimeMillis();
        Object result = performOperation(invocation.getArgument(0));
        long endTime = System.currentTimeMillis();
        System.out.println("Operation time: " + (endTime - startTime)
+ " ms");
        return result;
    });
```