

# JUnit Mockito Logging assignment

## Maithreyan K

Use Spring MVC or any Spring Boot application, add more validations to it.

1. Write JUnit test cases for each and every use case with 100% coverage
2. Coverage screenshot has to be attached along with assignment
3. Each test case should contain at least one assert statement
4. No real time calls should happen, all calls should be completely mocked
5. Add appropriate logs using Logger wherever applicable, possibly use all levels of logging

### 1)100% Coverage

100% classes, 100% lines covered in 'all classes in scope'			
Element	Class, %	Method, %	Line, %
com.example.demo.controller	100% (1/1)	100% (4/4)	100% (15/15)

### 2)Coverage Screenshot

The screenshot shows the IntelliJ IDEA interface with the 'Test Results' window open. The window displays the following test results:

Test Name	Status	Duration
OrderControllerTest	Passed	542 ms
createOrderTest()	Passed	530 ms
deleteItemTest()	Passed	6 ms
getItemListTest()	Passed	3 ms
addItemTest()	Passed	3 ms

A tooltip at the bottom left of the window indicates 'Tests passed: 4'. The bottom status bar shows various tool icons including Cover, TODO, Problems, Terminal, CheckStyle, SonarLint, and Build.

### 3) Assertions Used

```
@Test
void addItemTest(){
    Mockito.when(orderService.addItemIntoOrder(Mockito.any(Item.class), Mockito.anyInt())).thenReturn("Item Added Successfully");
    Item item = new Item( itemId: 1, itemName: "Test item 1", description: "description", quantity: 5);
    ResponseEntity<String> successResponse = orderController.addItem(item, orderId: 1);
    Assertions.assertEquals( expected: "Item Added Successfully", successResponse.getBody());

    Mockito.when(orderService.addItemIntoOrder(Mockito.any(Item.class), Mockito.anyInt())).thenReturn("Order Not Found !!");
    ResponseEntity<String> orderNotFoundResponse = orderController.addItem(item, orderId: 1);
    Assertions.assertEquals( expected: "Order Not Found !!", orderNotFoundResponse.getBody());
}
```

### 4)Calls Mocked

```
@ExtendWith(MockitoExtension.class)
class OrderControllerTest {

    @InjectMocks
    OrderController orderController;

    @Mock
    OrderService orderService;
}
```

### 5)Logger used

```
private static final Logger logger = LoggerFactory.getLogger(OrderController.class);
@PostMapping("/create")
public ResponseEntity<String> createOrder(@RequestBody Order order) {
    String response = orderService.createOrder(order);
    if(response.contains("Order Successfully Created!")) {
        logger.info("Order created", order);
        return new ResponseEntity<>(response, HttpStatus.CREATED);
    }
    logger.error("Not Created");
}
```