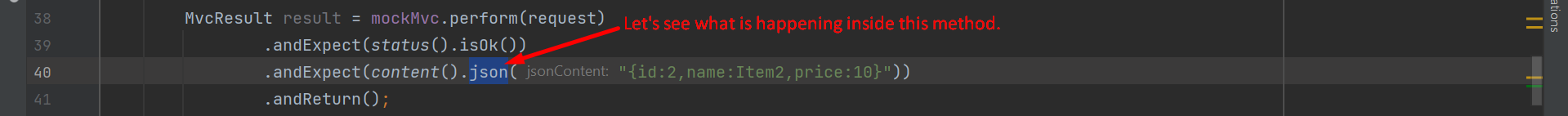
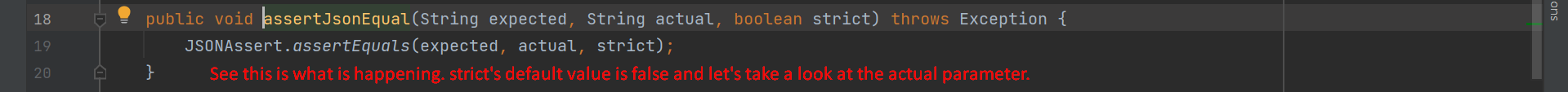
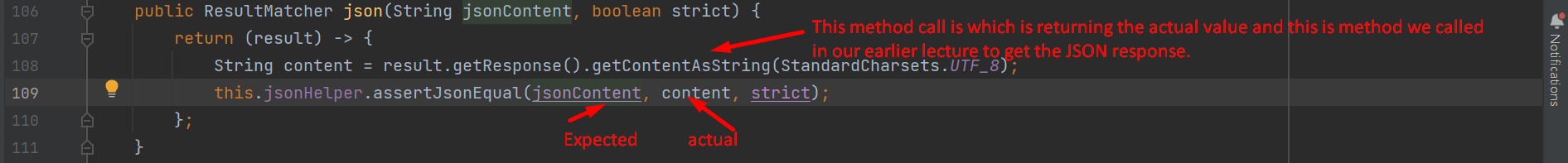
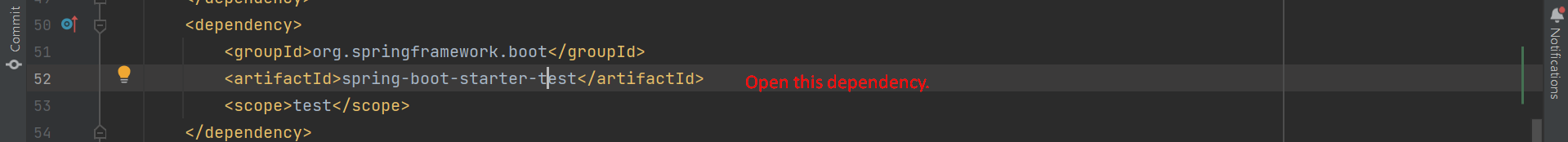
1. **Agenda**:
   1. More detail about json()
2. Text

   Description automatically generated
3. Let’s go to the method definition.
   1. 
   2. 
   3. 
   4. A screenshot of a computer

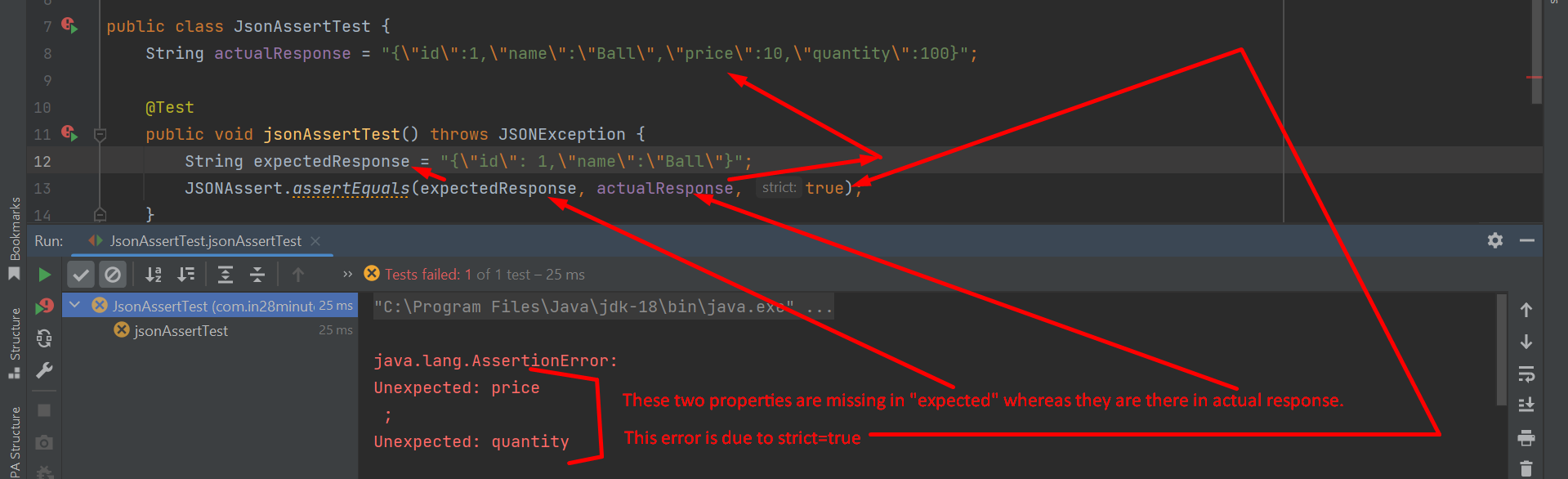
      Description automatically generated with medium confidence
   5. 
   6. A picture containing text

      Description automatically generated

This **jsonAssert** is very powerful framework to write unit testing for JSON content very easily.

1. Let’s play with **JSONassert class**.
   1. **Case 01**: When actual and expected JSON are same 🡺 Test case will pass.Text

      Description automatically generated with medium confidence
   2. **Case 02**: when there is space in expected JSON:  
      Graphical user interface

      Description automatically generated
   3. **Case 03**: When any property is missing from expected value but is there in actual response along with strict=true 🡺 The test case will fail saying “Unexpected <property-name>”  
        
        
      
   4. **Case 04**: When values mismatch 🡺 Test will fail.  
      Chart

      Description automatically generated with medium confidence
   5. **Case 05**: Without Escaped Characters 🡺 Test will pass.   
      Graphical user interface

      Description automatically generated with medium confidenceThe test case will pass no matter strict is false or true.  
      But if you have space in your values, then those values must be inside double quotation marks.  
      Website

      Description automatically generated with low confidence
   6. **case 06: Order of values doesn’t matter in JSON irrespective of the value of strict**String expectedResponse = "{id: 1,price:10, **name:Ball**, quantity:100}";  
      String expectedResponse = "{id: 1,price:10, quantity:100, **name:Ball** }";