danglotb / icst-2019 **Ÿ** Fork Watch \* Star 0

Insights

**Dismiss** 

Copy path

Find file

## Join GitHub today

Projects 0

icst-2019 / september-2018 / result / gson / commit\_8\_44cad04 / input\_amplification / com / google / gson / functional /

and review code, manage projects, and build software together. Sign up

GitHub is home to over 28 million developers working together to host

! Issues 0

Pull requests 0

<> Code

Branch: master ▼

```
ArtificialAmplPrimitiveTest.java
 danglotb Create ArtificialAmplPrimitiveTest.java
                                                                                                                        d1cff39 17 seconds ago
1 contributor
 126 lines (116 sloc)
                        6.76 KB
                                                                                                        Raw
                                                                                                              Blame
                                                                                                                       History
         package com.google.gson.functional;
    1
    2
    3
    4
         import com.google.gson.Gson;
    5
         import com.google.gson.JsonSyntaxException;
    6
         import com.google.gson.internal.LazilyParsedNumber;
    7
         import junit.framework.TestCase;
    8
         import org.junit.Assert;
   9
         import org.junit.Before;
   10
         import org.junit.Test;
   11
   12
   13
         public class AmplPrimitiveTest {
   14
             private Gson gson;
   15
             @Before
   17
             public void setUp() throws Exception {
                 gson = new Gson();
   19
             }
             @Test(timeout = 10000)
             public void testNumberDeserialization_literalMutationString4_failAssert3_literalMutationString204_failAssert37() throws Except
                 try {
                     try {
                         String json = "1";
                         Number expected = new Integer(json);
   27
                         Number actual = this.gson.fromJson(json, Number.class);
                         expected.intValue();
                         actual.intValue();
                         json = String.valueOf(Long.MAX_VALUE);
                         expected = new Long(json);
                         actual = this.gson.fromJson(json, Number.class);
                         expected.longValue();
                         actual.longValue();
                         json = "^";
                         actual = this.gson.fromJson(json, Number.class);
   37
                         actual.longValue();
                         org.junit.Assert.fail("testNumberDeserialization_literalMutationString4 should have thrown NullPointerException");
                     } catch (NullPointerException expected) {
                     org.junit.Assert.fail("testNumberDeserialization_literalMutationString4_failAssert3_literalMutationString204 should ha
   41
   42
                 } catch (JsonSyntaxException expected_1) {
                     Assert.assertEquals("Expecting number, got: STRING", expected_1.getMessage());
                 }
             }
   47
             @Test(timeout = 10000)
             public void testNumberDeserialization_add17_literalMutationString391_failAssert23() throws Exception {
   49
                     String json = "1";
   51
                     Number expected = new Integer(json);
                     Number actual = this.gson.fromJson(json, Number.class);
   52
                     int o_testNumberDeserialization_add17__6 = expected.intValue();
   53
                     int o_testNumberDeserialization_add17__7 = actual.intValue();
   55
                     json = String.valueOf(Long.MAX_VALUE);
                     expected = new Long(json);
   57
                     actual = this.gson.fromJson(json, Number.class);
                     long o_testNumberDeserialization_add17__14 = expected.longValue();
                     long o_testNumberDeserialization_add17__15 = actual.longValue();
                     json = "h1,";
```

61 Number o\_testNumberDeserialization\_add17\_\_17 = this.gson.fromJson(json, Number.class); actual = this.gson.fromJson(json, Number.class); 62 long o\_testNumberDeserialization\_add17\_\_20 = actual.longValue(); 63 64 65 } catch (JsonSyntaxException expected) { Assert.assertEquals("Expecting number, got: STRING", expected.getMessage()); 66 67 } 69 @Test(timeout = 10000) 71 public void testNumberDeserialization\_literalMutationString8\_failAssert7\_add383() throws Exception { try { String json = "1"; Assert.assertEquals("1", json); 74 Number expected = new Integer(json); Number actual = this.gson.fromJson(json, Number.class); int o\_testNumberDeserialization\_literalMutationString8\_failAssert7\_add383\_\_8 = expected.intValue(); Assert.assertEquals(1, ((int) (o\_testNumberDeserialization\_literalMutationString8\_failAssert7\_add383\_\_8))); int o\_testNumberDeserialization\_literalMutationString8\_failAssert7\_add383\_\_9 = actual.intValue();  $Assert.assert \texttt{Equals}(\texttt{1}, ((\texttt{int}) (o\_testNumberDeserialization\_literalMutationString8\_failAssert7\_add383\_9))); \\$ 81 json = String.valueOf(Long.MAX\_VALUE); Assert.assertEquals("9223372036854775807", json); expected = new Long(json); 83 actual = this.gson.fromJson(json, Number.class); long o\_testNumberDeserialization\_literalMutationString8\_failAssert7\_add383\_\_16 = expected.longValue(); 85 87 long o\_testNumberDeserialization\_literalMutationString8\_failAssert7\_add383\_\_17 = actual.longValue(); 88 long o\_testNumberDeserialization\_literalMutationString8\_failAssert7\_add383\_\_18 = actual.longValue(); json = "dhs"; Assert.assertEquals("dhs", json); actual = this.gson.fromJson(json, Number.class); actual.longValue(); org.junit.Assert.fail("testNumberDeserialization\_literalMutationString8 should have thrown JsonSyntaxException"); } catch (JsonSyntaxException expected) { } } @Test(timeout = 10000) public void testNumberDeserialization\_literalMutationString7\_failAssert6\_add181\_add1867() throws Exception { try { String json = "1"; Assert.assertEquals("1", json); Number expected = new Integer(json); Number actual = this.gson.fromJson(json, Number.class); int o\_testNumberDeserialization\_literalMutationString7\_failAssert6\_add181\_\_8 = expected.intValue(); int o\_testNumberDeserialization\_literalMutationString7\_failAssert6\_add181\_\_9 = actual.intValue(); 110 json = String.valueOf(Long.MAX\_VALUE); 111 Assert.assertEquals("9223372036854775807", json); 112 expected = new Long(json); 113 actual = this.gson.fromJson(json, Number.class); 114 long o\_testNumberDeserialization\_literalMutationString7\_failAssert6\_add181\_\_16 = expected.longValue(); 115 long o\_testNumberDeserialization\_literalMutationString7\_failAssert6\_add181\_\_17 = actual.longValue(); 116 long o\_testNumberDeserialization\_literalMutationString7\_failAssert6\_add181\_\_18 = actual.longValue(); 117 json = "k.0";118