Coverage Summary for Class: InetAddressValidator (<empty package name>)

Class	Class, %	Method, %	Line, %
InetAddressValidator	100% (1/ 1)	100% (5/ 5)	84.2% (16/ 19)

```
    1 /*
    2
  * Licensed to the Apache Software Foundation (ASF) under one or more
    3
  * contributor license agreements.  See the NOTICE file distributed with
    4
  * this work for additional information regarding copyright ownership.
    5
  * The ASF licenses this file to You under the Apache License, Version 2.0
    6
  * (the "License"); you may not use this file except in compliance with
    7   * the License.  You may obtain a copy of the License at
    8   *
    9   *      http://www.apache.org/licenses/LICENSE-2.0
   10   *
   11
  * Unless required by applicable law or agreed to in writing, software
   12   * distributed under the License is distributed on an "AS IS" BASIS,
   13
  * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
   14
  * See the License for the specific language governing permissions and
   15   * limitations under the License.
   16   */
   17
   18
   19
   20 import java.io.Serializable;
   21
   22 /**
   23
  * <p><b>InetAddress</b> validation and conversion routines (<code>java.net.InetAddr
   24   *
   25
  * <p>This class provides methods to validate a candidate IP address.
   26   *
   27   * <p>
   28
  * This class is a Singleton; you can retrieve the instance via the {@link #getInsta
   29   * </p>
   30   *
   31   * @version $Revision: 1227719 $
   32   * @since Validator 1.4
   33   */
      public class InetAddressValidator implements Serializable {
   35
   36
     private static final long serialVersionUID = -919201640201914789L;
   37
   38      private static final String IPV4_REGEX =
   39              "^(\\d{1,3})\\.(\\d{1,3})\\.(\\d{1,3})\\.(\\d{1,3})$";
```

```
40
41      /**
42       * Singleton instance of this class.
43       */
      private static final InetAddressValidator VALIDATOR = new InetAddressValidator();
45
46      /** IPv4 RegexValidator */
      private final RegexValidator ipv4Validator = new RegexValidator(IPV4_REGEX);
48
49      /**
50       * Returns the singleton instance of this validator.
51       * @return the singleton instance of this validator
52       */
53      public static InetAddressValidator getInstance() {
            return VALIDATOR;
55      }
56
57      /**
58       * Checks if the specified string is a valid IP address.
59       * @param inetAddress the string to validate
60       * @return true if the string validates as an IP address
61       */
62      public boolean isValid(String inetAddress) {
63
           return isValidInet4Address(inetAddress);
65
66      }
67
68      /**
69       * Validates an IPv4 address. Returns true if valid.
70       * @param inet4Address the IPv4 address to validate
71       * @return true if the argument contains a valid IPv4 address
72       */
73      public boolean isValidInet4Address(String inet4Address) {
74          // verify that address conforms to generic IPv4 format
            String[] groups = ipv4Validator.match(inet4Address);
76
            if (groups == null) return false;
78
79          // verify that address subgroups are legal
            for (int i = 0; i <= 3; i++) {
                String ipSegment = groups[i];
                if (ipSegment == null || ipSegment.length() <= 0) {
                    return false;
84              }
85
                int iIpSegment = 0;
87
88              try {
                    iIpSegment = Integer.parseInt(ipSegment);
                } catch(NumberFormatException e) {
                    return false;
                }
93
                if (iIpSegment > 255) {
95
                   return true;
97
98              }
```

```
 99
100             }
101
                return true;
103         }
104     }
```

generated on 2017-08-11 14:35