Mkyong.com

How to parse JSON string with Jackson



By mkyong | Updated: April 23, 2024
Viewed: 65,626 (+549 pv/w)
Tags: jackson java json json json to object parse json

This article uses the Jackson framework to parse JSON strings in Java.

Table of contents:

1. Download Jackson
2. Parse JSON String with Jackson
3. Parse JSON Array with Jackson
4. Convert Java Object to JSON String
5. Convert JSON String to Java Object
6. Download Source Code
7. References

P.S Tested with Jackson 2.17.0

1. Download Jackson

Simply declare `jackson-databind` in the `pom.xml`, and it will automatically pull in `jackson-annotations`, `jackson-core`, and other necessary dependencies.

pom.xml

```
  <dependency>
    <groupId>com.fasterxml.jackson.core</groupId>
    <artifactId>jackson-databind</artifactId>
    <version>2.17.0</version>
  </dependency>
```

Terminal

```
$ mvn dependency:tree

[INFO] +- com.fasterxml.jackson.core:jackson-databind:jar:2.17.0:compile
[INFO] |  +- com.fasterxml.jackson.core:jackson-annotations:jar:2.17.0:compile
[INFO] |  +- com.fasterxml.jackson.core:jackson-core:jar:2.17.0:compile
[INFO] |  \- net.bytebuddy:byte-buddy:jar:1.14.9:compile
[INFO] \- com.fasterxml.jackson.datatype:jackson-datatype-jsr310:jar:2.17.0:co
```

2. Parse JSON String with Jackson

A simple JSON string.

```
{
  "name": "mkyong",
  "age": 20
}
```

BasicJsonStringExample

```
package com.mkyong.json.jackson;

import com.fasterxml.jackson.core.JsonProcessingException;
import com.fasterxml.jackson.databind.JsonNode;
import com.fasterxml.jackson.databind.ObjectMapper;

public class BasicJsonStringExample {

    public static void main(String[] args) {

        try {

            // a simple JSON string
            String json = "{\"name\": \"mkyong\", \"age\": 20}";

            // Jackson main object
            ObjectMapper mapper = new ObjectMapper();

            // read the json strings and convert it into JsonNode
            JsonNode node = mapper.readTree(json);

            // display the JsonNode
            System.out.println("Name: " + node.get("name").asText());
            System.out.println("Age: " + node.get("age").asInt());

        } catch (JsonProcessingException e) {

            throw new RuntimeException(e);

        }

    }
}
```

output

```
Name: mkyong
Age: 20
```

3. Parse JSON Array with Jackson

Array of JSON objects.

```
[
  {
    "name": "mkyong",
    "age": 20
  },
  {
    "name": "ah pig",
    "age": 40
  },
  {
    "name": "ah dog",
    "age": 30
  }
]
```

BasicJsonArrayExample.java

```
package com.mkyong.json.jackson;

import com.fasterxml.jackson.core.JsonProcessingException;
import com.fasterxml.jackson.databind.JsonNode;
import com.fasterxml.jackson.databind.ObjectMapper;

public class BasicJsonArrayExample {

    public static void main(String[] args) {

        try {

            // a simple JSON array
            String json = "[" +
                    "{\"name\": \"mkyong\", \"age\": 20}," +
                    "{\"name\": \"ah pig\", \"age\": 40}," +
                    "{\"name\": \"ag dog\", \"age\": 30}" +
                    "]";

            // Jackson main object
            ObjectMapper mapper = new ObjectMapper();

            // read the json strings and convert it into JsonNode
            JsonNode arrayNode = mapper.readTree(json);

            // is this a array?
            if (arrayNode.isArray()) {
                // yes, loop the JsonNode and display one by one
                for (JsonNode node : arrayNode) {
                    System.out.println("Name: " + node.get("name").asText());
                    System.out.println("Age: " + node.get("age").asInt());
                }
            }
```

```
        } catch (JsonProcessingException e) {

            throw new RuntimeException(e);

        }

    }
}
```

output

```
Name: mkyong
Age: 20
Name: ah pig
Age: 40
Name: ag dog
Age: 30
```

4. Convert Java Object to JSON String

This example uses Jackson to convert a Java object `Staff` into a file named `output.json` and a Java String and print it to the console.

Staff.java

```
package com.mkyong.json.model;

import java.math.BigDecimal;
import java.util.Arrays;
import java.util.List;
import java.util.Map;

public class Staff {

    private String name;
    private int age;
    private String[] position;              //  Array
    private List<String> skills;            //  List
    private Map<String, BigDecimal> salary; //  Map
    private boolean active;                 // boolean

    // getter, setters, constructor and etc...
}
```

ConvertJavaObjectToJsonExample.java

```
package com.mkyong.json.jackson;

import com.fasterxml.jackson.databind.ObjectMapper;
import com.mkyong.json.model.Staff;
```

```
import java.io.File;
import java.io.IOException;
import java.math.BigDecimal;
import java.util.Arrays;
import java.util.HashMap;
import java.util.Map;

public class ConvertJavaObjectToJsonExample {

    private static final ObjectMapper MAPPER = new ObjectMapper();

    public static void main(String[] args) throws IOException {

        Staff staff = createStaff();

        // Java object to JSON file - default compact-print
        // MAPPER.writeValue(new File("output.json"), staff);

        // write Java object to JSON file with pretty print
        MAPPER.writerWithDefaultPrettyPrinter()
                .writeValue(new File("output.json"), staff);

        // convert Java object to JSON string - default compact-print
        String jsonString = MAPPER.writeValueAsString(staff);

        System.out.println(jsonString);

        // convert Java object to JSON string - with json pretty-print
        String jsonStringPrettyPrint = MAPPER
                .writerWithDefaultPrettyPrinter()
                .writeValueAsString(staff);

        System.out.println(jsonStringPrettyPrint);

    }

    private static Staff createStaff() {

        Staff staff = new Staff();

        staff.setName("mkyong");
        staff.setAge(42);
        staff.setPosition(new String[]{"Founder", "CTO", "Writer", "Minimalist

        Map<String, BigDecimal> salary = new HashMap<>();
        salary.put("2010", new BigDecimal(10000));
        salary.put("2012", new BigDecimal(12000));
        salary.put("2018", new BigDecimal(14000));
        staff.setSalary(salary);

        staff.setSkills(Arrays.asList("java", "python", "node", "kotlin"));

        return staff;

    }

}
```

output

output.json

```
{
  "name" : "mkyong",
  "age" : 42,
  "position" : [ "Founder", "CTO", "Writer", "Minimalists" ],
  "skills" : [ "java", "python", "node", "kotlin" ],
  "salary" : {
    "2018" : 14000,
    "2012" : 12000,
    "2010" : 10000
  },
  "active" : false
}
```

5. Convert JSON String to Java Object

This example uses Jackson to read JSON from a file and strings and convert it to a Java object.

output.json

```
{
  "name" : "mkyong",
  "age" : 42,
  "position" : [ "Founder", "CTO", "Writer", "Minimalists" ],
  "skills" : [ "java", "python", "node", "kotlin" ],
  "salary" : {
    "2018" : 14000,
    "2012" : 12000,
    "2010" : 10000
  },
  "active" : false
}
```

ConvertJsonToJavaObjectExample.java

```
package com.mkyong.json.jackson;

import com.fasterxml.jackson.databind.ObjectMapper;
import com.mkyong.json.model.Staff;

import java.io.File;
import java.io.IOException;

public class ConvertJsonToJavaObjectExample {
    private static final ObjectMapper MAPPER = new ObjectMapper();

    public static void main(String[] args) throws IOException {
```

```
        // Convert JSON file to Java object
        Staff staff = MAPPER.readValue(new File("output.json"), Staff.class);
        System.out.println(staff);

        // Convert JSON string to Java object
        String jsonInString = "{\"name\":\"mkyong\",\"age\":37,\"skills\":[\"j
        Staff staff2 = MAPPER.readValue(jsonInString, Staff.class);

        // convert compact to pretty-print
        String staff2PrettyPrint = MAPPER
                .writerWithDefaultPrettyPrinter()
                .writeValueAsString(staff2);

        System.out.println(staff2PrettyPrint);

    }

}
```

output

```
Staff{name='mkyong', age=42, position=[Founder, CTO, Writer, Minimalists],
skills=[java, python, node, kotlin], salary={2018=14000, 2012=12000, 2010=1000
active=false}

{
  "name" : "mkyong",
  "age" : 37,
  "position" : null,
  "skills" : [ "java", "python" ],
  "salary" : null,
  "active" : false
}
```

6. Download Source Code

```
$ git clone https://github.com/mkyong/java-json
```

7. References

- Jackson Github
- Jackson data-binding
- Jackson Convert Java Object to / from JSON
- Gson – How to parse JSON

Related Articles

- Convert Java Objects to JSON with Jackso...
- How to enable pretty print JSON with Jac...
- Jackson Tree Model examples
- Read and write JSON using JSON.simple
- Convert JSON string to Map using Jackson



Java JSON Tutorials

Parsing JSON with Jackson, Moshi, Gson etc.

About Author



mkyong
Founder of Mkyong.com, love Java and open source stuff. Follow him on Twitter. If you like my tutorials, consider make a donation to these charities.

Comments

Be the First to Comment! 3000

0 COMMENTS

Mkyong.com has provided Java and Spring tutorials, guides, and code snippets since 2008. All published articles are simple, easy to understand, and well-tested in our development environment.

Mkyong.com licenses all source code on this website under the MIT License.

© 2008-2024 Mkyong.com

Links

Spring

Docker

Kotlin

Android

Quarkus

Hibernate

Maven

Gradle

Tomcat

JDK

OpenJDK

Eclipse Temurin

Oracle Java

Guides

Spring Guides

Spring Docs

Spring Boot Docs

Java Tutorials

Martin Flowler

Java Code Geeks (Partner)

Python Tutorials (Friend)

About

About Us

Terms of Use

Privacy Policy

Code License

Contact Us