

HOW TO USE FASTJSON



INDEX

1. INTRODUCTION
2. HOW TO CONVERT JAVA OBJECTS TO JSON STRING
3. HOW TO CONVERT JSON STRINGS TO JAVA OBJECT
4. HOW TO CREATE JSON OBJECTS

1. INTRODUCTION

Fastjson is the fastest java library currently. This ambitious project offers serialization, deserialization and json objects creation. In this manual the basic methods will be explained. Remember to configure your pom.xml first:

```
<dependency>
  <groupId>com.alibaba</groupId>
  <artifactId>fastjson</artifactId>
  <version>1.2.76</version>
</dependency>
```

2. HOW TO CONVERT JAVA OBJECTS TO JSON STRING

To convert a Java object to a Json String it is necessary to define the following Person Java Bean:

```
public class Person {

    @JSONField(name = "AGE")
    private int age;

    @JSONField(name = "FULL NAME")
    private String fullName;

    @JSONField(name = "DATE OF BIRTH")
    private Date dateOfBirth;

    public Person(int age, String fullName, Date dateOfBirth) {
        super();
        this.age = age;
        this.fullName= fullName;
        this.dateOfBirth = dateOfBirth;
    }

    // standard getters & setters
}
```

Then, we use the function *JSON.toJSONString()* to convert Java object to a JSON String:

```
private List<Person> listOfPersons = new ArrayList<Person>();

@Before
public void setUp() {
    listOfPersons.add(new Person(15, "John Doe", new Date()));
    listOfPersons.add(new Person(20, "Janette Doe", new Date()));
}
```

```
@Test
public void whenJavaList_thanConvertToJsonCorrect() {
    String jsonOutput= JSON.toJSONString(listOfPersons);
}
```

The result:

```
[
  {
    "AGE":15,
    "DATE OF BIRTH":1468962431394,
    "FULL NAME":"John Doe"
  },
  {
    "AGE":20,
    "DATE OF BIRTH":1468962431394,
    "FULL NAME":"Janette Doe"
  }
]
```

3. HOW TO CONVERT JSON STRINGS TO JAVA OBJECT

Starting from the previous point, we can convert this java object to a json string using the following function: `JSON.parseObject()`

```
Person person = new Person(20, "John", "Doe", new Date());
String jsonObject = JSON.toJSONString(person);
Person newPerson = JSON.parseObject(jsonObject,
Person.class);
```

The result:

```
Person [age=20, fullName=John Doe, dateOfBirth=Wed Jul 20
08:51:12 WEST 2016]
```

4. HOW TO CREATE JSON OBJECTS

It's only a matter of combining `JSONObject` and `JSONArray` objects:

```
@Test
public void whenGenerateJson_thanGenerationCorrect() throws
ParseException {
    JSONArray jsonArray = new JSONArray();
    for (int i = 0; i < 2; i++) {
        JSONObject jsonObject = new JSONObject();
        jsonObject.put("AGE", 10);
        jsonObject.put("FULL NAME", "Doe " + i);
        jsonObject.put("DATE OF BIRTH", "2016/12/12 12:12:12");
    }
}
```

```
        jsonArray.add(jsonObject);
    }
    String jsonOutput = jsonArray.toJSONString();
}
```

And here is the result:

```
[
  {
    "AGE": "10",
    "DATE OF BIRTH": "2016/12/12 12:12:12",
    "FULL NAME": "Doe 0"
  },
  {
    "AGE": "10",
    "DATE OF BIRTH": "2016/12/12 12:12:12",
    "FULL NAME": "Doe 1"
  }
]
```