

N 对象和数组

针对创建的 JSON 对象，我们可以使用类似 LinkedHashMap 来创建对象，对于数组，我们就可以用我们常用的 ArrayList。

因为 JSON 对象使用的是 LinkedHashMap，所以我们可以利用 Map 的特性，使用 `LinkedHashMap.put()` 方法来 Push 数据到 Map 中。

```
...
@Test
public void createJsonStringTest() throws IOException {
    System.out.println(JSON.std.with(JSON.Feature.PRETTY_PRINT_OUTPUT)
        .asString(new LinkedHashMap<String, Object>() {{
            put("name", "John Doe");
            put("age", 30);
        }}));
}
```

...

Java

Jackson-jr Composer

Jackson-jr 同时还提供了 Composer 的这个创建方法，这个创建方法更加直接。

你可以按照你的想法直接进行定义和输出，在这之前，我们通常需要定义一个对象，然后对对象进行完成数据后再输出，对于一些简单的 JSON 格式。我们可以不用定义对象了，而是直接使用。

```
...
public static String jsonComposer() throws IOException {
    return JSON.std.with(JSON.Feature.PRETTY_PRINT_OUTPUT)
        .composeString()
```

```

        .startObject()
        .startArrayField("objectArray")
        .startObject()
        .put("name", "name1")
        .put("age", 11)
        .end()
        .startObject()
        .put("name", "name2")
        .put("age", 12)
        .end()
        .end()
        .startArrayField("array")
        .add(1)
        .add(2)
        .add(3)
        .end()
        .startObjectField("object")
        .put("name", "name3")
        .put("age", 13)
        .end()
        .put("last", true)
        .end()
        .finish();
    }
}
...

```

Java

```

        return new CustomDateDeserialzer();
    }
    return null;
}
}
...

```

Java

上面和下面的这些代码显示了如何我们如何使用自定义的序列化和反序列化方法。

```

...
Person person = new Person("John Doe", 30, LocalDate.now());

```

```
JSON jsonMapper = JSON.builder().register(new JacksonJrExtension() {
    @Override
    protected void register (ExtensionContext extensionContext) {
        extensionContext.insertProvider(new MyHandlerProvider());
    }
}).build().with(JSON.Feature.PRETTY_PRINT_OUTPUT);

String json = jsonMapper.asString(person);
Person deserializedPerson = jsonMapper.beanFrom(Person.class, json);
...
```

Java

Jackson-jr 对比 Jackson

关于Jackson-jr 对比 Jackson 的内容，有人在做了一张下面的图。

简单点来说就 Jackson-jr 是Jackson 的轻量级应用，因为我们在很多时候都用不到 Jackson 的很多复杂功能。

对很多应用来说，我们可能只需要使用简单的 JSON 读写即可。

如我们用不到什么复杂的功能，并且使用了 Jackson-jr 能够满足你的项目使用的话，就直接使用 Jackson-jr 即可。

如发现 Jackson-jr 没有办法满足你的所有需求的时候，可以再切换到传统的 Jackson 包。

结论

Jackson-jr 提供了相对 Jackson 来说更加轻量的 JSON 数据处理模块。

对于很多只需要简单读写功能的 JSON 应用来说，就可以直接进行使用，对于服务器的应用，可能这个功能的削减对你的影响不大。

对大部分程序员来说，只需要了解 Jackson-jr 提供了一个轻量的兼容 JSON 大部分功能的解决方案即可。

[www.isharkfly.com/t/java-jack...](<http://cxyroad.com/>
”<https://www.isharkfly.com/t/java-jackson-jr/15710>”)

原文链接: <https://juejin.cn/post/7364319614601396243>