



Gson

Enjoy JSON (De-)Serialization
in Java

Norman Peitek

Gson: Enjoy JSON (De-)Serialization in Java

Norman Peitek

This book is for sale at <http://leanpub.com/gson-enjoy-json-deserialization-in-java>

This version was published on 2019-02-17



This is a [Leanpub](#) book. Leanpub empowers authors and publishers with the Lean Publishing process. [Lean Publishing](#) is the act of publishing an in-progress ebook using lightweight tools and many iterations to get reader feedback, pivot until you have the right book and build traction once you do.

© 2016 - 2019 Future Studio

Tweet This Book!

Please help Norman Peitek by spreading the word about this book on [Twitter](#)!

The suggested hashtag for this book is [#gsonbook](#).

Find out what other people are saying about the book by clicking on this link to search for this hashtag on Twitter:

[#gsonbook](#)

Also By **Norman Peitek**

[Picasso: Easy Image Loading on Android](#)

[Retrofit: Love Working with APIs on Android](#)

[Glide: Customizable Image Loading on Android](#)

[Gson Workbook](#)

Contents

About the Book	i
What Topics Are Covered in This Book?	i
Who Is This Book For?	i
Code Examples	ii
Workbook	ii
Chapter 1 — Introduction to Java-JSON Mapping	1
Adding Gson as Dependency	1
Basics of Java-JSON Serialization	3
Basics of Java-JSON Deserialization	5
Outro	7
Workbook	7

About the Book

Due to the popularity of the Retrofit blog post series published on our Future Studio blog, and the positive feedback on our Retrofit book, we've decided to publish a book on Gson. A vast majority of our Retrofit readers also use Gson to do the JSON-Java parsing, but struggle with the details. This book is designed to give you an in-depth learning experience for Gson. We specifically tailored it for every Java application and not just Retrofit developers. This book will save you a ton of time researching and avoid stressful evenings of bug fixes. If you value your time, this is something for you.

We cover almost every aspect of Gson with explanations & specific code snippets. This book is for beginners and advanced readers as well. We'll walk you through each topic with direct reference to code examples. Once you've worked through this book, you'll have an extensive knowledge of Gson.

What Topics Are Covered in This Book?

The list below provides a comprehensive overview of covered topics within the book.

- Introduction to JSON & JSON-Java mapping
- **Quick start** to jump right into Gson
- Mapping of **any Java data type**
- Deep dive into **Gson's configuration & customization** options
- Extensive **Custom Serializer & Custom Deserializer**
- **Advanced topics** (Streaming, Generics, ...)
- Necessary **ProGuard** configuration for Android app releases
- **Practical Gson** applications (Storing objects in SharedPreferences, Retrofit, ...)
- Helpful appendix with an overview of Gson alternatives

Who Is This Book For?

This book is for Android or Java developers who want to get a substantial overview and reference book for Gson. You'll benefit from the clearly recognizable code examples in regard to your daily work with Gson.

Rookie

If you're just starting out with Gson (or coming from any other Java-JSON library like Jackson databind) this book will show you all important parts on how to work with Gson. The provided code snippets let you jumpstart and create the first mapping within minutes.

Expert

You already worked with Gson before? You'll profit from our extensive code snippets and can improve your existing code base. Additionally, the book illustrates various optimizations for an even better performance.

Code Examples

Every code snippet in this book has been tested. We provide our test project as a Leanpub extra. Feel free to download it and check the full code base as you move through the book. We try to keep the code snippets in the book as short as possible. If more context is necessary, the code examples should provide it.

Workbook

For the first time, we provide additional separated resources besides our book. The workbook gives you a chance to implement everything you'll learn in this book yourself. In our experience actually using a library like Gson helps to reinforce the newly gained knowledge. Thus, we've designed the workbook, which gives you a guided playground to experiment and learn the features of Gson.

You'll find the workbook on [leanpub](https://leanpub.com/gson-workbook)¹:



Get the Workbook for Free

If you bought the full package, please use the coupon code in your book extras to get the workbook **for free**.

We hope you're excited to get started. Let's jump in!

¹<https://leanpub.com/gson-workbook>

Chapter 1 — Introduction to Java-JSON Mapping

After publishing close to 50 [Retrofit](https://futurestud.io/tutorials/tag/retrofit)² tutorial posts and a [180+ page book](https://leanpub.com/retrofit-love-working-with-apis-on-android)³ we've gathered enough feedback that a lot of developers would love an extensive introduction to Google's Gson. Gson is a very powerful library to map data structures represented as JSON to Java objects. Of course, it also supports the other way around and can create an appropriate JSON representation of your Java objects.

Adding Gson as Dependency

This guide will be hands-on and start with some serialization in a minute. Since a lot of our readers are Android developers, we'll tailor it for you, but Gson can be used in any Java environment. Before we can get started, we need to pull in the Gson library to our project. At the time of writing, the latest version is 2.8.5. If you're using Gradle, add the following line to your `build.gradle`:

```
1 dependencies {  
2     implementation 'com.google.code.gson:gson:2.8.5'  
3 }
```

If you're using Maven, you can add the following dependency:

```
1 <dependencies>  
2     <dependency>  
3         <groupId>com.google.code.gson</groupId>  
4         <artifactId>gson</artifactId>  
5         <version>2.8.5</version>  
6         <scope>compile</scope>  
7     </dependency>  
8 </dependencies>
```

For the poor guys using neither dependency management system, you can download the jar at [the official GitHub repository](https://github.com/google/gson)⁴.

²<https://futurestud.io/tutorials/tag/retrofit>

³<https://leanpub.com/retrofit-love-working-with-apis-on-android>

⁴<https://github.com/google/gson>



JSON Members, Properties & Fields

In this book we'll use the Java terms members, properties and fields interchangeably. While there are [subtle](#)⁵ differences between them, it's not relevant to Gson. Generally, Gson requires at least a field to do any mapping (but it doesn't need to be a property). In the context of the book, a member is always a data field of a class.

⁵<http://docs.oracle.com/javase/tutorial/information/glossary.html>

Basics of Java-JSON Serialization

So let's do some serialization! Serialization in the context of Gson means mapping a Java object to its JSON representation. In the next few chapters our data is going to get more complex, but for now we'll start with a very plain `UserSimple` object:

```
1 public class UserSimple {  
2     String name;  
3     String email;  
4     int age;  
5     boolean isDeveloper;  
6 }
```

The user object has four properties:

- The user's `name` is a `String` object
- The user's `email` is also a `String` object
- The user's `age` is an integer, meaning the age is stored in years (for example 26 and not the exact birthday!)
- Finally a boolean flag `isDeveloper`

Our Android or Java application needs to convert a `UserSimple` object to its JSON representation. Assuming we keep the field names the same, we would expect such a JSON for Norman, the main author of this book:

```
1 {  
2     "name": "Norman",  
3     "email": "norman@futurestud.io",  
4     "age": 26,  
5     "isDeveloper": true  
6 }
```

So let's see how we can do the conversion with Gson. First of all, we need to create a Java object for Norman:

```
1 UserSimple userObject = new UserSimple(  
2     "Norman",  
3     "norman@futurestud.io",  
4     26,  
5     true  
6 );
```

In order to do the serialization, we need a `Gson` object, which handles the conversion. We can simply use the empty constructor:

```
1 Gson gson = new Gson();
```

Next, we need to call the function `toJson()` and pass the `UserSimple` object:

```
1 String userJson = gson.toJson(userObject);
```

The `userJson` object contains the following `String`:

```
1 {  
2     "age": 26,  
3     "email": "norman@futurestud.io",  
4     "isDeveloper": true,  
5     "name": "Norman"  
6 }
```

`Gson` changed the order of the properties (to alphabetically), but the content is identical as our hand-crafted JSON! Note how `Gson` respected the types. String values were wrapped in `"`, while integer values had no wrapping. We didn't have to mess around with JSON objects or copy single fields. A single call to `Gson` was enough to map the entire object. This comes in extremely handy when we're working with complex data structures.



Workbook Task A1

You'll find an exercise for basic serialization in the workbook at A1.

But before going too deep, let's test the other direction. Can `Gson` create a Java object from the above JSON?

Basics of Java-JSON Deserialization

First of all, we need to create a String, which contains the above-mentioned JSON:

```
String userJson = "{ 'age':26, 'email': 'norman@futurestud.io', 'isDeveloper':true, 'name': 'Norman' }";
```



JSON Strings in Java

We substituted the quotation marks " with apostrophes ' to avoid tons of \ Java escaping. While the JSON standard defines that JSON properties are wrapped in quotation marks "", we use this short cut to make the code more readable. Additionally, Gson accepts both styles anyway.

The next step is, as you probably have guessed, to create a Gson instance:

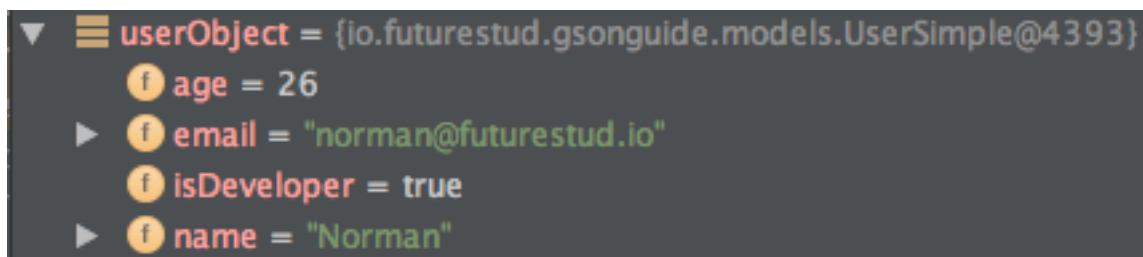
```
1 Gson gson = new Gson();
```

Finally, we've to map from a JSON to a Java object with `fromJson()`:

```
1 UserSimple userObject = gson.fromJson(userJson, UserSimple.class);
```

Note how we've to pass the class of the expected Java object as the second parameter. Otherwise Gson doesn't know what it should map the JSON to. It's not a magician!

If we attach a debugger and check the resulting `userObject` it'll show us that Gson successfully mapped all properties correctly:



Mapped User Java Object from JSON String



Workbook Task A2

You'll find an exercise for basic deserialization in the workbook at A2.

Chapter Summary

In this getting started chapter you've seen the basic usage of Gson. We've shown how easy it is to map from and to JSON data. We're sure you've a bunch of questions at the moment:

- Do the Java model classes need a constructor/getter/setter?
- Can the Java model fields be private?
- How do null values get handled?
- What if the Java class has different field naming than the JSON?
- How to (de)serialize nested objects?
- How to (de)serialize arrays/lists of objects?
- Does Gson keep a defined default value for a class property while executing `.fromJson()` and there isn't any value for that property available in JSON?

No worries, we'll get into that (and much more) in the following chapters! The second chapter will show you how to map all kinds of data structures.

Additional Chapter Resources

- [Gson's Github Repository](https://github.com/google/gson)⁶

⁶<https://github.com/google/gson>

Outro

Our goal is to truly help you getting started and ultimately master Gson. We hope you learned many new things throughout this book. We want you to save time while learning the basics and details about Gson. The existing Gson documentation lacks various information and this book should help you to gain extensive in-depth knowledge without loosing time searching StackOverflow for correct answers.

Nonetheless, we'll update the content of this book to future Gson versions as new releases become available. However, it will take some time to update everything, so please be patient. Of course, if you've activated the email updates on Leanpub, we'll let you know about any book updates.

As a reader of this book, we'll always reward your loyalty by publishing exclusive content and any future update will —of course— be free for you! For example, the readers of our Retrofit 1 book got the complete rewrite and update to Retrofit 2 for free, even though the price increased and content of the book doubled.

For us it's really important to exceed our reader's expectations. In all our products and guides we aim for a high quality. If you feel like a section or chapter in this book wasn't clear or extensive enough, please let us know at norman@futurestud.io⁷. We always love hearing back from you, so if you have anything to say, don't hesitate to shoot us an email. We welcome any feedback, critic, suggestions for new topics or whatever is currently on your Gson mind!

This book includes an Android project with code examples for almost all the snippets you've seen in this book. We recommend to download the code package from this book's Leanpub extras and give it a spin. It contains all code examples used within this book and additionally sets you in the context of where to use them.

Workbook

Additionally, if you bought the full package, you should download the workbook and cheat sheet! Reading and understanding features of Gson is just the first step of learning. Actually using them is necessary to really keep everything in your brain. You've invested a lot of time by reading this book, so don't stop here and keep going the last few meters to the peak.

⁷<mailto:norman@futurestud.io>

We are big proponents of fair use and value. Feel free to copy any code snippets from the book or the Android project into your apps. You don't have to mention or link back to us. We value comments, emails and tweets if you benefit from the book or code. Those messages keep us motivated and everybody loves kind words of appreciation :)



Talk to Us

- Follow us on Twitter: [@futurestud_io](https://twitter.com/futurestud_io)⁸
- Watch our training videos in 4k resolution: [YouTube](https://www.youtube.com/channel/UCWIA0jrjry_aEd9u6JTe1S9Q)⁹
- Email us: norman@futurestud.io¹⁰
- Our website provides valuable tutorials: futurestud.io/tutorials¹¹

Don't forget, we're publishing new tutorials at least every Thursday, mainly about Android and Node.js within the Future Studio University. Feel free to visit our homepage and the University :)

Thanks a lot for reading this book! We truly appreciate your interest and hope you learned a lot from reading this book! <3

— Norman, Christian & Marcus

⁸http://twitter.com/futurestud_io

⁹https://www.youtube.com/channel/UCWIA0jrjry_aEd9u6JTe1S9Q

¹⁰<mailto:norman@futurestud.io>

¹¹<https://futurestud.io/tutorials>