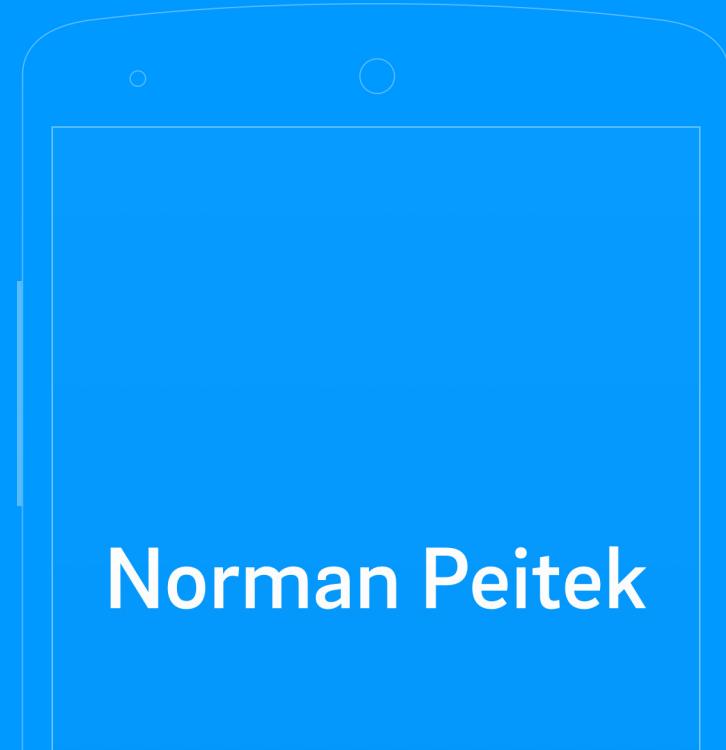




Gson

Practical Workbook



Norman Peitek

Gson Workbook

Norman Peitek

This book is for sale at <http://leanpub.com/gson-workbook>

This version was published on 2016-10-31



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 Future Studio

Also By Norman Peitek

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

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

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

[Gson: Enjoy JSON \(De-\)Serialization in Java](#)

Contents

About the Workbook	i
What Topics Are Covered in This Book?	i
Chapter 0 — Workbook Basics	1
Task Structure	2
Chapter 1 — Introduction to Java-JSON Mapping	3
A1: Basic Serialization	3
A2: Basic Deserialization	5
Outro	7

About the Workbook

Due to the popularity of the Retrofit tutorial 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.

We've published a [book¹](#) on Gson, which is designed to give you an in-depth learning experience. We specifically tailored it for every Java application and not just Retrofit developers. This workbook is an practical addon to the theoretical learning of the book.

The idea of this workbook is three-fold:

- Give the readers of the Gson book a specific playground to practice the learned theory
- Let experienced Gson users test their knowledge
- Offer a way to learn Gson for people who don't like reading technical books and rather experiment on their own.

We cover a lot of aspects of Gson in this workbook. We've roughly 40 tasks for you. If you wish, you can use our project framework to solve the tasks of this workbook. Automated unit tests will tell you if your implementation for that particular task was correct. Of course, every task comes with a sample solution.

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
- Mapping of **various Java data type**
- Dive into **Gson's configuration & customization** options
- Multiple tasks on **Custom Serializer & Custom Deserializer**
- Six **Advanced topics** (Streaming, Low-Level-Parsing, ...)

¹<https://leanpub.com/gson-enjoy-json-deserialization-in-java>



Download the Project Framework

In order to save time and give you immediate feedback, we've created a Java project, which prepares everything for you. Download it on [leanpub.com](https://leanpub.com/gson-enjoy-json-deserialization-in-java/extras) in the [extras section²](#) of this book.

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

²<https://leanpub.com/gson-enjoy-json-deserialization-in-java/extras>

Chapter 0 — Workbook Basics

Welcome to our Gson workbook! Before we get started with the first task, let's make sure you're all set up and ready to go.

If you want, you can just solve the tasks in this workbook in your mind or in your own little Gson playground. However, we recommend to use the provided project (download on [leanpub.com](https://leanpub.com/futurestud/gson-workbook) under book extras).

Java Learning Project

It'll provide a skeleton for each task, so you only have to fill in the important bits. Additionally, we've written unit tests for each task. A passing unit test for the task you've worked on indicates that your implementation is correct.



Incorrect Unit Test?

We've reviewed all the tests for the tasks, but they're probably a few scenarios we've missed. If the unit tests fails where it shouldn't be failing (or vice versa), please [drop me an email](mailto:norman@futurestud.io)³. We appreciate it!

The project is a gradle based Java project. Please make sure you've Java and Gradle installed on your machine. We've developed and tested it with JetBrains' [IntelliJ](#)⁴.

If you're running into any technical issues, don't hesitate to [contact us](#)⁵.

The project is structured into three parts:

- Tasks
- Solutions
- Tests

The **tasks** section is where you will spend most of your time. It gives you the code skeleton so you only have to add a few lines of Gson specific code.

The **solutions** provide one way of solving each problem. Please understand that there are often multiple ways and you don't have to use the one in the solution!

Finally, we provide **tests** for each task, which check if your implementation is correct. You shouldn't need to make any changes there.

³ <mailto:norman@futurestud.io>

⁴ <https://www.jetbrains.com/idea/>

⁵ <mailto:norman@futurestud.io>

Task Structure

Almost all tasks have a very similar structure. It gives you an empty `implement` method, which you'll need to fill with life. The parameter and return type are already correct. There is no need to change those!

Each task also comes with one or more Java model classes. We'll indicate if it's necessary to make changes to the models. The models are always specific to the task. Thus, if you change the model in task A1, it'll have no influence to other tasks.

Chapter 1 — Introduction to Java-JSON Mapping

It finally is time to get working on some actual problems. We'll start easy and only go through the simple scenarios of mapping some small Java objects from and to JSON. Let's start!

A1: Basic Serialization

Welcome to your first task in chapter 1!

In this task you'll practice serialization. You'll need to use Gson to create a JSON from a Java object. We already provided you with the input class A1.BookModel.

We suggest the following steps:

1. Familiarize yourself with the model class A1.BookModel
2. Create a new Gson (default) instance
3. Do the conversion from the A1.BookModel object to a JSON string with the Gson instance

```
1  public class A1 implements TaskSerializationInterface<A1.BookModel> {  
2  
3      public String implementTask(A1.BookModel inputModel) {  
4          // todo implement step 2 & 3 here  
5          return null;  
6      }  
7  
8      // todo do step 1 here  
9      // There is no need to touch the models!  
10     // Otherwise the automated testing might break.  
11     public static class BookModel extends ModelBase {  
12         public String title;  
13         public String author;  
14         public float price;  
15     }
```

```
16     public BookModel(String title, String author, float price) {  
17         this.title = title;  
18         this.author = author;  
19         this.price = price;  
20     }  
21 }  
22 }
```

A2: Basic Deserialization

Welcome to your second task in chapter 1!

In this task you'll practice deserialization. In other words you'll need to use Gson to read a JSON input and convert it to a Java object.

You'll receive a JSON which has four properties:

- String name
- String email
- String status
- boolean paid

Example JSON:

```
1  {
2      "name": "Norman Peitek",
3      "email": "norman@futurestud.io",
4      "status": "fulltime",
5      "paid": true
6  }
```

We suggest the following steps:

1. You need to fill the `SubscriberModel` with some life based on the example JSON we provided above
2. Create a default Gson instance in the `implementTask()` method
3. Do the actual conversion from the `inputJson` String to a `SubscriberModel` object in the `implementTask()` method.

```
1  public class A2 implements TaskDeserializationInterface<A2.SubscriberModel> {
2
3      public SubscriberModel implementTask(String inputJson) {
4          // todo do step 2 & 3 here
5          return null;
6      }
7
8      // todo do step 1 here
9      // You need to create the model properties!
10     // There is no need to change the class name.
```

```
11  public class SubscriberModel extends ModelBase {  
12  
13  }  
14 }
```

Outro

Our goal is to truly help you getting started and ultimately master Gson. We hope this workbook was helpful for you to solidify your knowledge in Gson.

Nonetheless, we'll update the content of this workbook 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 sharing on Leanpub, we'll let you know about any updates.

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 task in this workbook wasn't clear or extensive enough, please let me know at norman@futurestud.io⁶.

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!



Talk to Us

- Follow us on Twitter: [@futurestud_io](https://twitter.com/futurestud_io)⁷
- Watch our training videos in 4k resolution: [YouTube](https://www.youtube.com/channel/UCWIA0jryy_aEd9u6JTe1S9Q)⁸
- Email us: norman@futurestud.io⁹
- Our website provides valuable tutorials: futurestud.io/tutorials¹⁰

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

— Norman, Christian & Marcus

⁶ <mailto:norman@futurestud.io>

⁷ [http://twitter.com/futurestud_io](https://twitter.com/futurestud_io)

⁸ https://www.youtube.com/channel/UCWIA0jryy_aEd9u6JTe1S9Q

⁹ <mailto:norman@futurestud.io>

¹⁰ <https://futurestud.io/tutorials>