



# JSON

DeepDive Week 1 day 4 & week 2 Day 0





## Материалы и Ресурсы

Наш framework, который позволяет создавать REST API, будет использовать JSON для основного формата представления данных. Соответственно, нам нужно написать логику сериализации и десериализации из/в JSON. Для начала стоит изучить:

- что такое [REST API](#)
- а также, [что такое JSON](#)
- и что такое [рефлексия в Java](#)

У меня немного хороших новостей. Мы НЕ будем писать парсер полностью поддерживающий формат JSON (RFC 7159). Мы напишем парсер для части JSON, а именно, наш JSON будет работать только с такими типами данных как:

- String
- int
- null
- SimpleClass

SimpleClass - это Java класс, у которого полями являются только String или только int.

Например:

```
public static class SimpleOneStringClass {  
    public String one = "_ololo1_";  
}
```

Единственное исключение! Для преобразователя из Java объекта в JSON он должен уметь еще и преобразовать массивы в JSON. **Обратное делать не нужно.**

## JsonHelper

API сегодняшнего дня очень простой:

```
package academy.kovalevskiy.javadeepdive.week2.day0;

import java.lang.reflect.Array;
import java.lang.reflect.InvocationTargetException;

public class JsonHelper {

    public static <T> String toJsonString(T target) {
        // TODO
    }

    public static <T> T fromJsonString(String json, Class<T> cls)
        throws IllegalAccessException, InvocationTargetException,
        InstantiationException,
        NoSuchFieldException, NoSuchMethodException {
        // TODO
    }
}
```

Если кажется, что у вас много свободного времени, то начните делать полноценный и нормальный парсер JSON в обе стороны. Однако учтите, что тестами покрыта только часть, которая требуется в задании.

При выполнении задания нужно провести BCE проверки на null!