



Parsing JSON from Java

Using `javax.json` and `org.json`



What is “parsing”?

A “parser” is a component which takes some input and turns it into some datastructure like an object or a tree etc. It allows us to check for validity and gives us a structured representation of the input.

So, take some text, like JSON, build a tree or some other data type from it, and check that it is well-formed (syntactically correct).

When do we need to “parse” JSON?

If our Java application gets data from some service or file in the form of a JSON document, we need to transform it from application/JSON to some kind of representation in Java objects.

The goal is to get the JSON from somewhere and create a representation of that data in Java.

So we need to “parse” the JSON and traverse the tree and create some Java data type object from it (like a List for instance).

In this simple example, we'll just get the data out from the JSON object, piece-by-piece.

Example with an address book entry

Let's pretend our application gets an address book entry in the form of JSON.

We then need to “parse” the JSON in order to get the data from the document.

The JSON file

```
{
  "firstName": "Duke",
  "lastName": "Java",
  "age": 18,
  "streetAddress": "100 Internet Dr",
  "city": "JavaTown",
  "state": "JA",
  "postalCode": "12345",
  "phoneNumbers": [
    { "Mobile": "111-111-1111" },
    { "Home": "222-222-2222" }
  ]
}
```

Parsing using javax.json

We'll need to have the API with javax.json in a JAR file on our class path, so that we can use the interfaces and classes in

<https://docs.oracle.com/javaee/7/api/javax/json/package-summary.html>

The steps are:

- Create a JsonReader (from the file or stream)
- Create a JsonStructure from the reader
- Convert the structure to a JsonObject
- Get the values
- Get the JsonArray using getJsonArray("name of the array")
- Loop through the array and get the values

Code example - first steps

```
JsonReader reader = Json.createReader( new FileReader(jsonFile) );
JsonStructure jsonStruct = reader.read();
if(jsonStruct.getValueType().equals(OBJECT)){
    System.out.println("Casting to JsonObject...");
    JsonObject jo = (JsonObject) jsonStruct;
    ...
}
```

Code example - getting the values

```
System.out.println("First name: " + jo.getString("firstName"));
System.out.println("Last name: " + jo.getString("lastName"));
System.out.println("Age: " + jo.getInt("age"));
System.out.println("Street address: " + jo.getString("streetAddress",
                                                    "No st. addr.));
System.out.println("City: " + jo.getString("city", "No city"));
System.out.println("State: " + jo.getString("state", "No state"));
System.out.println("Postal code: " +
                   jo.getString("postalCode", "No postal code"));
```

```
"firstName": "Duke",
"lastName": "Java",
"age": 18,
"streetAddress": "100 Internet Dr",
"city": "JavaTown",
"state": "JA",
"postalCode": "12345"
```


Code example - the array

```
JSONArray arr = jo.getJSONArray("phoneNumbers");
System.out.println("Phone numbers:");
for(JsonValue jv : arr){
    /* JsonObject is a Map - check if it has the correct key
    */
    if( ((JsonObject)jv).keySet().contains("Mobile")){
        String mobile = ((JsonObject)jv).getString("Mobile");
        if(mobile!=null){
            System.out.println(" Mobile: " + mobile);
        }
    }else if(((JsonObject)jv).keySet().contains("Home")){
        String home = ((JsonObject)jv).getString("Home");
        if(home!=null){
            System.out.println(" Home: " + home);
        }
    }
}
```

```
"phoneNumbers": [
{
    "Mobile": "111-111-1111"
},
{
    "Home": "222-222-2222"
}
]
```

More complete code - import statements

```
import javax.json.Json;  
import javax.json.JsonArray;  
import javax.json.JsonValue;  
import javax.json.JsonObject;  
import static javax.json.JsonValue.ValueType.*;  
import javax.json.JsonReader;  
import javax.json.JsonStructure;
```

More complete code

```
public class JSONParser{
    private String jsonFile;
    public JSONParser(String jsonFile){
        this.jsonFile = jsonFile;
    }
    public void parseAndPrint()throws FileNotFoundException{
        try{
            JsonReader reader = Json.createReader( new FileReader(jsonFile) );
            JsonStructure jsonStruct = reader.read();
            if(jsonStruct.getValueType().equals(OBJECT)){
                System.out.println("Casting to JsonObject...");
                JsonObject jo = (JsonObject) jsonStruct;
                System.out.println("First name: " + jo.getString("firstName"));
                System.out.println("Last name: " + jo.getString("lastName"));
                System.out.println("Age: " + jo.getInt("age"));
                System.out.println("Street address: " + jo.getString("streetAddress", "No st. addr.));
                System.out.println("City: " + jo.getString("city", "No city"));
                System.out.println("State: " + jo.getString("state", "No state"));
                System.out.println("Postal code: " + jo.getString("postalCode", "No postal code"));
            }
        }
    }
}
```

More complete code

```
JSONArray arr = jo.getJSONArray("phoneNumbers");
System.out.println("Phone numbers:");
for(JsonValue jv : arr){
    /* JsonObject is a Map - check if it has the correct key */
    if( ((JsonObject)jv).keySet().contains("Mobile")){
        String mobile = ((JsonObject)jv).getString("Mobile");
        if(mobile!=null){
            System.out.println(" Mobile: " + mobile);
        }
    }else if(((JsonObject)jv).keySet().contains("Home")){
        String home = ((JsonObject)jv).getString("Home");
        if(home!=null){
            System.out.println(" Home: " + home);
        }
    }
}
}
}
}
}
}
}
}
}
}
}
}
```

Parsing using org.json

Steps:

- Create a JSONTokener from the JSON string
- Create a JSONObject from the tokenizer
- Get the values from the JSONObject
- Get the array
 - loop through the array
 - get the values

Code - first steps

```
JSONTokener jt = new JSONTokener(sb.toString());
JSONObject jo = new JSONObject(jt);
System.out.println("First name: " + jo.getString("firstName"));
System.out.println("Last name: " + jo.getString("lastName"));
System.out.println("Age: " + jo.getInt("age"));
System.out.println("Street address: " + jo.getString("streetAddress"));
System.out.println("City: " + jo.getString("city"));
System.out.println("State: " + jo.getString("state"));
System.out.println("Postal code: " + jo.getString("postalCode"));
```

Code - the array

```
JSONArray arr = jo.getJSONArray("phoneNumbers");
System.out.println("Phone numbers:");
for(int i=0;i<arr.length();i++){
    jo = arr.getJSONObject(i);
    if(jo.has("Mobile") && !jo.isNull("Mobile")){
        System.out.println("Mobile: " + jo.get("Mobile"));
    }
    if(jo.has("Home") && !jo.isNull("Home")){
        System.out.println("Home: " + jo.get("Home"));
    }
}
```

More complete code

```
import org.json.*; // JSONObject, JSONTokener, JSONArray
try{
    JSONTokener jt = new JSONTokener(sb.toString());
    JSONObject jo = new JSONObject(jt);
    System.out.println("First name: " + jo.getString("firstName"));
    System.out.println("Last name: " + jo.getString("lastName"));
    System.out.println("Age: " + jo.getInt("age"));
    System.out.println("Street address: " + jo.getString("streetAddress"));
    System.out.println("City: " + jo.getString("city"));
    System.out.println("State: " + jo.getString("state"));
    System.out.println("Postal code: " + jo.getString("postalCode"));
}
```


More complete code

```
JSONArray arr = jo.getJSONArray("phoneNumbers");
System.out.println("Phone numbers:");
for(int i=0;i<arr.length();i++){
    jo = arr.getJSONObject(i);
    if(jo.has("Mobile") && !jo.isNull("Mobile")){
        System.out.println("Mobile: " + jo.get("Mobile"));
    }
    if(jo.has("Home") && !jo.isNull("Home")){
        System.out.println("Home: " + jo.get("Home"));
    }
}
}catch(JSONException e){
    System.err.println("Couldn't parse: " + e.getMessage());
}
}
```

Links

javax.json

<https://docs.oracle.com/javaee/7/api/javax/json/package-summary.html>

<https://mvnrepository.com/artifact/org.glassfish/javax.json/1.0.4>

org.json

<https://mvnrepository.com/artifact/org.json/json/1.5-20090211>

<https://developer.android.com/reference/org/json/package-summary.html>