The json.simple library is a lightweight and simple API for JSON parsing and encoding in Java. It is especially useful for Java applications that require basic JSON functionality without heavy dependencies.

Here’s a summary of the json.simple library’s key classes and methods, along with a few examples.

**Key Classes in json.simple**

1. **JSONObject**: Represents a JSON object, which is a collection of key-value pairs.
2. **JSONArray**: Represents a JSON array, which is an ordered list of values.
3. **JSONParser**: Parses JSON data into Java objects.

**JSONObject: Methods and Usage**

The JSONObject class provides a simple way to create and manipulate JSON objects.

* **Creating a JSON object**:

java

Copy code

JSONObject jsonObject = new JSONObject();

**Common Methods in JSONObject**

| **Method** | **Description** |
| --- | --- |
| put(Object key, Object value) | Adds a key-value pair to the JSON object. |
| get(Object key) | Retrieves the value associated with a specific key. |
| remove(Object key) | Removes a key-value pair by key. |
| size() | Returns the number of key-value pairs in the JSON object. |
| isEmpty() | Checks if the JSON object is empty. |
| containsKey(Object key) | Checks if a specific key exists in the JSON object. |
| containsValue(Object value) | Checks if a specific value exists in the JSON object. |
| toJSONString() | Converts the JSON object to a JSON string. |
| toString() | Returns a JSON string representation of the object. |

**Example Usage of JSONObject**

java

Copy code

import org.json.simple.JSONObject;

public class Main {

public static void main(String[] args) {

JSONObject jsonObject = new JSONObject();

jsonObject.put("name", "Alice");

jsonObject.put("age", 25);

jsonObject.put("isStudent", true);

System.out.println(jsonObject.toJSONString());

}

}

Output:

json

Copy code

{"name":"Alice","age":25,"isStudent":true}

**JSONArray: Methods and Usage**

The JSONArray class represents a JSON array, which can contain a list of elements, each of which can be any data type.

* **Creating a JSON array**:

java

Copy code

JSONArray jsonArray = new JSONArray();

**Common Methods in JSONArray**

| **Method** | **Description** |
| --- | --- |
| add(Object value) | Adds an element to the array. |
| get(int index) | Retrieves the element at a specific index. |
| remove(int index) | Removes an element at a specific index. |
| size() | Returns the number of elements in the array. |
| isEmpty() | Checks if the array is empty. |
| toJSONString() | Converts the array to a JSON string. |
| toString() | Returns a JSON string representation of the array. |

**Example Usage of JSONArray**

java

Copy code

import org.json.simple.JSONArray;

public class Main {

public static void main(String[] args) {

JSONArray jsonArray = new JSONArray();

jsonArray.add("Java");

jsonArray.add("Python");

jsonArray.add("JavaScript");

System.out.println(jsonArray.toJSONString());

}

}

Output:

json

Copy code

["Java","Python","JavaScript"]

**JSONParser: Methods and Usage**

The JSONParser class is used to parse JSON strings into Java objects (JSONObject or JSONArray).

* **Creating a JSON parser**:

java

Copy code

JSONParser parser = new JSONParser();

**Common Methods in JSONParser**

| **Method** | **Description** |
| --- | --- |
| parse(String s) | Parses a JSON string and returns a JSONObject or JSONArray. |
| parse(Reader r) | Parses JSON data from a Reader and returns a JSONObject or JSONArray. |

**Example Usage of JSONParser**

java

Copy code

import org.json.simple.JSONObject;

import org.json.simple.parser.JSONParser;

import org.json.simple.parser.ParseException;

public class Main {

public static void main(String[] args) {

String jsonString = "{\"name\":\"Alice\",\"age\":25,\"isStudent\":true}";

JSONParser parser = new JSONParser();

try {

JSONObject jsonObject = (JSONObject) parser.parse(jsonString);

System.out.println("Name: " + jsonObject.get("name"));

System.out.println("Age: " + jsonObject.get("age"));

System.out.println("Is Student: " + jsonObject.get("isStudent"));

} catch (ParseException e) {

e.printStackTrace();

}

}

}

**Adding json.simple to Your Project**

You can add json.simple via Maven or download the .jar file and add it as a library in your project.

**Maven Dependency**:

xml

Copy code

<dependency>

<groupId>com.googlecode.json-simple</groupId>

<artifactId>json-simple</artifactId>

<version>1.1.1</version>

</dependency>

This should give you a comprehensive overview of json.simple methods and how to use them in your project!