

Set 1

JSON-java

- `boolean similar(Object other)` - `JSONObject.java`
- `Writer write(Writer writer, int indentFactor, int indent)` - `JSONObject.java`
- `JSONArray(JSONTokener x)` - `JSONArray.java`
- `String unescape(String string)` - `XML.java`

json-iterator

- `Decoder createDecoder(String cacheKey, Type type)` - `extra/GsonCompatibilityMode.java`
- `void encode_(Object obj, JsonStream stream)` - `output/ReflectionObjectEncoder.java`
- `void enableDecoders()` - `extra/Base64FloatSupport.java`
- `boolean equals(Object o)` - `extra/GsonCompatibilityMode.java`

Set 2

JSON-java

- String toString(JSONArray ja) - JSONML.java
- BigInteger objectToBigInteger(Object val, BigInteger defaultValue) - JSONObject.java
- JSONArray(JSONTokenizer x) - JSONArray.java
- Float getFloat(int index) – JSONArray.java

json-iterator

- Object read() - JsonIterator.java
- int parse(JsonIterator iter) - IterImplString.java
- boolean equals(Object o) - extra/GsonCompatibilityMode.java
- void skipFixedBytes(JsonIterator iter, int n) - IterImplForStreaming.java

Set 3

JSON-java

- String toString(JSONArray ja) - JSONML.java
- String unescape(String string) - XML.java
- void populateMap(Object bean) - JSONObject.java
- JSONObject toJSONObject(String string) - Cookie.java

json-iterator

- void enableDecoders() - extra/Base64FloatSupport.java
- void writeRaw(String val, int remaining) - output/JsonStream.java
- Any fillCacheUntil(int target) - any/ArrayLazyAny.java
- Object read() - JsonIterator.java