

json

Reading and Writing JSON

2.2.3

21 June 2025

Christopher Jefferson

Christopher Jefferson

Email: caj21@st-andrews.ac.uk

Homepage: <https://heather.cafe/>

Address: St Andrews

Scotland

UK

Contents

1	GAP-JSON mapping	3
1.1	Methods	3
	Index	5

Chapter 1

GAP-JSON mapping

This package defines a mapping between the JSON markup language and GAP. The built-in datatypes of GAP provide an easy mapping to and from JSON. This package uses the following mapping between GAP and JSON.

- JSON lists are mapped to GAP lists
- JSON dictionaries are mapped to GAP records
- JSON strings are mapped to GAP strings
- Integers are mapped to GAP integers, non-integer numbers are mapped to Floats
- true, false and null are mapped to true, false and fail respectively

Note that this library is **NOT** intended to provide a general purpose library for transmitting any GAP object. If you wish to do this, look at the `openmath` package, or `IO_Pickle` in the `IO` package.

1.1 Methods

1.1.1 GapToJsonStream

▷ `GapToJsonStream(stream, value)` (function)

Converts the *value* to JSON, and outputs it to *stream*. This function disables GAP's usual line splitting while JSON is being outputted.

1.1.2 GapToJsonString

▷ `GapToJsonString(value)` (function)

Returns: string

Converts a GAP *value* to a JSON string.

1.1.3 JsonStringToGap

▷ `JsonStringToGap(string)` (function)

Returns: value

Converts a JSON *string* into a GAP value.

1.1.4 JsonStreamToGap

▷ `JsonStreamToGap(stream)`

(function)

Returns: value

Reads a single JSON object from a *stream* and converts it to a GAP value.

Index

GapToJsonStream, [3](#)

GapToJsonString, [3](#)

JsonStreamToGap, [4](#)

JsonStringToGap, [3](#)