ADAMS

Advanced **D**ata mining **A**nd **M**achine learning **S**ystem

Module: adams-json

Peter Reutemann

January 10, 2024

Contents

Chapter 1

Flow

Flows cannot only be saved as JSON[?] files (and read in again), but flows themselves can process JSON data structures as well.
The following sources are available:

- *NewJsonStructure* – for creating empty JSON data structures like objects or arrays.

The following transformers are available:

- *DeleteJsonValue* – removes the specified key/value pair from the JSON object passing through.
- *GetJsonKeys* – outputs all named elements of a JSON object.
- *GetJsonValue* – outputs the named value from a JSON object, can use simple key or a JSON path[2].
- *JsonFileReader* – reads the specific JSON file and forwards a JSON object/array.
- *SetJsonValue* – stores a value in a JSON object, can use simple key or a JSON path[2].

The following sinks are available:

- *JsonDisplay* – displays a JSON object in a browseable tree structure.
- *JsonFileWriter* – writes the JSON object/array to disk.

The following conversion are available:

- *ArrayToJsonArray* – generates a JSON array from any object array.
- *ListToJsonArray* – generates a JSON array from a list.
- *JsonArrayToArray* – turns a JSON array into a regular Java object array.
- *JsonArrayToList* – turns a JSON array into a regular Java list.
- *JsonObjectToMap* – turns the JSON object into a simple Map.
- *JsonToSpreadSheet* – turns the JSON object into a spreadsheet (i.e., flattening it).
- *JsonToString* – turns the JSON object/array into a string.
- *ListToJson* – turns a *java.util.List* object into a JSON object.
- *MapToJson* – turns a *java.util.Map* object into a JSON object.

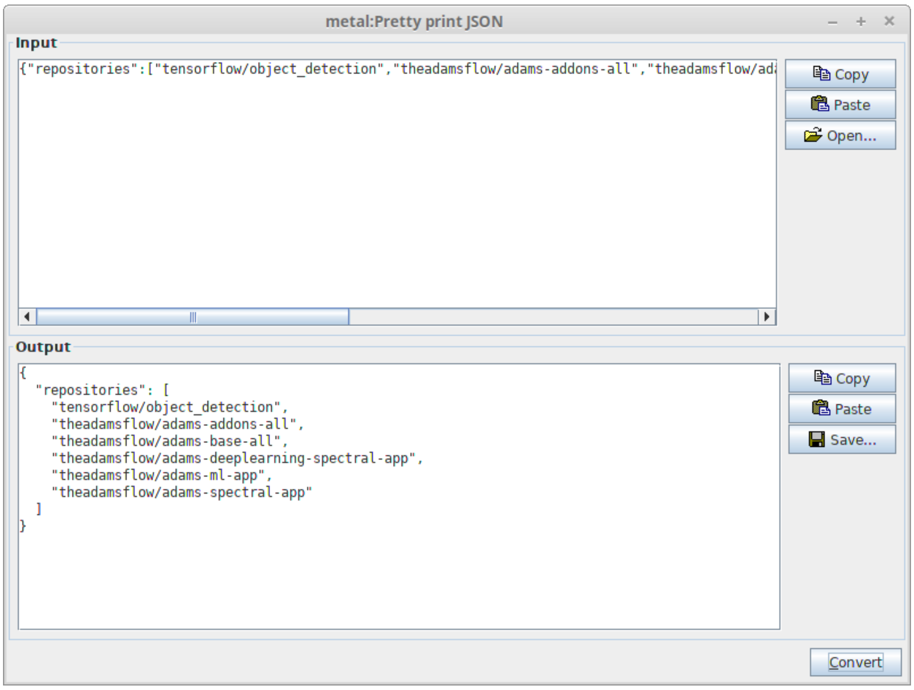
- *ReportToJson* – turns an *adams.data.report.Report* objectinto a JSON object.
- *SpreadSheetToJson* – turns a spreadsheet into a JSON array object.
- *StringToJson* – parses the string and generates a JSON object/array.

Chapter 2

Tools

2.1 Pretty print JSON

In order to preserve space, JSON is often optimized and removes any unnecessary whitespaces. However, for a human to inspect such data, it is much more useful to have it properly indented, aka *pretty printed*. Figure 2.1 shows a screenshot of the *Pretty print JSON* tool that allows you to turn JSON into a more human-readable format.



```
{"repositories":["tensorflow/object_detection","theadamsflow/adams-addons-all","theadamsflow/ad
```

```
{
  "repositories": [
    "tensorflow/object_detection",
    "theadamsflow/adams-addons-all",
    "theadamsflow/adams-base-all",
    "theadamsflow/adams-deeplearning-spectral-app",
    "theadamsflow/adams-ml-app",
    "theadamsflow/adams-spectral-app"
  ]
}
```

Figure 2.1: Pretty print JSON

Bibliography

[1] *ADAMS* – Advanced Data mining and Machine learning System
`https://adams.cms.waikato.ac.nz/`

[2] *Jayway JsonPath* – A Java DSL for reading JSON documents.
`https://github.com/json-path/JsonPath`