V1.0

reducedjson@seantechsol.com

ReducedJSON user's guide

V1.0

reducedjson@seantechsol.com

About the author

I am the director of Seantech solutions Limited, providing consultancy services in the field of development and enhancement of the enterprise solutions and infrastructures of the clients of Seantech solutions limited. I have been providing end2end solutions in the industry for last 12 years in different sectors. I am helping lots my clients and development communities to sort out their very complicated situations when it comes to the mapping of data structures from both consumer and service side. Please feel free to visit my office at www.seantechsol.com.

Kind regards,

Syed S A Naqvi
Software development consultant
00447846426801
syed@seantechsol.com

reducedjson@seantechsol.com

Why do you need to use ReducedJSON?

The trigger behind the development of this library is, I have been unable to find any smart JavaScript framework or a library which can help me to find my target information from JSON data structure without writing a single line of parsing code. Most of the JavaScript and jQuery developers go through the situation where they have to write thousands of lines of code to parse the JSON data in order to find the required information and this has caused a huge over head of the efforts in terms of time and money which ends up the failure of the projects most of the time. This has caused a dire need to develop such a reusable, highly optimised, highly responsive, less resource hungry, efficient and effective JavaScript library which can work with any existing JavaScript solutions and frameworks with cross browser and platform functionality.

- **Configurations**
  - Download or clone the ReducedJSON library
  - Copy the jsoncommand folder into appropriate resource location e.g. inside a JS directory of your web page project
- **Run ReducedJSON library functions to parse the JSON**
  - Adding JS library jsoncommand/jsoncommand.nocache.js
  - Initializing the ReducedJSON object by copying the following JS script into you web page

```
<script type="text/javascript" language="javascript" src="jsoncommand/jsoncommand.nocache.js"></script>

<script>

    var reducedJSON ;
    function initReducedJSON() {
        reducedJSON =  new reducedjson.Reducer();
    }
</script>
```

  - Call the ‘reduce’ function to target your required data

```
<script>
       // output is the result
       var output = reducedJSON.reduce(json,cmd);
       alert("ReducedJSON output → "+output);
</script>
```

reducedjson@seantechsol.com

- **ReducedJSON command build and design**

JSON example:

```
{
  "glossary": {
    "title": "example glossary",
    "GlossDiv": {
      "title": "S",
      "GlossList": {
        "GlossEntry": {
          "ID": "SGML",
          "SortAs": "SGML",
          "GlossTerm": "Standard Generalized Markup Language",
          "Acronym": "SGML",
          "Abbrev": "ISO 8879:1986",
          "GlossDef": {
            "para": "A meta-markup language, used to create markup languages such as DocBook.",
            "GlossSeeAlso": [
              "GML",
              "XML"
            ]
          },
          "GlossSee": "markup"
        }
      }
    }
  }
}
```

- Build and Design

There is a basic principal to follow while building a command, which is to correctly use of '/' before each key in the JSON data. Like in above JSON example the highest parent key is 'glossary' then 'title', so in order to target the title data our command will be as following

```
var cmd = '/glossary/title'
var jsonData = JSON data in above example
var output = reducedJSON.reduce(json,cmd);
```

Live Demo

Let's run the above scenario at **https://reducedjson.github.io/reducedjson/**

- Go to demo page copy the above JSON into **Input area**
- Copy cmd value without quotes like /glossary/title into **Command:**
- Then press '/' key then press enter
- Now check the result into the **Plain output** and **Output areas**

More command examples:

Requirement	Command	Output
To find out GlossDiv data	/glossary/GlossDiv/	{"title":"S", "GlossList":{"GlossEntry":{"ID":"SGML", "SortAs":"SGML", "GlossTerm":"Standard Generalized Markup Language", "Acronym":"SGML", "Abbrev":"ISO

		8879:1986", "GlossDef":{"para":"A meta-markup language, used to create markup languages such as DocBook.", "Gloss
To find out title of GlossDive	/glossary/GlossDiv/title/	"S"
To find out GlossList of GlossDiv	/glossary/GlossDiv/GlossList/	{"GlossEntry":{"ID":"SGML", "SortAs":"SGML", "GlossTerm":"Standard Generalized Markup Language", "Acronym":"SGML", "Abbrev":"ISO 8879:1986", "GlossDef":{"para":"A meta-markup language, used to create markup languages such as DocBook.", "GlossSeeAlso":["GML","XML"]}, "GlossSee":"markup"}}
To find out GlossEntry inside GlossList of GlossDiv	/glossary/GlossDiv/GlossList/GlossEntry	{"ID":"SGML", "SortAs":"SGML", "GlossTerm":"Standard Generalized Markup Language", "Acronym":"SGML", "Abbrev":"ISO 8879:1986", "GlossDef":{"para":"A meta-markup language, used to create markup languages such as DocBook.", "GlossSeeAlso":["GML","XML"]}, "GlossSee":"markup"}
Find GlossTerm	/glossary/GlossDiv/GlossList/GlossEntry/GlossTerm	"Standard Generalized Markup Language"
Find out GlossDef	/glossary/GlossDiv/GlossList/GlossEntry/GlossDef/	{"para":"A meta-markup language, used to create markup languages such as DocBook.", "GlossSeeAlso":["GML","XML"]}

Find out GlossSeeAlso	/glossary/GlossDiv/GlossList/GlossEntry/GlossDef/GlossSeeAlso/	["GML","XML"]
Find out 2nd element from GlossSeeAlso	/glossary/GlossDiv/GlossList/GlossEntry/GlossDef/GlossSeeAlso/[1]	"XML"

- Command with expressions

Expression operation	Meaning	Operation details
EQ	Equals	find target data on bases of equality
ED	Ends with	Find the date on the basis of value ends with parameter's value
CT	Contains	Find the date on the basis of value contains parameter's value
ST	Starts with	Find the date on the basis of value starts with parameter's value
LT	Less than	Find the date on the basis of value less than the parameter's value
GT	Greater than	Find the date on the basis of value greater than then parameter's value

Note: All expression operations are case-insensitive

- Expression syntax

  Find the list of string data which matches the following criteria
  /{ key operation 'input value'}

  Find the list of JSON objects which matches the following criteria
  /{@key operation 'input value'}

reducedjson@seantechsol.com

Command with expression examples

JSON data:

```
{
  "name": "John",
  "age": 30,
  "cars": [
    {
      "name": "Ford",
      "price": 30000,
      "models": [
        "Fiesta",
        "Focus",
        "Mustang"
      ]
    },
    {
      "name": "BMW",
      "price": 45000,
      "models": [
        "320",
        "X3",
        "X5"
      ]
    },
    {
      "name": "Fiat",
      "price": 35000,
      "models": [
        "500",
        "Panda"
      ]
    }
  ]
```

Command	Meaning	Result
/cars/[*]/	Select all cars	[{"name":"Ford", "price":30000, "models":["Fiesta","Focus","Mustang"]},{"name":"BMW", "price":45000, "models":["320","X3","X5"]},{"name":"Fiat", "price":35000, "models":["500","Panda"]}]
/cars/[0]/	Select first car	{"name":"Ford", "price":30000, "models":["Fiesta","Focus","Mustang"]}
/cars/[*]/name/	Select all cars names	["Ford","BMW","Fiat"]
/cars/[*]/models/	Select all cars models	[["Fiesta","Focus","Mustang"],["320","X3","X5"],["500","Panda"]]
/cars/[1]/models/[2]/	Select BMW X5	"X5"
/cars/[*]/{@name eq Fiat}/models	Select all models where name of car is Fiat	["500","Panda"]
/cars/[*]/{@name st F}/models/	Select all car's models where name of car starts with F	[["Fiesta","Focus","Mustang"],["500","Panda"]]
/cars/[*]/{@price gt 30000}/	Select all the cars where price of car is greater than 30000	[{"name":"BMW", "price":45000, "models":["320","X3","X5"]},{"name":"Fiat", "price":35000, "models":["500","Panda"]}]
/cars/[*]/{@price lt 37000}/	Select all the cars where price of car is less than 37000	[{"name":"Ford", "price":30000, "models":["Fiesta","Focus","Mustang"]},{"name":"Fiat", "price":35000, "models":["500","Panda"]}]
/cars/[*]/{price lt 37000}/	Select the all car price's where price is less than 37000	[30000,35000]
/cars/[*]/{name ct or}/	Select names of all cars where name contains 'or'	["Ford"]

/cars/[*]/{name ed W}/	Select names of all cars where name ends with ‘W’	["BMW"]
/cars/[*]/{@name ed W}/	Select all cars where name ends with ‘W’	[{"name":"BMW", "price":45000, "models":["320","X3","X5"]}]