

JSON-ND Experimental Tool Set

(Project Brief)

[JSON-ND](#) is a draft standard designed to complement [ECMA-404 The JSON Data Interchange Syntax](#), that includes data-type information within a JSON message, and provides a syntax for defining remote procedure calls without the need to for a complex schema definition.

The authors and future researchers of this proposed standard require a tool set in order compare the performance and developer experience using JSON-ND with existing methods (eg JSON Schema, WADL, Swagger).

The tools set will include a number of different server and client implementations of a simplistic REST API. These applications should be able to run on different platforms. The applications will be implemented in a mix of strongly typed languages and dynamically typed languages. Each of the servers will support a SQLite database backend.

The tool set must:

1. Expose the definition of the REST service in multiple definition languages (eg Swagger, JSON Schema)
2. Be capable of accepting and returning data in JSON and JSON-ND, but also other common formats such as XML, URL-encoded and Form-encoded.
3. Query and store a limited set data in a SQLite database
4. Create logs sufficient to generate statistics on
 1. message format and implementation language used
 2. successful and unsuccessful requests
 3. timing for upload/download, parsing, and processing times
 4. exceptions/faults encountered
5. Allow a user to interact with the REST API or to create test data for automated processing.

The development team is asked to deliver a part of this tool set.

The specific definition of the REST service to be implemented is up to the development team but must be chosen to specifically test the features that JSON-ND provides over standard JSON.

Developer teams are responsible for selecting suitable client and server languages to implement.

The team should create at least:

- 1 server (using either a strongly typed or dynamically typed language);
- 1 strongly-typed client implementation;
- and 1 dynamically typed client implementation.

At least one of the clients must be web based, and at least 1 other client must be a console or GUI application that can run on multiple platforms.

Client: Glen Kleidon (Galkam PTY LTD)

Email: glenk@galkam.com.au Phone: 0400821667 skype: glen.kleidon

