



Encoding FIX using JSON

Release Candidate 1

Technical Proposal

November 11, 2016

Document Revision: 0.1.0

Proposal Status: Public Comment

For Global Technical Committee Governance Internal Use Only

Submission Date	Dec 15, 2016	Control Number	
Submission Status	Public Comment	Ratified Date	
Primary Contact Person	Mike Gatny	Release Identifier	

DISCLAIMER

THE INFORMATION CONTAINED HEREIN AND THE FINANCIAL INFORMATION EXCHANGE PROTOCOL (COLLECTIVELY, THE "FIX PROTOCOL") ARE PROVIDED "AS IS" AND NO PERSON OR ENTITY ASSOCIATED WITH THE FIX PROTOCOL MAKES ANY REPRESENTATION OR WARRANTY, EXPRESS OR IMPLIED, AS TO THE FIX PROTOCOL (OR THE RESULTS TO BE OBTAINED BY THE USE THEREOF) OR ANY OTHER MATTER AND EACH SUCH PERSON AND ENTITY SPECIFICALLY DISCLAIMS ANY WARRANTY OF ORIGINALITY, ACCURACY, COMPLETENESS, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. SUCH PERSONS AND ENTITIES DO NOT WARRANT THAT THE FIX PROTOCOL WILL CONFORM TO ANY DESCRIPTION THEREOF OR BE FREE OF ERRORS. THE ENTIRE RISK OF ANY USE OF THE FIX PROTOCOL IS ASSUMED BY THE USER.

NO PERSON OR ENTITY ASSOCIATED WITH THE FIX PROTOCOL SHALL HAVE ANY LIABILITY FOR DAMAGES OF ANY KIND ARISING IN ANY MANNER OUT OF OR IN CONNECTION WITH ANY USER'S USE OF (OR ANY INABILITY TO USE) THE FIX PROTOCOL, WHETHER DIRECT, INDIRECT, INCIDENTAL, SPECIAL OR CONSEQUENTIAL (INCLUDING, WITHOUT LIMITATION, LOSS OF DATA, LOSS OF USE, CLAIMS OF THIRD PARTIES OR LOST PROFITS OR REVENUES OR OTHER ECONOMIC LOSS), WHETHER IN TORT (INCLUDING NEGLIGENCE AND STRICT LIABILITY), CONTRACT OR OTHERWISE, WHETHER OR NOT ANY SUCH PERSON OR ENTITY HAS BEEN ADVISED OF, OR OTHERWISE MIGHT HAVE ANTICIPATED THE POSSIBILITY OF, SUCH DAMAGES.

DRAFT OR NOT RATIFIED PROPOSALS (REFER TO PROPOSAL STATUS AND/OR SUBMISSION STATUS ON COVER PAGE) ARE PROVIDED "AS IS" TO INTERESTED PARTIES FOR DISCUSSION ONLY. PARTIES THAT CHOOSE TO IMPLEMENT THIS DRAFT PROPOSAL DO SO AT THEIR OWN RISK. IT IS A DRAFT DOCUMENT AND MAY BE UPDATED, REPLACED, OR MADE OBSOLETE BY OTHER DOCUMENTS AT ANY TIME. THE FPL GLOBAL TECHNICAL COMMITTEE WILL NOT ALLOW EARLY IMPLEMENTATION TO CONSTRAIN ITS ABILITY TO MAKE CHANGES TO THIS SPECIFICATION PRIOR TO FINAL RELEASE. IT IS INAPPROPRIATE TO USE FPL WORKING DRAFTS AS REFERENCE MATERIAL OR TO CITE THEM AS OTHER THAN "WORKS IN PROGRESS". THE FPL GLOBAL TECHNICAL COMMITTEE WILL ISSUE, UPON COMPLETION OF REVIEW AND RATIFICATION, AN OFFICIAL STATUS ("APPROVED") OF/FOR THE PROPOSAL AND A RELEASE NUMBER.

No proprietary or ownership interest of any kind is granted with respect to the FIX Protocol (or any rights therein).

Copyright 2013-2016 FIX Protocol Ltd., all rights reserved.



Encoding FIX using JSON by [FIX Protocol Ltd.](#) is licensed under a [Creative Commons Attribution-NonDerivatives 4.0 International License](#).

Based on a work at <https://github.com/FIXTradingCommunity/fix-json-encoding-spec>

Document History

Revision	Date	Author	Revision Comments
0.1.0	2016-11-11	Mike Gatny	Initial draft.

Table of Contents

Document History	3
1 Introduction	5
1.1 Authors.....	5
2 Business Requirements	6
3 Issues and Discussion Points	6
3.1 Semantic.....	6
3.2 Versioning.....	6
3.3 No Templates	6
3.4 Field Name and Value Encodings	6
3.5 Header, Body, and Trailer	7
3.6 Repeating Groups.....	7
3.7 Dates and Times	7
3.8 Decimal Representation and Byte Order	7
4 References.....	8
5 Relevant and Related Standards	8
6 Appendix A - Usage Examples	8
7 Appendix B - Compliance Strategy.....	8

1 Introduction

As financial applications have continued to move from the traditional desktop to the web, a need has emerged for an encoding of FIX optimized for operations in the web browser without the need for other software distribution to clients.

JavaScript Object Notation (JSON) is the *lingua franca* for communications between web browsers and the services that power them. This proposal describes the use of JSON to encode FIX messages.

JSON Encoding of FIX supports the development of web applications that require FIX semantics, and is an alternative to existing FIX encodings, including *tag=value*, FIXML, Simple Binary Encoding (SBE), and Google Protocol Buffers (GPB). It is not optimized for low latency, but rather for human readability and interoperability between web services and web languages.

1.1 Authors

Name	Affiliation	Contact	Role
Mike Gatny	Connamara Systems	mgatny@connamara.com	JSON Encoding Co-chair
Krishna Tharanoju	Credit Suisse Group	krishna.tharanoju.2@credit-suisse.com	JSON Encoding Co-chair

2 Business Requirements

It is proposed that the attached technical specification draft be admitted to the FPL standardization process and eventually be made available to FIX implementers and users. In the interest of standardization, this proposal specifies a normative encoding of FIX to JavaScript Object Notation (JSON).

3 Issues and Discussion Points

3.1 Semantic

RC1 specifies that all of the semantic data types of FIX protocol should be mapped to JSON *string* values in order to maximize the ability of applications to simply display the data.

3.2 Versioning

RC1 does not specify explicit versioning. For web applications, versioning is typically realized as distinct URLs instead of at the JSON level.

3.3 No Templates

The use of a template/schema for validation was considered. Several popular formats (e.g. JSON Schema) exists for this purpose, however none is standard. In RC1, it was decided to optimize for simplicity and not require the use of template/schema. RC1 does recommend generating the template/schema format of choice (e.g. JSON Schema) from the FIX Repository or FIX Orchestra, if desired.

3.4 Field Name and Value Encodings

RC1 specifies using field *Name* instead of *Tag* number to optimize for human readability. *Tag* numbers may be used for user-defined fields.

Using *SymbolicName* for field value was considered. However, RC1 specifies the field *Value* should be used instead of its *SymbolicName* to optimize for interoperability with and convertibility to/from the other standard encodings of FIX (e.g. FIXML, SBE, GPB).

3.5 Header, Body, and Trailer

A guiding principal of the development of RC1 was not to throw away any encoding information that might be useful when converting JSON Encoding to/from other FIX Encodings. As such, RC1 specifies that the Standard Header, Standard Trailer, and Body are explicitly framed as such.

Inclusion of the CheckSum field was considered, but discarded since it is unlikely to be useful at best, and likely to be incorrect at worst (e.g. if copied over from another FIX encoding).

3.6 Repeating Groups

The count of entries is implicit in JSON Encoding. There is no explicit *NumInGroup* field in the JSON encoding.

The name of a repeating group is the name of the associated *NumInGroup* field name as it appears in the FIX Repository. Using the Component name was considered, but rejected as less simple, and more likely to make the use of FIX Repository or Orchestra required as opposed to optional.

3.7 Dates and Times

JSON has no explicit provision for encoding dates or times. RC1 recommends using the ISO 8601 conversion facilities of the target platform.

RC1 specifies that applications should adhere to the recommendations of [FIX.5.0 SP2 EP206: Clock Synchronization Data Types Enhancements](#) when sub-millisecond precision is required.

3.8 Decimal Representation and Byte Order

Since all values, including numbers, are serialized as their string equivalent, there is no issue with either decimal representation or byte order.

4 References

Reference	Version	Relevance	Normative
Encoding FIX Using JSON - User Guide			
GitHub project FIXTradingCommunity/fix-json-encoding-spec		Specifications, working drafts, issue tracking.	
GitHub project FIXTradingCommunity/fix-json-encoding-examples		Examples in a variety of programming languages.	
<u>Standard ECMA-404 The JSON Data Interchange Format</u>		Standard Specification of JSON	

5 Relevant and Related Standards

Related Standard	Version	Reference Location	Relationship	Normative
FIX Simple Open Framing Header			Recommended if an additional framing protocol is needed	
FIX.5.0 SP2 EP206 Clock Synchronization Data Types Enhancements			Recommended when sub-millisecond date/time precision is needed	

6 Appendix A - Usage Examples

Examples are provided in the specification document, and on the GitHub project page (see References).

7 Appendix B - Compliance Strategy

None.