

Development of a Common Format of Questionnaire Tests for a Web-based Platform of Population and Experimental Psychological Research

Evgeny Nikulchev^{1,2}, Dmitry Ilin^{1,2,}, Pavel Kolyasnikov¹, Victoria Ismatullina³ and Ilya Zakharov³*

¹Russian Academy of Education, 119121 Moscow, Russia

²Moscow Technological University MIREA, 119454 Moscow, Russia

³Psychological Institute of Russian Academy of Education, 125009 Moscow, Russia

Abstract. A web platform for psychological research needs a single format of questionnaire tests to ensure interaction between its components. The study proposes a general test structure, variant questions, variations in response types, and an embedded domain-specific language for computations. The use of JSON is proposed and justified to store the hierarchical structure of the questionnaire test, and JSON Schema is defined as a technology suitable for the formation of the standard. From the considered validation instruments for compliance with the standard described in JSON Schema, ajv was defined as the most applicable to the task. To build the documentation, Docu is relevant, but this tool needs to be modified to meet the requirements of the task.

1 Introduction

Today, online tools are increasingly used for research [1]. In addition to universal solutions, which are equally applied in different disciplines, there are also used developments that are specific for a given area of research. For research in the field of psychology, survey facilities [2] are used, which are a substitute for the traditional pen and paper approach. Such a tool reduces the work of the researcher [1, 3], allowing to work with large samples [4]. There are also projects [5, 6] that transfer the tools for conducting cognitive research online.

The RAE Psychological Institute is developing a web platform [2] for conducting psychological research both online and offline in order to take into account a wide range of requirements for setting up experiments. The platform consists of many components, the integration of which requires the creation of unified standards for data exchange. For a simple interaction between components, the REST API is sufficient [7], whereas to establish reproducible experiments a more formalized approach is needed. Previous analogues have been analyzed [2]. The results showed that there is no single standard for the development of psychological questionnaire tests, and therefore the developed of psychological platform requires selection of appropriate software instruments and identification of approaches that would allow to solve the task.

2 Analysis of requirements and the formation of the structure of the psychological test

2.1 The overall structure of the questionnaire tests

Based on the needs of researchers, each questionnaire test can contain:

1. **Questions** as a basis for data collection.
2. **Scales** for computation based on answers to questions.
3. **Feedback**, which the subject will receive as a result of passing the test.
4. **Reports**, namely the description of how to provide summary information to the researcher.

Variants of representation of questions can differ, therefore it is expedient to group questions into blocks within which there will be specific for it settings. One of the requirements is to interrupt the passage of a block of questions on the fulfillment of a certain condition (for example, exceeding the number of incorrect answers). In addition to the questions, two more kinds of elements should be included in these blocks:

1. **Text areas** that allow writing instructions to the test.
2. **Page breaks** that may be needed if the test involves exceptions to the rule that determines the number of questions/text areas displayed on a single page.

* Corresponding author: i@dmityilin.com

2.2 The structure of questions in the questionnaire tests

By analyzing the requirements for the content of psychological questionnaire tests, existing systems and examples of tests, key types of questions for the functioning of the platform were obtained. Each type has a characteristic set of properties. Types of questions can be divided into simple and composite. Simple questions will be answered by one single key regardless of whether it is a scalar or a vector value. Compound questions imply that there may be more than one key in their composition, for each of which a response can be saved. They can also mean different types of data stored in the platform. For clarity, the types of questions are presented in Table 1, indicating the main characteristics.

Table 1. Characteristics of types of questions.

Type	Composite	Valid values	Comment on the answer	Restrictions
Single choice	No	Preset values	Available	No
Multiple choice	No	Preset values	Available	No
Input field	No	Any input values	Not available	Input mask, answer length, number bounds
Complete the gaps	Yes	Any input values	Not available	Input mask, answer length, number bounds
Slider	No	Values in the specified range	Not available	Bounds of number
Date input	No	Values in the specified range	Not available	Date boundaries
Sequence	No	Preset values	Not available	No
Map	No	Preset values	Not available	No
File upload	No	File	Not available	File size, file type
Single choice matrix	Yes	Preset values	Available	No
Multiple choice matrix	Yes	Preset values	Available	No
Input field matrix	No	Any input values	No	Input mask, answer length, number bounds

In turn, the values of answers can also differ in types. Table 2 provides a list of questions, their corresponding

data structures for information on responses, and lists of possible value types.

Table 2. Characteristics of the stored response data.

Type	Data on the response	Possible value data types
Single choice	Value, response number and comment	Text, Number, Boolean, Index Number
Multiple choice	Vector of values, answer and comment numbers	Text, Number, Boolean, Index Number
Input field	Value	Text, Number
Complete the gaps	Value	Text, Number
Slider	Value	Number
Date input	Value	Date
Sequence	Vector of values	Text, Number
Map	Vector of values	Text, Number
File upload	Link to file	Link to the uploaded file
Single choice matrix	Value, response number and comment	Text, Number, Boolean, Index Number
Multiple choice matrix	Vector of values, answer and comment numbers	Text, Number, Boolean, Index Number
Input field matrix	Vector of values, row numbers and column numbers	Text, Number

In addition to the above, the answer to each of the simple questions can contain a note of compliance with the expected value (for example, the correct answer). For compound questions, a similar mark can be specified for each of the sub-questions.

2.3 The domain-specific language for calculating the values of the scales

For carrying out the calculations, it is expedient to develop a domain-specific programming language. It must meet the following criteria:

1. Be applicable for calculations both on the server and in the browser.
2. Use only pure calculations to ensure reproducibility of results.
3. Be easily editable both manually and with software tools.

The language corresponding to them can be obtained using declarative and functional programming paradigms. The main aspect of the functional paradigm that is required in the task is pure functions. For execution of the code, the interpreter is used as it can be used regardless of the environment, and the speed of calculations for the task is not critical, because of their

relatively small volume within the limits of one psychological test.

The following groups are the main built-in functions of the language:

- Logical functions (and, or, not).
- Comparison functions (equal, greater than, etc.).
- Higher-order functions (map, reduce, etc.).
- Mathematical functions (add, subtract, etc.).
- Functions for working on vectors (sum, size, etc.).
- Functions for obtaining data (getAnswers, getScale, etc.).

At the current stage, the provision of advanced functionality in a domain-specific language does not seem appropriate. Thus, declaring user-defined functions and declaring variables explicitly is not necessary.

3 Data formats analysis

Based on the analysis of functional requirements and limitations related to psychological tests, questionnaires, it can be said with certainty that their description is possible in the form of a document with hierarchical connections. To describe them, the most suitable are the semi-structured data formats. The most common of these are XML and JSON [8]. For this task, BSON can be considered as a variation of JSON for efficient storage in document-oriented non-relational databases, so it will not be considered as a separate format.

In favor of choosing XML, the following series of arguments exist:

- XML is more common, there are more tools to work with it.

In favor of choosing JSON, there are the following arguments:

- Serializing and deserializing JSON is faster [9, 10] than XML
- Since the main target execution environment is the browser, JSON is preferable, because its usage will imply less overhead costs for resources.
- The selected tools (Node.js, MongoDB) also work with JSON directly [11], whereas for XML, conversion is required.

The effectiveness of the software components is important for the web platform, so JSON is preferable.

4 Selection of tools

The choice of tools for forming a standard for describing psychological questionnaire tests is based on the following criteria:

1. The format description tools should allow the validation of the questionnaire instance against the format.
2. There should be an option to automatically build a document describing the format in a way that is understandable to a person.

For the formation of standardized JSON documents, de facto, there is only one solution, which can be called well-established: JSON Schema [12, 13]. This standard

contains several revisions. At the moment, the latest version is Draft-07, however, at the time of the analysis, version Draft-06 was actual. The JSON Schema itself is not a tool for solving the tasks, but it is a standard for describing the structure of JSON documents, to which a whole list of tools that allow automating the construction of documentation has been developed. Also a wide range of libraries was developed for various programming languages that can check JSON-objects for compliance with a specified scheme.

The tools for checking JSON objects for JSON Schema compliance must be integrable into the components of the web platform, and therefore, it is necessary to consider development in the JavaScript language. In addition, it is necessary to limit the list of considered tools to those that support Draft-04 and Draft-06, and work in both the browser and Node.js. A brief description of the selected tools is presented in Table 3.

Table 3. Object validation tools for JSON Schema.

Name	ajv	djv
Supported version of JSON Schema	Draft-04, Draft-06	Draft-04, Draft-06
Number of tests per second based on the json-schema-benchmark	47068	48052
Number of failed tests	1	1
Number of downloads per month	~3400000 [14]	~6500 [15]

Both tools are leaders in the number of tests per second according to the version of the json-schema-benchmark [13, 16]. In addition, they have the least number of failed tests. With equal characteristics, they have a significant difference in the size of the community, which is confirmed by the multiple differences in the number of projects with the integration of these tools. Thus, the choice in favor of ajv is adequate, since the reliability of the tool is tested by a large community.

All the documentation tools [13] are free to use. A more detailed description is given in Table 4.

Each of the tools supports Draft-04, which is a sufficient condition for the application. Nevertheless, considering more in detail each of the tools, one can identify additional characteristics that affect the choice. Because the Web platform is based on technologies that are closely related to the JavaScript language, the use of Prmd may require additional time to master. The Matic and Docson tools provide documentation in the form of nested entities that, in general, correspond to the essence of JSON, but can cause difficulties when reading the documentation. They do not implement the construction of a list of all entities included in the schema, which can be good for small objects, but it will negatively affect the readability of large documentation. Each of the tools is designed to build documentation, primarily for the API, which is redundant for the task. This may require one to

rework some of the theme files. From this, it should be concluded that each of the tools does not meet the desired characteristics, but the most adequate to the task is the Doca tool.

which is necessary to ensure backward compatibility of the tools.

This research is supported by the RFBR grant no. 17-29-02198.

Table 4. Documents construction tools.

Name	Matic	Docson	Doca	Prmd
License	MIT	Apache 2.0	BSD	MIT
Implementation language	JavaScript	JavaScript	JavaScript	Ruby
Supported version of JSON Schema	Draft-03, Draft-04	Draft-04	Draft-04, Draft-07 (в планах)	Draft-04
Using themes	Yes	No	Yes	Yes
Building content	No	No	Yes	Yes
Number of downloads per month	~500 [17]	~200 [18]	~1400 [19]	-

During operation, it was revealed that the Doca tool has a drawback: the lack of processing of cyclic references inside the JSON schema. In this connection, it was required to move from using a method that combines a schema divided into components into a single JSON object to a method in which a single file with internal links is formed. This led to a loss of the ability to view the cyclically related objects in depth, but allowed to keep the possibility of using Doca as a whole.

5 Conclusions

The study carried out an analysis of existing services for conducting psychological research in the form of questionnaire tests. Since there are no solutions available on the market that would provide a single standard, it was therefore suggested to develop one. Authors formed the structure of the questionnaire test, consisting of four sections (questions, scales, feedback, and reports). In the paper, the first two were considered in more detail, since they are the basis for research and data collection. Analysis of data formats showed that JSON is relevant for the construction, storage, and playback of questionnaires. The review of existing solutions for the formation of standards of semi-structured data in the JSON format showed that the only applicable is JSON Schema. Among the tools for validating questionnaire tests and for constructing documentation, ajv and Doca were selected, respectively. Their use is justified, since they support the latest versions of the JSON Schema standard, and the Doca user interface is the most suitable for the task. Subsequent studies may be aimed at examining two sections of questionnaire tests that are outside the scope of the article (feedback and reports). An important issue for further consideration is migration to a newer version of the standard of questionnaire tests,

References

1. S.D. Gosling, W. Mason, Annual review of psychology, **66**, 877 (2015)
2. I. Zakharov, E. Nikulchev, D. Ilin, V. Ismatullin, A. Fenin, ITM Web of Conferences, **10**, 04006 (2017)
3. G. Andersson, N. Titov, World Psychiatry, **13**, 4 (2014)
4. C. Bartneck, A. Duenser, E. Moltchanova, K. Zawieska, PloS one, **10**, e0121595 (2015)
5. [Online]. Available: <https://www.labvanced.com/>. [Accessed 20 02 2018].
6. J.R. De Leeuw, Behavior research methods, **47**, 1 (2015)
7. R. Sellami, S. Bhiri and B. Defude, ODBAPI: a unified REST API for relational and NoSQL data stores, In Big Data Congress (2014)
8. A.B.M. Moniruzzaman, S.A. Hossain, International Journal of Database Theory and Application, **6**, 1 (2013)
9. N. Nurseitov, M. Paulson, R. Reynolds, C. Izurieta, Comparison of JSON and XML data interchange formats: a case study, *In Caine*, pp. 157-162 (2009)
10. D. Peng, L. Cao, W. Xu, Journal of Computational Information Systems, **7**, 5883 (2011)
11. C. Györödi, R. Györödi, G. Pecherle, A. Olah, A comparative study: MongoDB vs. MySQL, *In 2015 13th International Conference on Engineering of Modern Electric Systems* (2015)
12. F. Pezoa, J. L. Reutter, F. Suarez, M. Ugarte, D. Vrgoč, Foundations of JSON schema, *In Proceedings of the 25th International Conference on World Wide Web* (2016)
13. [Online]. Available: <http://json-schema.org/implementations.html>
14. [Online]. Available: <https://www.npmjs.com/package/ajv>
15. [Online]. Available: <https://www.npmjs.com/package/djv>
16. [Online]. Available: <https://github.com/ebdrup/json-schema-benchmark>
17. [Online]. Available: <https://www.npmjs.com/package/matic>
18. [Online]. Available: <https://www.npmjs.com/package/docson>.
19. [Online]. Available: <https://www.npmjs.com/package/doca>.