*ICT for Health Science Research*
*A. Shabo (Shvo) et al. (Eds.)*


*doi:10.3233/978-1-61499-959-1-11*

# Automated Verification of Structured Questionnaires Using HL7® FHIR®

Anna M. LACKERBAUER [a,1], Oliver KRAUSS [a] and Emmanuel HELM [a]
[a] *University of Applied Sciences Upper Austria, Advanced Information Systems and Technology (AIST), Hagenberg, Austria*

**Abstract.** Informed consent of patients to research studies is a cornerstone to modern healthcare, which has lead to considerable administrative effort. The purpose of this paper is to show how forms and questionnaires and their respective answers can be captured in a standardized, structured way, in order to enable automated verification. The use of the HL7 FHIR resources Questionnaire and QuestionnaireResponse is discussed with respect to the different implementation options of Extensions, POST Interceptors, FHIR Operations, and CDS Hooks. These four approaches are described and it is determined whether they produce standard-compliant results and how they can be integrated with other solutions. Since all approaches yield advantages and disadvantages, the choice amongst any option must be based on the actual use case.

**Keywords.** HL7 FHIR, Questionnaire, CDS Hooks, FHIR Operation

## 1. Introduction

In healthcare, patient information can be captured using forms and questionnaires that are completed by either healthcare professionals or by the patients themselves. These forms and questionnaires have a need for structured data entry, with questions and answers entered as free text or multiple/single choice. In some cases (e.g. where informed consent is obtained remotely using an eConsent system), it is important to verify whether participants understand what they are providing consent to; as such, it is necessary to confirm their knowledge using, for example, a multiple-choice quiz [1].

Fast Healthcare Interoperability Resources (FHIR®) [2] is a standard for medical healthcare data exchange developed by Health Level Seven (HL7®). FHIR is based on the concept of Representational State Transfer (RESTful) webservices, which, in turn, is based on the HyperText Transfer Protocol (HTTP).

The use of FHIR Questionnaires is an evolving topic of high interest. Geßner et al. showed that FHIR Questionnaires are in high demand in medical research infrastructure [3]. The use cases for FHIR Questionnaires are widespread, including:

- performing studies and surveys such as Case Report Forms (CRF) [4, 5]

[1]Corresponding Author: Anna M. Lackerbauer; University of Applied Sciences Upper Austria, Softwarepark 13, Hagenberg 4232, Austria; E-mail: anna.lackerbauer@fh-hagenberg.at

- templating for information display and generation of a user interface (UI) [6]
- generation of dynamic forms and forums [7]
- implementing scoring systems, e.g. for triage of patients in the emergency department [8]

Aside from work published on eConsent [1], the literature does not show any implementations of performing a quiz with FHIR Questionnaires. This paper describes the options for quiz verification that are supported by FHIR. FHIR resources required for data capture are discussed in the context of technical solutions, with advantages and disadvantages considered. Different options for identification and verification of the correct answer(s) given to the questions of a FHIR Questionnaire are also described.

The underlying FHIR version is 3.5.0 and was published on August 21, 2018 as part of the *Release 4* (R4) sequence (`https://www.hl7.org/fhir/2018Sep`).

## 2. Results

Before it can be verified if a user passed the quiz, the system must identify which answers are correct for each question. The *identification* of the correct answers occurs when the quiz is initially defined. There are two different FHIR structures that can be used to identify the correct answers: Questionnaire and QuestionnaireResponse (QR) [2].

The *verification* of a potentially correct set of answers (a specific QR) can either be done by the client or by the server, using one of the different approaches described below. In this context, verification succeeds if (a) all the required answers were provided and (b) the correct answers have been selected.

### *2.1. Correct answers in questionnaire extensions*

One approach to identify the correct answers is to add them to the same Questionnaire that defines the questions and the answer options of the quiz. The identification of the correct answer can be done for each item of the Questionnaire by a FHIR Extension [2]. An example of such an Extension is depicted in Listing 1.

The verification can occur directly on the client's side by comparing the given answers to the correct ones defined in the Extension. Depending on the policy, users can be informed about an incorrect input immediately for each field or at the end when they are trying to submit the entire quiz form. The client does not POST the QR to the server unless it contains all the correct values. For multiple-choice scenarios, it is essential that the comparison is independent of the order of the given answers. For example, if the correct answers are defined as the two options [A, B], the verification should also succeed if the given answers are in the order [B, A].

The danger of this approach is that a tech-savvy user can potentially auto-fill the correct answers on the client's side or circumvent the verification entirely. Furthermore, no information is stored about how often the user attempted to submit the quiz with incorrect answers. In the context of eConsent, a user could simply bypass learning what they are consenting to by trying answer permutations until the verification succeeds. While a client-side verification has these disadvantages, it can still be combined with other, server-side verification mechanisms, to notify users of missing input or minor typing errors (e.g. age 200 instead of 20).

Listing 1: FHIR quiz Questionnaire using an Extension to identify the correct answers [1,2].

```
{
  "resourceType":"Questionnaire",
  "name":"Study Quiz",
  "item":[
    { "linkId":"question.1",
      "text":"I can quit the study at any time.",
      "type":"boolean",
      "extension":[{
          "url":"http://example.com/StructureDefinition/correct-answer",
          "valueBoolean": true }]
    },
    { "linkId":"question.2",
      "text":"The duration of the study is:",
      "type":"choice",
      "extension":[{
          "url":"http://example.com/StructureDefinition/correct-answer",
          "valueString":"4 weeks (28 days)" }],
      "answerOption":[
        { "valueString":"4 weeks (28 days)" },
        { "valueString":"8 weeks (56 days)" },
        ...
      ]
    }
  ]
}
```

### 2.2. *Correct answers in questionnaire response instance on server*

To conceal the correct answers from the client, one can store a predefined reference QR resource on the server that contains the correct answers and can be used for comparisons with the QR instances to validate. The quiz Questionnaire does not contain the introduced Extension but solely the information to render the required form. For verification purposes, the client must send requests to the server. The verification can happen in two different ways: by intercepting the HTTP POST request or by executing a FHIR Operation.

#### 2.2.1. *POST interceptor*

The client attempts to submit a (potentially incorrect) set of quiz answers by performing a POST request with the unverified QR to the FHIR server. The server intercepts this POST to compare the submitted answers with the stored reference QR. If the verification succeeds, the server stores the QR untouched. Otherwise, the server will respond with a comprehensive HTTP error code. Another option is that the server stores the QR regardless of the verification result but attach an Extension that labels whether this resource passed verification.

The *create* operation of FHIR (POST in HTTP) states that a server should accept the resource as submitted, with the only notable exceptions being transactional creates (which may rollback the create later) and the allowed modification of metadata values [2]. As the POST interception would either deny the *create* or add an Extension (essentially modifying it) this solution does not comply with FHIR. The primary advantage of the POST Interceptor is the simplicity from a client-implementer's perspective.

#### *2.2.2. FHIR operation*

The client can trigger the verification and creation of a QR by calling a FHIR Operation on the server [2]. An Operation, e.g. `$verifyAndCreate`, can be executed. The QR resource is automatically created by the Operation if the verification succeeds. Otherwise, the Operation results in a comprehensive error code and does not create any resource.

The disadvantage of using FHIR Operations is mainly the additional overhead it creates from a developer's perspective. The client has to implement an Operation call instead of a simpler POST. A server developer needs to provide an endpoint that supports the Operation, as well as creating the related OperationDefinition resource and exposing it in the FHIR server's Conformance Statement [2].

### *2.3. CDS hooks*

CDS Hooks[2] is a RESTful Application Programming Interface (API) that enables calls to a Clinical Decision Support (CDS) system, essentially creating the ability to *hook* it into other software. While CDS Hooks integrates with HL7 FHIR and is driven by the FHIR community, it is not part of the FHIR standard. CDS Hooks can be used to verify the content of a QR using the Clinical Query Language (CQL). Similar to the other server-side operations, the result is that no information about the correct answers is transmitted to the client.

The major advantage of CDS Hooks is that, in addition to the verification, it can enable workflows based on the answers (e.g. only switching to the next question if the previous was correct or even switching to a specific question depending on the response). The major disadvantages of CDS Hooks are the technical overhead requiring a CDS system in the background and not having any control over the follow-up. While CDS Hooks can verify the QR, it is not intended to create the QR resource on a FHIR server. The creation of the QR resource has to happen on the client's side, enabling attacks that circumvent the verification entirely.

## 3. Discussion

The four approaches described above are summarized in Table 1, which also shows their possible combinations. The POST Interceptor is the only approach that is not standard compliant. The client-side verification and the CDS Hooks approach both open up possible attack angles since they require the QR resource to be created by the client.

**Table 1.** Comparison of the four identified verification approaches.

| Approach | Verification By | Standard Compliance | Integration With | Security |
|---|---|---|---|---|
| Extension (Ex) | Client | ✓ | In, Op, CDS | - |
| POST Interceptor (In) | Server | - | Ex, CDS | ✓ |
| FHIR Operation (Op) | Server | ✓ | Ex, CDS | ✓ |
| CDS Hooks (CDS) | Server | ○[3] | Ex | - |

[2] https://cds-hooks.org

[3] CDS Hooks is not part of the FHIR standard but unlike POST Interceptor does not break with it.

Of note, constraints on the syntax or the data format of the required input, such as allowed numerical ranges or regular expressions, are captured as common extensions in the FHIR standard [2], and therefore are not discussed here.

FHIR is currently in Version *Standard for Trial Use 3*, with the official R4 version slated to be published in January 2019 [2]. The resources Questionnaire and QuestionnaireResponse are likely to change in the future. CDS Hooks is currently on a Draft of Release 1.0, and as such, the API may change in the future as well.

## 4. Conclusion and outlook

This paper identified four approaches of verifying correctness of a specific QR instance. The approach must be carefully chosen based on the specific use case. For some use cases, it may not be suitable to store the correct answers in the Questionnaire, since a tech-savvy user could cheat on the quiz. In terms of the standard compliance, the POST Interceptor changes the behavior of the POST request, and thus does not comply with the FHIR standard. FHIR Operations are suitable for automated verification and creation of a QR but are more elaborate in terms of implementation. CDS Hooks has similar capabilities to FHIR Operations but requires a CDS system in the background. CDS Hooks is also not able to publish the resources by default, and as such, this approach has similar disadvantages to ones associated with client-side verification. In the use case of eConsent, the FHIR Operation approach would suit best, since workflow integration is not a requirement and it offers a secure and standard-compliant option.

Further research on the applicability of FHIR resources in the context of informed consent will be considered. To lower the initial barriers for defining and establishing FHIR Operations, a comprehensive documentation and an easy-to-use development framework should be pursued in the future.

## References

[1] A. M. Lackerbauer: Digitalising Consent in Healthcare: Development of an eConsent Backend, *Master thesis, University of Applied Sciences Upper Austria* (2018).

[2] HL7, *Fhir R4 Ballot #2*, `https://www.hl7.org/fhir/2018Sep`, 2018.

[3] S. Geßner, P. Neuhaus, J. Varghese, P. Bruland, A. Meidt, I. Soto-Rey, M. Storck, J. Doods, and M. Dugas: The Portal of Medical Data Models: Where Have We Been and Where Are We Going?, *International Medical Informatics Association (IMIA) and IOS Press*, 2017.

[4] H. Leroux, A. Metke-Jimenez, and M. J. Lawley: Towards achieving semantic interoperability of clinical study data with FHIR, *Journal of Biomedical Semantics* **8:1** (2017).

[5] P. B. Pfiffner, I. Pinyol, M. D. Natter, and K. D. Mandl: C3-PRO: Connecting ResearchKit to the Health System Using i2b2 and FHIR, *Plos One* **11:3** (2016).

[6] A. M. Lackerbauer, A. C. Lin, O. Krauss, J. Hearn, and E. Helm: A Model for Implementing an Interoperable Electronic Consent Form for Medical Treatment Using HL7 FHIR, *European Journal for Biomedical Informatics* **14:3** (2018).

[7] B. Lohmann: Introducing dynamic forms in a client-server based hospital information system in Uganda, *Master thesis, Heilbronn University* (2016).

[8] R. Guldhammer Blendal and L. Pape-Haugaard: An International Minimal Patient Care Report Exemplified in FHIR to Facilitate Standardisation and Interoperability of Emergency Medical Services Data, *in-Proceedings from The 16th Scandinavian Conference on Health Informatics; Linkping University Electronic Press* (2018), 85-90.