HL7® FHIR® Foundational Implementer Exam Study Guide

Overview

The purpose of the HL7 FHIR Foundational Implementer Exam is to identify candidates that can demonstrate a basic understanding of the FHIR specifications and FHIR, working with resources, RESTFul Operations, conformance and terminology, and security basics.

This entry-level certification is a first step for candidates with an IT background pursuing a career in FHIR-based interoperability.

Recommendations	Description
6-months experience using FHIR	This is HL7's entry-level credential and although no formal barrier for newcomers exists without 6 months of experience, it is very unlikely that a candidate will be able to pass the exam.
Understand basic concepts of XML/JSON	Lack of understanding the basic concepts of XML/JSON will make implementation difficult. Understanding the basic concepts is highly recommended before sitting for the exam.
Understand basic concepts of RESTful APIs	RESTful APIs are the most common exchange method for FHIR implementation, and it is required by most regulations. Basic understanding of the concepts is not mandatory but recommended before sitting for the exam.

Competencies

The Foundational Implementer certificate demonstrates a basic understanding of HL7 FHIR R4 specification and its use. Those that pass the test will have demonstrated their knowledge of the following concepts:

Competency	% of test
Understanding Implementation Guides	**4-8%**
Select profiles based on use case	3-5%
Determine which implementation guide to use for a use case	1-3%
FHIR API Behavior	**19%-33%**
Determine exchange paradigms for a specific use case	4-6%
Select RESTful API methods for specific interactions	4-6%
Select operations for a use case	2-4%
Interpret a capability statement	2-4%
Select Bundle features for a use case	3-5%
Follow safety checklist	2-4%
Follow security checklist	2-4%
Resource Model and Structure	**25%-33%**
Select resources based on a use case	7-9%
Determine suitability of an element based on use case	9-11%
Determine search parameter availability for resources	2-4%
Determine existing extensions for a use case	2-4%
Determine terminology for a use case	3-5%
Implementation	**19%-29%**
Use existing extensions for a use case	2-4%
Use selected element based on a use case	9-11%
Use search parameters for resources	2-4%
Use operations for a use case	2-4%
Use terminology for a use case	4-6%
Troubleshooting and Validation	**13%-19%**
Troubleshoot validation errors	4-6%
Troubleshoot the rest API server errors	5-7%
Apply profile rules to validate resource instance	4-6%

Preparation

HL7 FHIR implementation experience will prove helpful to the exam taker. Formal training is also strongly encouraged.

This test applies to the current FHIR Version (is intended to be version independent).

Understanding Implementation Guides

Select profiles based on use case:
- Locate the list of available resources / profiles in the standard or implementation guide.
- Interpret the resource scope and usage, description, boundaries, and relationships sections and compare it to the use case.
- Infer any other related resource/profile needed to complete information requirements for the use case.

Determine which implementation guide to use for a use case:
- Locate the registries of Implementation Guides (global, local, those recommended in the domain etc.).
- Search by keywords to find ones that may be appropriate e.g., "laboratory," "genomics," "UK".
- Examine each implementation guide, read the scope, use cases, and interactions to determine appropriateness.

API Behavior

Determine exchange paradigms for a specific use case:
- Recall the available exchange paradigms (include new paradigms like bulk-transfer and subscription).
- Compare the key features of the paradigm and the use case.
- Estimate suitability of each paradigm and make a selection.

Select RESTful API methods for specific interactions:
- Locate the category of API method for an interaction (creating, updating, deleting, read, searching, executing an operation).
- Interpret the use of conditional updates.
- Compare the use of direct get vs search.

Select operations for a use case:
- Locate the resource/system operation list.
- Determine candidate operations in the list.
- Compare the operation parameters with provided requirements to identify gaps.

Interpret a capability statement:
- Distinguish the sections of the capability statement.
- Locate which resources are supported in REST mode.
- Locate which methods are supported for each resource in REST mode.

Select Bundle features for a use case:
- Determine the bundle type within the paradigm.
- Determine key components and parts of the bundle.

Follow safety checklist:
- Locate the checklist.
- Interpret the checklist.
- Apply items from the checklist to specific problems.
- Give an undesired consequence and ask what could have been done to prevent it.

Follow security checklist:
- Locate the authentication method used either by the client or server depending on the context.
- Compare the AuditEvent resource entries to Basic Audit Log Patterns Implementation Guide to determine conformance with recommendations.
- Consider methods for secure communications with the FHIR API.

Resource Model and Structure

Select resources based on use case:
- Select the right resource given a scenario.
- Select the right "workflow" resource for a scenario.
- Identify when Provenance is used.

Determine suitability of each element based on use case:
- Define the component data elements, constraints, and relationships for use case.
- Determine element cardinality and flags.
- Determine the terminology for a coded element.
- Determine correct reference mechanism to link various resource.

Determine search capability for resources given a use case:
- Interpret which search parameters are available for a specific resource.
- Estimate how to combine search parameters to retrieve that resource.
- Determine which search parameters are applicable for a particular scenario.
- Find out if there is an operation that fits your use case.

Select suitable extension for a use case:
- Decide if data requirement is not met by the resource(s) and existing properties.
- Search FHIR standard extensions, and those of relevant IGs and other extensions from registries, to find a suitable extension.
- Compare the data type of the extension with the existing data.

Interpret terminology requirements for an element:
- Distinguish the correct data type to represent an element of a use case.
- Determine the binding strength.
- Determine the value set.

Implementation

Use selected extensions for a use case:

- Format the extension instance in FHIR data.
- Put the extension in the correct place - extend the resource or the element (or data type) within it.
- Identify the correct placement sequence of an extension in a resource for XML or JSON
- Identify the correct URL for the extension.
- Use the correct data type for the value.

Use selected element based on a use case:

- Use simple and complex data types.
- Use the name of the element correctly.
- Confirm elements must be in the correct order, and so must elements in data types, for XML.
- Put the data in the right place (XML).
- Use the value attribute (not between elements as text).
- Use the choice data types correctly.
- Use repeating element correctly (JSON needs arrays even for repeatable things that have only repetition).
- Use a backbone element correctly.
- Use quotes correctly for strings but not for

Use search parameters for resources:

- Given the parameters, write a simple search using parameter name and search criteria data types.
- Use AND/OR to combine search criteria.
- Use special syntax for dates and quantities to include this type of parameters.

Use operations for a use case:

- Use the in/out parameter data types correctly.
- Determine if it is a simple operation (URL based) or POST operation (complex).
- Write the operation call based on the parameters.

Use terminology for a use case:

- Use structure correctly for a basic code data type e.g., Patient.gender.
- Use structure correctly for a more complex CodeableConcept data type.
- Use the different attributes: system, code, display, and text.
- Manage more than one code in a CodeableConcept

Troubleshooting and Validation

Troubleshoot validation errors:
- Determine location of validation error in the evaluated resource.
- Determine suitable corrective action(s) based on the validation error(s).
- Determine if corrective action(s) are needed based on message severity.

Troubleshoot the rest API server errors:
- Interpret the different HTTP code errors.
- Interpret the OperationOutcome response.
- Triage based on the kind of problem (lack of connectivity/server down, permissions failure, incorrect use of API / Wrong URL/wrong HTTP verb, missing data, incorrect data, etc.
- Fetch and read the CapabilityStatement of the API to check that it supports your intent.

Apply profile rules to validate resource instance:
- Choose the method to validate the instance.
- Execute the validation (for example, server to validate).
- Interpret the validation output.

Formal Training

HL7 offers training that can help those preparing for the exam. The test taker might consider one of the following training paths:
- The HL7 FHIR Fundamentals Course (online)
- The HL7 FHIR Foundational Implementer Exam Prep Course (online)
- The HL7 FHIR Certification Bundle (on-demand recordings)

Note: While training is helpful, no training replaces knowledge and application of the specification.

Test Format

The test is closed book, and the question formats are multiple choice. There are 125 questions on the exam. Candidates will have 3.5 hours to complete the exam. There is no penalty for guessing.

Candidates will use their camera and a photo ID to identify themselves for testing purposes. No reading or writing materials are allowed during the test.

Mobile phones, messaging devices, and pagers **must be turned off** (not just silenced) during the test. Use of such devices during the test may result in disqualification.

After the test, you are not allowed to discuss the contents of the test with others.