Open Source Implementations

Introduction

This page lists known open source implementations of the FHIR specification. For a list of running servers, see [Publicly Available FHIR Servers for testing](#).

Reference Libraries

- **JAVA**: [https://github.com/jamesagnew/hapi-fhir](https://github.com/jamesagnew/hapi-fhir) - James Agnew / University Health Network
  - Open-source Java library for quickly creating FHIR Servers and Clients
  - Geared towards adding FHIR capability to existing applications (e.g. it's not a database, it's a library for quickly bolting FHIR on to your own database, or connecting to other FHIR servers as a client)
  - Supports all resource types, most operations, and both xml and json encodings
  - "Tinder" code generator (Maven plugin) creates model objects and clients from Profiles and Conformance statements

- **.NET Client - Firely .NET SDK**: [https://github.com/FirelyTeam/firely-net-sdk](https://github.com/FirelyTeam/firely-net-sdk)
  - Reference client with branches supporting each version of FHIR
  - Supports all resource types, searches, all operations, xml + json
  - Validation, FhirPath, very basic terminology server implementation
  - Available via NuGet
  - Documentation available here [https://docs.fire.ly/projects/Firely-NET-SDK/](https://docs.fire.ly/projects/Firely-NET-SDK/)

- **.NET Server - Spark**: [https://github.com/FirelyTeam/spark](https://github.com/FirelyTeam/spark)
  - Reference server built in parallel with Grahame's to test the specification (actively developed during DSTU2, recent updates to support STU3)
  - Supports all resource types, searches, many operations, xml + json
  - Built in .NET with the .NET reference implementation, WebApi 2.0 library, Mongo DB for storage and search.

- **Java - SMART on FHIR**: [http://docs.smarthealthit.org/](http://docs.smarthealthit.org/)
  - Open source reference implementation of the SMART on FHIR stack in Java and MySql based on MitreId Connect and HAPI FHIR.
  - Incorporates synthetic and de-identified data sets for DSTU2 and STU3, as well data management tools including the SMART Patient Browser and FRED FHIR Resource Editor.
  - Sample applications (DSTU2, STU3, R4), client libraries (JavaScript, Python, Swift), and developer tools such as the FHIR Support Matrix
  - Technical documentation is at [http://docs.smarthealthit.org](http://docs.smarthealthit.org) and code is at [https://github.com/smart-on-fhir](https://github.com/smart-on-fhir)

- **Delphi**: [http://github.com/grahamegrieve/fhirserver](http://github.com/grahamegrieve/fhirserver)
  - De facto reference server (runs at [http://test.fhir.org](http://test.fhir.org) and [http://tx.fhir.org](http://tx.fhir.org))
  - General purpose server that implements all of the specification
  - Development platform: (delphi / windows / mssql)

- **R - fhircrackr**: [https://cran.r-project.org/web/packages/fhircrackr/index.html](https://cran.r-project.org/web/packages/fhircrackr/index.html)
  - Open-source R library for managing FHIR resources
  - Tools for conveniently downloading FHIR resources in xml format and converting them to user specified R data frames
  - Supports all resource types; FHIR version independent

- **Ruby**: [https://github.com/fhir-crucible/fhir_client](https://github.com/fhir-crucible/fhir_client) and [https://github.com/fhir-crucible/fhir_models](https://github.com/fhir-crucible/fhir_models)
  - Open-source Ruby library for quickly creating FHIR applications
  - Geared towards adding FHIR capability to your Ruby applications.
  - Supports R4 (default), STU3, and DSTU2, all resource types, most operations, and both xml and json encodings

- **Python**
  - There is no single reference implementation for Python.
  - **client-py** - a flexible Python client for FHIR servers supporting the SMART on FHIR protocol.
    - Supports R4, STU3, and DSTU2, all resource types, json encodings, some validation
    - "fhirclient", a flexible Python client for FHIR servers supporting the SMART on FHIR protocol
    - Data model classes that handle (de)serialization and allow to work with FHIR data in a Pythonic way
    - "fhir-parser" application to build custom resources
    - Demo Flask App
  - **fhir-py** - async FHIR client for python3.
    - Open-source Python library on top of asyncio network layer.
    - Plain python's dictionaries used as a data structure so the library is agnostic to FHIR version.
    - The library provides Django like style for search parameters [https://github.com/beda-software/fhir-py#searchset-examples](https://github.com/beda-software/fhir-py#searchset-examples)
  - **fhir.resources** - Powered by pydantic python classes with built-in data validation.
    - Open-source Python library implements FHIR Specification in python, each resource represents a python class.
    - Available for R4, STU3, and DSTU2 currently.
  - **fhirpath-py** – a Python implementation of FHIRPath
    - Open-source Python library inspired by fhirpaths.js

- **Swift - SMART on FHIR**
  - Supports R4, all resource types, json encodings, some validation
  - "Swift-SMART": a full client implementation of the FHIR specification for building apps: [https://github.com/smart-on-fhir/Swift-SMART](https://github.com/smart-on-fhir/Swift-SMART)
  - Swift classes representing data models of FHIR elements and resources, compatible with iOS 11 and OS X 10.13 and later
  - "fhir-parser" application to build custom resources

- **PHP - PHP FHIR**

---

This page was last updated on 2023-01-24.
- **Java - IBM FHIR Server**: [http://github.com/ibm/fhir](http://github.com/ibm/fhir)
  - Open-source FHIR server and libraries for manipulating FHIR resources
  - Supports R4 (all resource types), JSON and XML formats, FHIRPath 2.0
  - [Aegis Touchstone Conformance Report](https://www.aegis.com)

- **Dart/Flutter**: [http://github.com/ibm/fhir](http://github.com/ibm/fhir)
  - Open-source Dart/Flutter library for creating mobile FHIR applications
  - [Pub.dev Packages](https://pub.dev) - all packages available for free download and inclusion in all Dart/Flutter projects

  - **fhir** - [GitHub/Pub.dev](https://github.com/tiloc/faiadashu)
    - Base package supports R4, Stu3, Dstu2 and R5 (Preview #3)
    - json encodings/decodings
    - yaml encodings/decodings
    - most validation
    - all packages below also support all of these versions

  - **fhir_auth** - [GitHub/Pub.dev](https://github.com/tiloc/faiadashu)
    - Authentication using SMART on FHIR
    - Also allows authentication with Google, Google Identity Platform and Azure
    - Authentication for AWS Health Lake coming soon

  - **fhir_at_rest** - [GitHub/Pub.dev](https://github.com/tiloc/faiadashu)
    - Helper classes to simplify RESTful requests to FHIR servers

  - **fhir_bulk** - [GitHub/Pub.dev](https://github.com/tiloc/faiadashu)
    - Package for uploading/downloading FHIR bulk files (.ndjson)
    - Also works with .zip (and gzip.tar)

  - **fhir_db** - [GitHub/Pub.dev](https://github.com/tiloc/faiadashu)
    - For storing information securely on a mobile device
    - Build using a locally embedded NoSQL DB on top of SQLite
    - Allows encryption with AES-256 or Salsa20

  - **fhir_uscore** - [GitHub/Pub.dev](https://github.com/tiloc/faiadashu)
    - Collection of the above packages
    - Trimmed down to conform to USCore IG
    - Only available in R4 currently

- **Flutter**: UI add-on SDK to Grey Faulkenberry's Dart/Flutter packages

- **Faiadashu FHIRDash**: [on GitHub](https://github.com/faiadashu) / on [pub.dev](https://pub.dev)
  - Plus some smaller things, such as formatted output of Observations
  - Supports R4
  - MIT License

- **Java - SmartEhr / SmartServer implementations of “EHR Launch” embedded SMART on FHIR apps**
  - [Extensive tutorial and walkthrough](https://github.com/tiloc/faiadashu/blob/main/doc/sdc-capabilities.md) (also see series parts 2, 3, 4)
  - Full code under MIT License: [GitHub](https://github.com/tiloc/faiadashu)
  - Standalone app (no external web server required) or embed into any web framework
  - Minimal dependency (none beyond gson)

---

**JavaScript**

There is no single reference implementation for JavaScript.

- **Client for communicating with FHIR servers**: [https://github.com/FHIR/fhir.js](https://github.com/FHIR/fhir.js)
  - Open-source JavaScript library for communication with FHIR servers

- **Node.js / Express**: [https://github.com/Asymmetrik/node-fhir-server-core](https://github.com/Asymmetrik/node-fhir-server-core)
  - Standalone open-source FHIR server from Asymmetrik [https://www.asymmetrik.com/asymmetrik-fhir/](https://www.asymmetrik.com/asymmetrik-fhir/)
  - Additional resource profiles in heavy development
  - Implements a core FHIR server, with custom adapters for non-FHIR data sources.
  - Example adapter for MongoDB: [https://github.com/Asymmetrik/node-fhir-server-mongo](https://github.com/Asymmetrik/node-fhir-server-mongo) [1]

- **JSON-to-XMl-Converter**: [https://github.com/Asymmetrik/node-fhir-server-core](https://github.com/Asymmetrik/node-fhir-server-core)
  - A Node.js FHIR client library

- **FHIR Kit Create React App** - [Create React App - "Create React App" template for a confidential FHIR ReactJS application](https://github.com/Asymmetrik/node-fhir-server-core)
  - [fhirpaths.js](https://github.com/Asymmetrik/node-fhir-server-mongo) - a JavaScript implementation of FHIRPath
  - [LHC-Forms](https://github.com/Asymmetrik/node-fhir-server-echr) - NLM’s JavaScript form rendering package with support for FHIR Questionnaire and partial support for the Questionnaire’s SDC profile.
  - [NLM's Form Builder](https://github.com/lhncbc/formbuilder-lhcforms) – A builder for forms that can import or export FHIR Questionnaire resources. Open source, at [https://github.com/lhncbc/formbuilder-lhcforms](https://github.com/lhncbc/formbuilder-lhcforms).
  - [NLM’s SMART on FHIR SDC Questionnaire App](https://github.com/lhncbc/formbuilder-lhcforms) - uses LHC-Forms
  - [UCUM-LHC](https://github.com/tiloc/faiadashu) – NLM’s open source library for validating and converting units from the Unified Code for Units of Measure (UCUM).

- **typescript-fhir-types** - Open source Typescript / Javascript object model for FHIR standard Model follows FHIR R4 specifications (description how to use it)

- **Web Application Frameworks (Node)**:
  - **Clinical Meteor** - Opensource web application framework using the Meteor build tool (developed at MIT). Includes an ecosystem of ~200 packages, with one package for each FHIR resource type, containing isomorphic hand-crafted DSTU2 schemas using SimpleSchema and UI components with a handlebars type syntax. Relies on the Data Distribution Protocol for FHIR-over-websockets data subscriptions and optimistic UI. Requires the Atmosphere package manager (Meteor/websockets specific). Somewhat deprecated after 80+ prototypes, after successfully refactoring MeteorOnFHIR into the NodeOnFHIR project. Clinical Meteor is something of a
scrapyard of Node+FHIR projects.

- **NodeOnFHIR** - Successor to the Clinical Meteor and MeteorOnFHIR projects. Modern Node based web stack, supporting Javascript, ECMAscript, CoffeeScript, TypesScript and other languages that compile to a JVM. Supports Cordova and Electron build pipelines for distribution on iOS, Android, and other mobile devices. Includes support for major FHIR libraries found on NPM, including 'fhir-kit-client' and 'fhirclient'. Supports an express based FHIR server; as well as React based FHIR UI components.

  - Example Plugin - NodeOnFHIR is built with a WordPress style plugin architecture. The example plugin demonstrates basic page routing, config files, UI components, etc.
  - Covid19 On FHIR - SMART on FHIR data mining utility for query an FHIR server for COVID19 related data.
  - Vault Server (Freemium) - Express based FHIR Server for the NodeOnFHIR web application framework.

Other Open source Implementations

- **FHIRBase** - open source relational storage for FHIR with document API based on PostgreSQL. FHIRBase is an attempt to take the best parts of Relational & Document Databases for persistence of FHIR resources. FHIRBase stores resources relationally and gives you the power of SQL for querying & aggregating. At the same time FHIRBase provides a set of SQL procedures & views to persist and retrieve resources as a json documents in one hop.
- the FHIR build tool itself is open source and includes various definitional and reasoning tools. See [2].
- FHIR terminology resources are available on Valentia SnoChillies terminology Server: https://valentiatech.snochillies.com/
  - Value Sets
    - $expand
    - $validate-code
  - Code Systems (SNOMED CT, LOINC)
    - $lookup
    - $subsumes
  - Concept Map
    - $translate
    - $closure
  - Capability Statement ( format can in xml or in json )
    - https://valentiatech.snochillies.com/fhir/metadata (as in XML default)
    - https://valentiatech.snochillies.com/ ( GUI )
  - SNOMED browser: https://snochillies.com/ by Abdul Rauf (abdul_rauf_2005@msn.com)
- Patients First NZ Terminology Server - Exemplar FHIR-based Terminology Service, including SNOMED CT NZ Edition & LOINC.
- Synthea - Synthetic patient simulation that generates longitudinal FHIR (R4, STU3, and DSTU2) records suitable for software development, integration, testing, demoing. Over 60 diseases with reasonable comorbidities and treatments. Supports Patient, Condition, Encounter, Observation, DiagnosticReport, Immunization, AllergyIntolerance, MedicationRequest, Procedure, ImagingStudy, Organization, Practitioner, Claim, ExplanationOfBenefit.
- onfhir.io - HL7 FHIR Based secure data repository. Open source, at https://github.com/srdc/onfhir
- Pathling - FHIR server targeted at analytics applications, source code at https://github.com/aehrc/pathling