FHIR FLI

A huge amount of healthcare data is generated each day from a number of sources like hospitals, clinics etc. To perform research, analysis and make the maximum utilization of the health records this data is now digitized. These electronic records require a standardized format to make them easily available across the healthcare ecosystem. FHIR (Fast Healthcare Interoperability Resources) Specification is used to standardize the format for exchanging healthcare data. Patient data related to health can also be retrieved from a number of other sources like gyms, wearable devices and health applications. They contain the following data about an individual:

- Daily observation such as the count of steps, heart rate, BMI etc
- Various activities such as swimming, bicycling etc performed by the user; duration of activities and the calories burned during the activity
- The daily sleep patterns and levels of sleep

FHIR has been commonly used to exchange medical patient data but has not been taken advantage of to share *lifestyle and fitness* data. Our aim is to transform the data from the health applications like Google Fit, HealthKit and Fitbit into the FHIR format.

In this article, we consider the Sleep log data from Fitbit for conversion to FHIR. The data below shows a part of the response received from Fitbit when an API request for sleep log for a particular date is made.

```
{ "sleep": [
        {
            "dateOfSleep": "2017-04-02",
            "duration": <value in milliseconds>,
            "efficiency": <value>,
            "isMainSleep": true,
            "levels": {
                "summary": {
                    "deep": {
                        "count": <value>,
                        "minutes": <value>,
                        "thirtyDayAvgMinutes": <value>
                    },
                    "light": {
                        "count": <value>,
                        "minutes": <value>,
                        "thirtyDayAvgMinutes": <value>
                    },
                    "rem": {
                        "count": <value>,
                        "minutes": <value>,
                        "thirtyDayAvgMinutes": <value>
                    },
                    "wake": {
                        "count": <value>,
                        "minutes": <value>,
                        "thirtyDayAvgMinutes": <value>
                    }
                }
        ]}
```

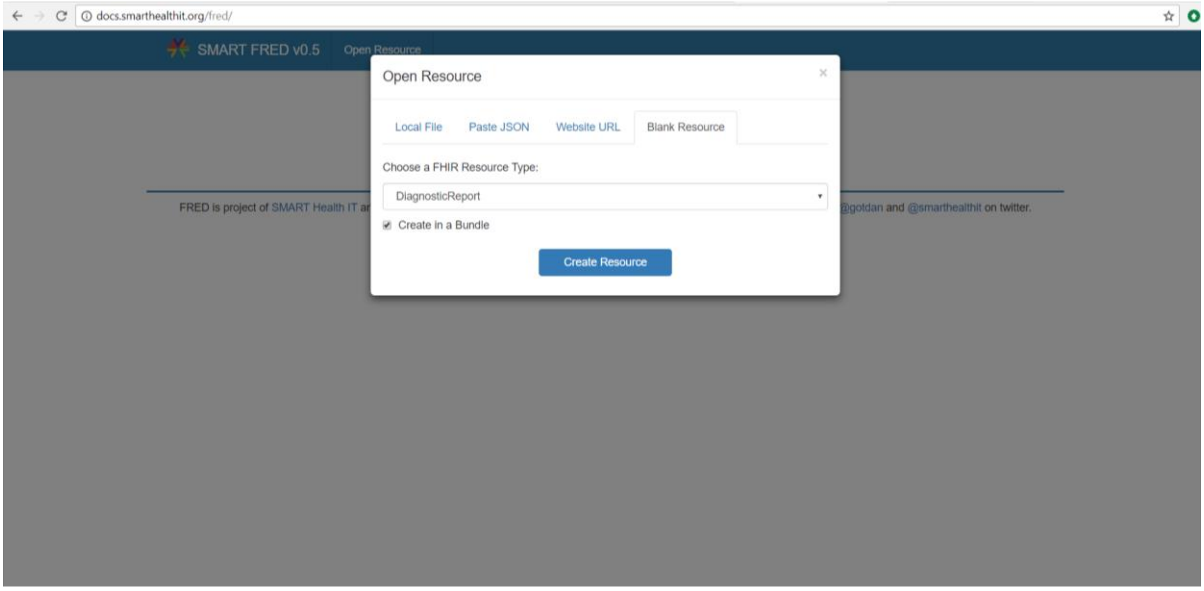
As seen, the response consists of the various attributes of sleep like the date and duration. The sleep log also consists of nested attributes such as the levels of sleep and their respective attributes.

Resources are used in FHIR to support granular concepts. These resources can be grouped together to represent a larger concept. The data above can be represented in FHIR using a bundle [https://www.hl7.org/fhir/bundle.html]of the resource 'Observation' which is commonly used to capture laboratory data, vital signs and clinical assessments. [https://www.hl7.org/fhir/DSTU2/observation.html] A resource called 'DiagnosticReport' is used to logically group these observation resources in the bundle.

Procedure to create sleep log :

The FHIR sleep log was created using a tool known as FRED (FHIR Resource Editor). These resources are json files that can also be created manually.

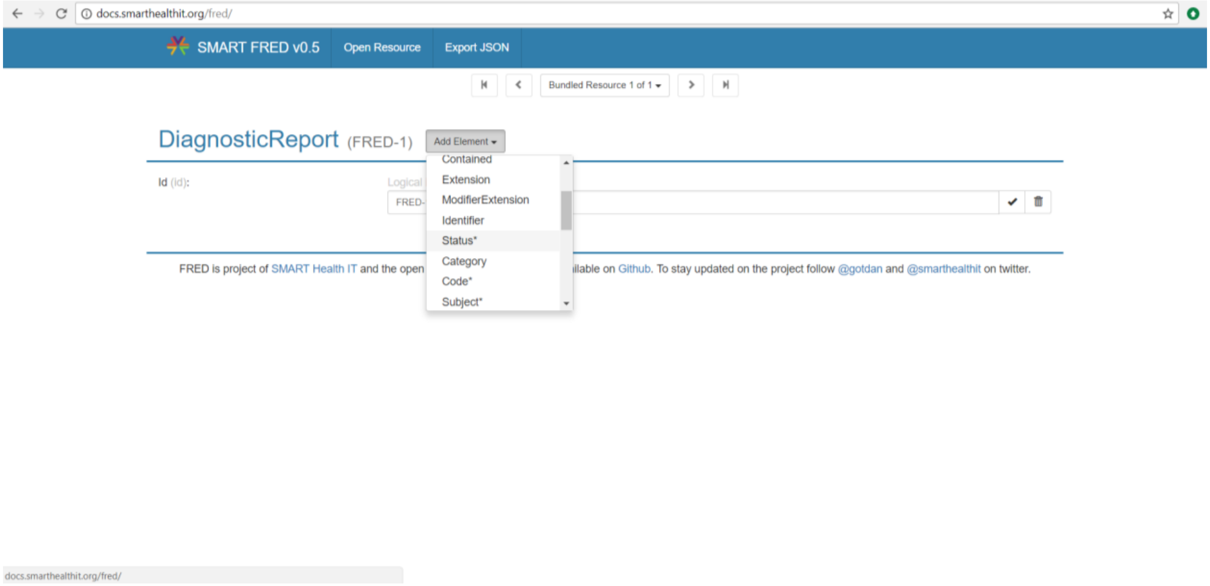
1. The first step is to create a new DiagnosticReport resource for a bundle of Observations. Go to 'Blank Resource' under the 'Open source' tab. Choose 'DiagnosticReport' as the resource and make sure to select 'Create in a Bundle'.


docs.smarthealthit.org/fred/
SMART FRED v0.5
Open Resource
Open Resource
×
Local File
Paste JSON
Website URL
Blank Resource
Create Resource
FRED is project of SMART Health IT ar
@gotdan and @smarthealthit on twitter.

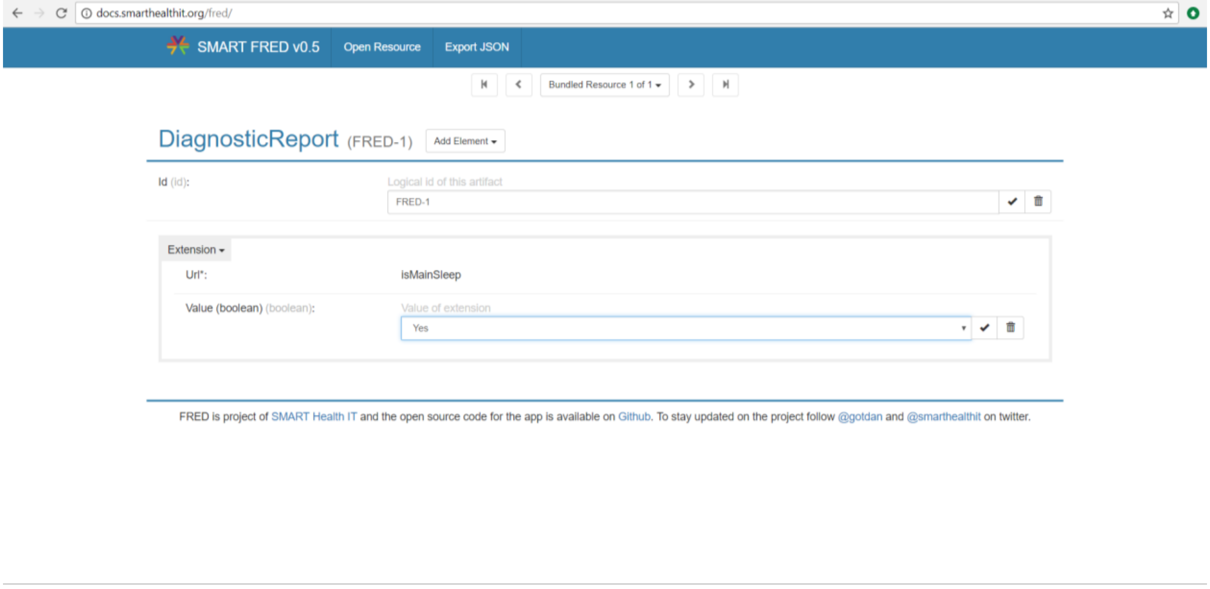
Choose a FHIR Resource Type: DiagnosticReport

☑ Create in a Bundle

2. Go to the ‘Add Element’ option and add necessary elements to the resource such as status, code, category, subject etc.


docs.smarthealthit.org/fred/
SMART FRED v0.5
Open Resource
Export JSON
Bundled Resource 1 of 1
DiagnosticReport (FRED-1)
Add Element
Contained
Extension
ModifierExtension
Identifier
Status*
Category
Code*
Subject*
Id (id):
Logical
FRED-
FRED is project of SMART Health IT and the open …ilable on Github. To stay updated on the project follow @gotdan and @smarthealthit on twitter.
docs.smarthealthit.org/fred/

3. Next, we add an extension for ‘isMainSleep’ which has a Boolean value. This element corresponds to the ‘isMainSleep’ attribute in the fitbit data.

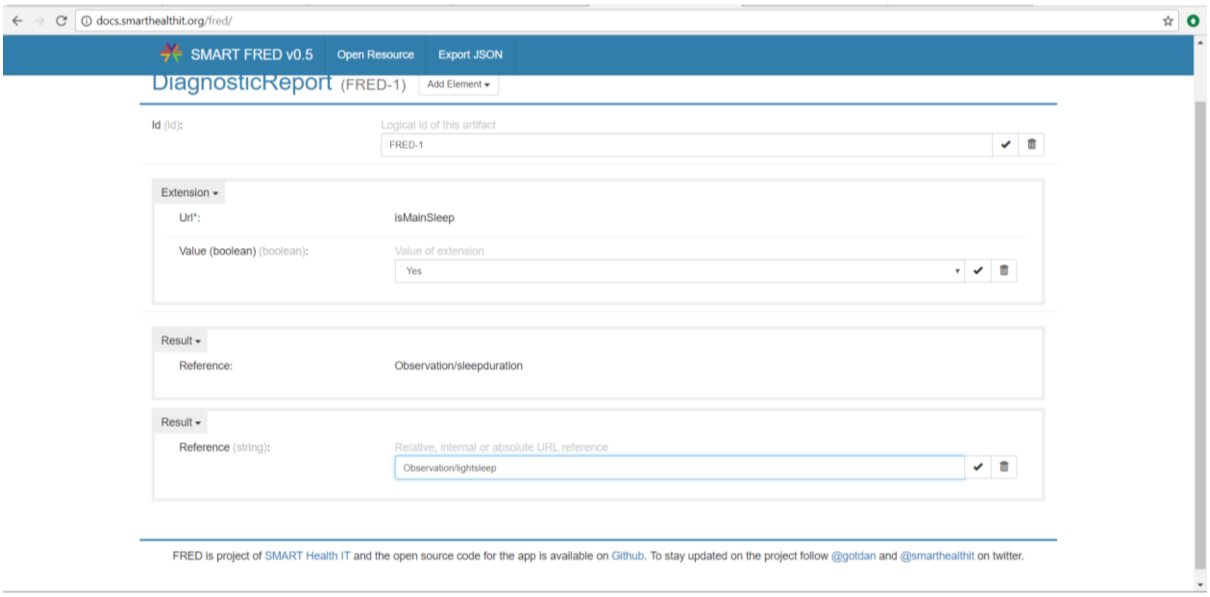

docs.smarthealthit.org/fred/
SMART FRED v0.5
Open Resource
Export JSON
Bundled Resource 1 of 1
DiagnosticReport (FRED-1)
Add Element
Extension
FRED is project of SMART Health IT and the open source code for the app is available on Github. To stay updated on the project follow @gotdan and @smarthealthit on twitter.

Id (id): FRED-1

Url*: isMainSleep

Value (boolean) (boolean): Yes

4. For the other attributes such as sleep duration and levels, we add an element ‘result’ which refers to the other observations in the bundle.



DiagnosticReport (FRED-1)

Id (id): Logical id of this artifact: FRED-1

Url*: isMainSleep

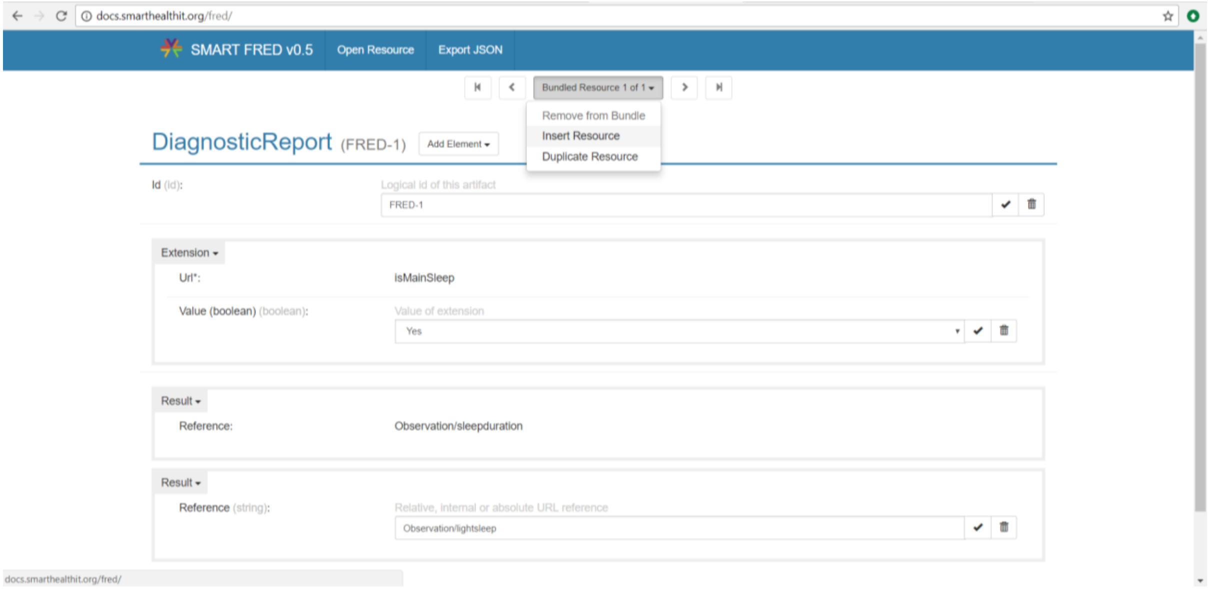
Value (boolean) (boolean): Value of extension: Yes

Reference: Observation/sleepduration

Reference (string): Relative, internal or absolute URL reference: Observation/lightsleep

FRED is project of SMART Health IT and the open source code for the app is available on Github. To stay updated on the project follow @gotdan and @smarthealthit on twitter.

5. The next step is to insert Observation resources in the bundle. Each of these resources correspond to the attributes like sleep levels (light, rem, deep, wake) and sleep duration.



DiagnosticReport (FRED-1)

Id (id): Logical id of this artifact: FRED-1

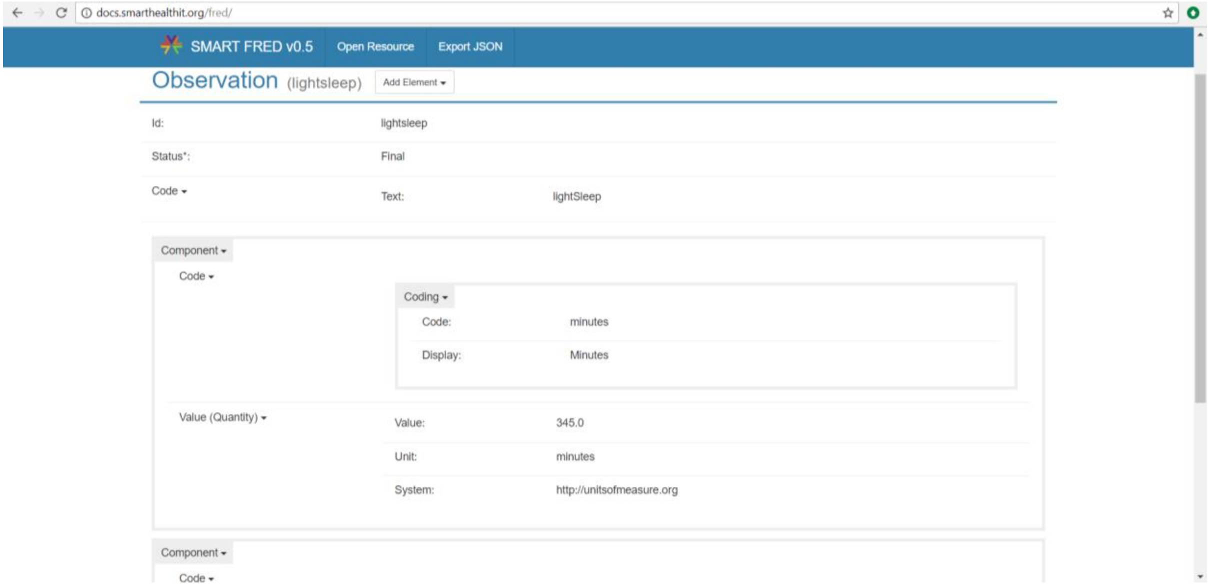
Url*: isMainSleep

Value (boolean) (boolean): Value of extension: Yes

Reference: Observation/sleepduration

Reference (string): Relative, internal or absolute URL reference: Observation/lightsleep

6. Lets consider the Observation for light sleep in the bundle. We add the element ‘component’ in the resource to represent the count and the minutes for each sleep level. The value and unit inside each component represents the actual reading and its unit. Other necessary elements can also be added.



Observation (lightsleep)

Id: lightsleep

Status*: Final

Code ▾ Text: lightSleep

Component ▾

Code ▾

Coding ▾

Code: minutes

Display: Minutes

Value (Quantity) ▾ Value: 345.0

Unit: minutes

System: http://unitsofmeasure.org

Component ▾

Code ▾

In this way, a number of nested attributes can be represented as multiple Observations. Once the bundle is complete, the file can be exported as JSON.

This is just one example of representing the lifestyle data using the FHIR specification. A lot of other data from health apps can be made compatible with FHIR. For more profiles on lifestyle data, please visit our profile[https://simplifier.net/FhirFli/~resources] on simplifier.net . This profile was created by the guidance received from experts and implementers on the FHIR chat (Zulip) by FHIR Foundation. [https://chat.fhir.org/#narrow/stream/implementers/topic/Fitness.20data]