

# Package ‘BiocFHIR’

November 14, 2024

**Title** Illustration of FHIR ingestion and transformation using R

**Version** 1.8.0

**Date** 2022-11-29

**Description** FHIR R4 bundles in JSON format are derived from <https://synthea.mitre.org/downloads>. Transformation inspired by a kaggle notebook published by Dr Alexander Scarlet, <https://www.kaggle.com/code/drscarlat/fhir-starter-parse-healthcare-bundles-into-tables>. This is a very limited illustration of some basic parsing and reorganization processes. Additional tooling will be required to move beyond the Synthea data illustrations.

**License** Artistic-2.0

**Encoding** UTF-8

**Depends** R (>= 4.2)

**Imports** DT, shiny, jsonlite, graph, tidyr, visNetwork, dplyr, utils, methods, BiocBaseUtils

**Suggests** knitr, testthat, rjsoncons, igraph, BiocStyle

**VignetteBuilder** knitr

**biocViews** Infrastructure, DataImport, DataRepresentation

**RoxygenNote** 7.2.2

**URL** <https://github.com/vjcitn/BiocFHIR>

**BugReports** <https://github.com/vjcitn/BiocFHIR/issues>

**git\_url** <https://git.bioconductor.org/packages/BiocFHIR>

**git\_branch** RELEASE\_3\_20

**git\_last\_commit** e6712b7

**git\_last\_commit\_date** 2024-10-29

**Repository** Bioconductor 3.20

**Date/Publication** 2024-11-14

**Author** Vincent Carey [aut, cre] (<<https://orcid.org/0000-0003-4046-0063>>)

**Maintainer** Vincent Carey <stvjc@channing.harvard.edu>

## Contents

add_procedures	2
allin	3
available_retention_schemas	3
build_procond_igraph	4
display_procond_igraph	4
FHIRtabs	5
FHIR_ResourceTypes	5
FHIR_retention_schemas	6
freq_app	6
getHumanName	7
make_condition_graph	7
make_test_json_set	8
print.BiocFHIR.FHIRgraph	8
print.FHIR.bundle	9
process_AllergyIntolerance	9
process_CarePlan	10
process_Claim	10
process_Condition	11
process_Encounter	11
process_fhir_bundle	12
process_Immunization	13
process_MedicationRequest	13
process_Observation	14
process_Patient	14
process_Procedure	15
stack_fhir	16
summarise_bundles	16
summary_selections	17
<b>Index</b>	<b>18</b>

---

add_procedures	<i>update a fhir graph on patients and conditions with procedures</i>
----------------	---

---

### Description

update a fhir graph on patients and conditions with procedures

### Usage

```
add_procedures(fhirgraph, listOfProcessedBundles)
```

### Arguments

```
fhirgraph      instance of BiocFHIR.FHIRgraph
listOfProcessedBundles
                list
```

### Value

instance of BiocFHIR.FHIRgraph

**Examples**

```
data("allin")
g <- make_condition_graph(allin)
g <- add_procedures(g, allin)
g
```

---

allin	<i>collection of synthea FHIR documents ingested</i>
-------	--

---

**Description**

collection of synthea FHIR documents ingested

**Usage**

```
data("allin", package = "BiocFHIR")
```

**Format**

list

**Examples**

```
data("allin", package="BiocFHIR")
allin[[1]]
```

---

available_retention_schemas	<i>list available 'retention schemas'</i>
-----------------------------	---

---

**Description**

list available 'retention schemas'

**Usage**

```
available_retention_schemas()
```

**Value**

character vector

**Examples**

```
available_retention_schemas()
```

build\_prococond\_igraph *build graph with patients, conditions and procedures*

---

**Description**

build graph with patients, conditions and procedures

**Usage**

```
build_prococond_igraph(listOfBundles)
```

**Arguments**

listOfBundles list of processed FHIR bundles, processed with ‘process\_fhir\_bundle‘

**Value**

instance of visIgraph from visNetworks

**Examples**

```
data("allin")
build_prococond_igraph( allin )
```

---

display\_prococond\_igraph  
*make network visualization*

---

**Description**

make network visualization

**Usage**

```
display_prococond_igraph(igraph)
```

**Arguments**

igraph instance of igraph produced by ‘build\_prococond\_igraph‘

**Value**

visIgraph instance

**Examples**

```
data("allin")
g <- build_prococond_igraph( allin )
if (interactive()) {
  display_prococond_igraph( g )
}
```

---

FHIRtabs	<i>table app</i>
----------	------------------

---

**Description**

table app

**Usage**

```
FHIRtabs()
```

**Value**

No value returned.

**Examples**

```
if (interactive()) {  
  FHIRtabs()  
}
```

---

FHIR_ResourceTypes	<i>FHIR Resource types recognized in package</i>
--------------------	--

---

**Description**

FHIR Resource types recognized in package

**Usage**

```
FHIR_ResourceTypes()
```

**Value**

vector of strings

**Examples**

```
FHIR_ResourceTypes() # Oct 2022
```

---

FHIR\_retention\_schemas

*collection of FHIR Resource components to be retained*

---

**Description**

collection of FHIR Resource components to be retained

**Usage**

```
FHIR_retention_schemas()
```

**Value**

list of vectors of strings

**Examples**

```
FHIR_retention_schemas() # Oct 2022
```

---

freq\_app

*produce interactive tables with FHIR resources from a list of ingested bundles*

---

**Description**

produce interactive tables with FHIR resources from a list of ingested bundles

**Usage**

```
freq_app(blist)
```

**Arguments**

blist            list of ingested bundles

**Value**

side-effects of shiny app invocation

**Examples**

```
if (interactive()) {  
  tset = make_test_json_set()  
  bl = lapply(tset, process_fhir_bundle)  
  freq_app(bl)  
}
```

---

getHumanName      *get human name from a BiocFHIR.Patient instance*

---

**Description**

get human name from a BiocFHIR.Patient instance

**Usage**

```
getHumanName(Patient)
```

**Arguments**

Patient      BiocFHIR.Patient instance

**Value**

string with name components concatenated

**Examples**

```
testf <- system.file("json/Vince741_Rogahn59_6fa3d4ab-c0b6-424a-89d8-7d9105129296.json",
  package="BiocFHIR")
tbun <- process_fhir_bundle(testf)
getHumanName(tbun$Patient)
```

---

make\_condition\_graph      *create graph with links from patients to conditions*

---

**Description**

create graph with links from patients to conditions

**Usage**

```
make_condition_graph(listOfProcessedBundles, keep_with_condition_only = TRUE)
```

**Arguments**

listOfProcessedBundles  
list with elements generated by 'process\_fhir\_bundle'  
keep\_with\_condition\_only  
logical(1) omit bundles that lack a "Condition" element, defaults to TRUE

**Value**

instance of BiocFHIR.FHIRgraph

**Examples**

```
data("allin")
make_condition_graph(allin, TRUE)
```

---

```
make_test_json_set      produce 50 json FHIR files in a folder
```

---

**Description**

produce 50 json FHIR files in a folder

**Usage**

```
make_test_json_set(target = paste0(tempdir(), "/jstest"), reuse = TRUE)
```

**Arguments**

target	character(1) a path, defaults to 'jstest' under 'tempdir()'; the contents of synthfir.zip, in inst/zip of BiocFHIR, will be deposited there.
reuse	logical(1) if TRUE, just use what is there, if folder already exists

**Value**

a vector of paths to FHIR JSON, invisibly

**Examples**

```
z <- make_test_json_set()
z[1:3]
```

---

```
print.BiocFHIR.FHIRgraph
      show a combination of graph and patient attributes
```

---

**Description**

show a combination of graph and patient attributes

**Usage**

```
## S3 method for class 'BiocFHIR.FHIRgraph'
print(x, ...)
```

**Arguments**

x	instance of BiocFHIR.FHIRgraph
...	not used

**Value**

print method

**Examples**

```
data("allin")
make_condition_graph(allin)
```



---

```
print.FHIR.bundle      print method
```

---

**Description**

print method

**Usage**

```
## S3 method for class 'FHIR.bundle'
print(x, ...)
```

**Arguments**

x	BiocFHIR FHIR.bundle instance
...	not used

**Value**

print method

---

```
process_AllergyIntolerance
      extract information from retained fields in AllergyIntolerance component of FHIR Bundle, produce simple data.frame
```

---

**Description**

extract information from retained fields in AllergyIntolerance component of FHIR Bundle, produce simple data.frame

**Usage**

```
process_AllergyIntolerance(AllergyIntolerance)
```

**Arguments**

AllergyIntolerance	component of FHIR.bundle instance
--------------------	-----------------------------------

**Value**

data.frame

**Examples**

```
testf <- system.file("json/Vince741_Rogahn59_6fa3d4ab-c0b6-424a-89d8-7d9105129296.json",
  package="BiocFHIR")
tbun <- process_fhir_bundle(testf)
process_AllergyIntolerance(tbun$AllergyIntolerance)
```

---

process_CarePlan	<i>extract information from retained fields in CarePlan component of FHIR Bundle, produce simple data.frame</i>
------------------	---

---

**Description**

extract information from retained fields in CarePlan component of FHIR Bundle, produce simple data.frame

**Usage**

```
process_CarePlan(CarePlan)
```

**Arguments**

CarePlan            component of FHIR.bundle instance

**Value**

data.frame

**Note**

Nov 13 2022, added code to refine the 'category' data processing.

**Examples**

```
testf <- system.file("json/Vince741_Rogahn59_6fa3d4ab-c0b6-424a-89d8-7d9105129296.json",  
  package="BiocFHIR")  
tbun <- process_fhir_bundle(testf)  
process_CarePlan(tbun$CarePlan)
```

---

process_Claim	<i>extract information from retained fields in Claim component of FHIR Bundle, produce simple data.frame</i>
---------------	--

---

**Description**

extract information from retained fields in Claim component of FHIR Bundle, produce simple data.frame

**Usage**

```
process_Claim(Claim)
```

**Arguments**

Claim                component of FHIR.bundle instance

**Value**

data.frame

**Examples**

```
testf <- system.file("json/Vince741_Rogahn59_6fa3d4ab-c0b6-424a-89d8-7d9105129296.json",
  package="BiocFHIR")
tbun <- process_fhir_bundle(testf)
process_Claim(tbun$Claim)
```

---

process_Condition	<i>extract information from retained fields in Condition component of FHIR Bundle, produce simple data.frame</i>
-------------------	--

---

**Description**

extract information from retained fields in Condition component of FHIR Bundle, produce simple data.frame

**Usage**

```
process_Condition(Condition)
```

**Arguments**

Condition	component of FHIR.bundle instance
-----------	-----------------------------------

**Value**

data.frame

**Examples**

```
testf <- system.file("json/Vince741_Rogahn59_6fa3d4ab-c0b6-424a-89d8-7d9105129296.json",
  package="BiocFHIR")
tbun <- process_fhir_bundle(testf)
process_Condition(tbun$Condition)
```

---

process_Encounter	<i>extract information from retained fields in Encounter component of FHIR Bundle, produce simple data.frame</i>
-------------------	--

---

**Description**

extract information from retained fields in Encounter component of FHIR Bundle, produce simple data.frame

**Usage**

```
process_Encounter(Encounter)
```

**Arguments**

Encounter            component of FHIR.bundle instance

**Value**

data.frame

**Examples**

```
testf <- system.file("json/Vince741_Rogahn59_6fa3d4ab-c0b6-424a-89d8-7d9105129296.json",
  package="BiocFHIR")
tbun <- process_fhir_bundle(testf)
process_Encounter(tbun$Encounter)
```

---

process\_fhir\_bundle    *process a bundle of FHIR R4 JSON*

---

**Description**

process a bundle of FHIR R4 JSON

**Usage**

```
process_fhir_bundle(json_file, schemas = FHIR_retention_schemas())
```

**Arguments**

json\_file            character(1) path to text in JSON format  
schemas              list of character vectors defining expected fields, defaults to FHIR\_retention\_schemas()

**Value**

instance of FHIR.bundle, extending list

**Note**

If one encounters the error "Element ... lacks field", the schemas argument can be modified by removing the noted field from the schema.

**Examples**

```
testf = system.file("json/Vince741_Rogahn59_6fa3d4ab-c0b6-424a-89d8-7d9105129296.json",
  package="BiocFHIR")
tbun = process_fhir_bundle(testf)
tbun
```

---

process\_Immunization *extract information from retained fields in Immunization component of FHIR Bundle, produce simple data.frame*

---

**Description**

extract information from retained fields in Immunization component of FHIR Bundle, produce simple data.frame

**Usage**

```
process_Immunization(Immunization)
```

**Arguments**

Immunization      component of FHIR.bundle instance

**Value**

data.frame

**Examples**

```
testf <- system.file("json/Vince741_Rogahn59_6fa3d4ab-c0b6-424a-89d8-7d9105129296.json",
  package="BiocFHIR")
tbun <- process_fhir_bundle(testf)
process_Immunization(tbun$Immunization)
```

---

process\_MedicationRequest  
*extract information from retained fields in MedicationRequest component of FHIR Bundle, produce simple data.frame*

---

**Description**

extract information from retained fields in MedicationRequest component of FHIR Bundle, produce simple data.frame

**Usage**

```
process_MedicationRequest(MedicationRequest)
```

**Arguments**

MedicationRequest  
                    component of FHIR.bundle instance

**Value**

data.frame

**Examples**

```
testf <- system.file("json/Vince741_Rogahn59_6fa3d4ab-c0b6-424a-89d8-7d9105129296.json",
  package="BiocFHIR")
tbun <- process_fhir_bundle(testf)
process_MedicationRequest(tbun$MedicationRequest)
```

---

process\_Observation     *extract information from retained fields in Observation component of FHIR Bundle, produce simple data.frame*

---

**Description**

extract information from retained fields in Observation component of FHIR Bundle, produce simple data.frame

**Usage**

```
process_Observation(Observation)
```

**Arguments**

Observation     component of FHIR.bundle instance

**Value**

data.frame

**Examples**

```
testf <- system.file("json/Vince741_Rogahn59_6fa3d4ab-c0b6-424a-89d8-7d9105129296.json",
  package="BiocFHIR")
tbun <- process_fhir_bundle(testf)
process_Observation(tbun$Observation)
```

---

process\_Patient     *flatten information in Patient component of a bundle to a one-line data.frame*

---

**Description**

flatten information in Patient component of a bundle to a one-line data.frame

**Usage**

```
process_Patient(Patient)
```

**Arguments**

Patient     element of FHIR.bundle instance

**Value**

data.frame

**Examples**

```
testf <- system.file("json/Vince741_Rogahn59_6fa3d4ab-c0b6-424a-89d8-7d9105129296.json",
  package="BiocFHIR")
tbun <- process_fhir_bundle(testf)
tpat <- process_Patient(tbun$Patient)
head(names(tpat))
tags <- c("identifier.system3", "identifier.value3")
tpat[tags,,FALSE]
tags2 <- grep("extension.extension", rownames(tpat), value=TRUE)
tpat[tags2,,FALSE]
```

---

process_Procedure	<i>extract information from retained fields in Procedure component of FHIR Bundle, produce simple data.frame</i>
-------------------	--

---

**Description**

extract information from retained fields in Procedure component of FHIR Bundle, produce simple data.frame

**Usage**

```
process_Procedure(Procedure)
```

**Arguments**

Procedure      component of FHIR.bundle instance

**Value**

data.frame

**Examples**

```
testf <- system.file("json/Vince741_Rogahn59_6fa3d4ab-c0b6-424a-89d8-7d9105129296.json",
  package="BiocFHIR")
tbun <- process_fhir_bundle(testf)
process_Procedure(tbun$Procedure)
```

---

stack_fhir	<i>convert data of a given FHIR type in a list of bundles to a data.frame</i>
------------	---

---

**Description**

convert data of a given FHIR type in a list of bundles to a data.frame

**Usage**

```
stack_fhir(blist, type, droperr = TRUE)
```

**Arguments**

blist	list of FHIR bundles imported with process_fhir_bundle
type	character(1) type, in names(blist[[1]]), e.g.
droperr	logical(1) exclude records for which process_[type] fails, defaults to TRUE

**Value**

data.frame

**Examples**

```
jj = make_test_json_set()
b2 = lapply(jj[1:2], process_fhir_bundle)
ss = stack_fhir(b2, "Procedure")
head(ss, 2)
```

---

summarise_bundles	<i>produce tables summarizing FHIR data</i>
-------------------	---

---

**Description**

produce tables summarizing FHIR data

**Usage**

```
summarise_bundles(
  blist,
  resource = "Condition",
  selection_map = summary_selections()
)
```

**Arguments**

blist	list of ingested bundles
resource	character(1) FHIR resource name
selection_map	character() named vector of single strings selected for summarisation

**Value**

data.frame



---

`summary_selections`     *vector of fields to be selected for summarization*

---

**Description**

vector of fields to be selected for summarization

**Usage**

`summary_selections()`

**Value**

named vector of strings

# Index

## \* datasets

allin, [3](#)

add\_procedures, [2](#)

allin, [3](#)

available\_retention\_schemas, [3](#)

build\_procond\_igraph, [4](#)

display\_procond\_igraph, [4](#)

FHIR\_ResourceTypes, [5](#)

FHIR\_retention\_schemas, [6](#)

FHIRtabs, [5](#)

freq\_app, [6](#)

getHumanName, [7](#)

make\_condition\_graph, [7](#)

make\_test\_json\_set, [8](#)

print.BiocFHIR.FHIRgraph, [8](#)

print.FHIR.bundle, [9](#)

process\_AllergyIntolerance, [9](#)

process\_CarePlan, [10](#)

process\_Claim, [10](#)

process\_Condition, [11](#)

process\_Encounter, [11](#)

process\_fhir\_bundle, [12](#)

process\_Immunization, [13](#)

process\_MedicationRequest, [13](#)

process\_Observation, [14](#)

process\_Patient, [14](#)

process\_Procedure, [15](#)

stack\_fhir, [16](#)

summarise\_bundles, [16](#)

summary\_selections, [17](#)