HL7® FHIR® DevDays 2019 REDMOND

HL7 FHIR

Exercise
Let's build! A FHIR package

Track: Modeling
Session lead: **Ardon Toonstra**
Email: *ardon@fire.ly*

FHIR defines NPM based packages as a solution for proper version management of conformance resources. In this ‘Let’s build!´ session you will get your hands dirty with packages. We will create, publish and consume packages to fully understand the package mechanism.

Tool	Create	Publish	Consume
Simplifier	X	X	-
IG Publisher	X	X	
Torinox	X		X
Java Validator			X
Forge			X
No tool	Text/JSON editor	NPM upload	NPM download

Table 1. Overview of some tools and their package functionality. Definition of publish: upload/host package, available for download. Definition of consume: tool’s ability to pull a package and use it for e.g. validation.

Requirements

- A free Simpifier.net account
- Forge R4
- Torinox for R4
- Knowledge of FHIR Conformance Resources and profiling

Exercises (details on next pages)

1. Consume a package and validate using Torinox R4
2. Create a derived profile with Forge R4 based on the profile in package
3. Create a package using Simplifier
4. Validate against your own package

Materials

- Example FHIR R4 Conformance Resources on Simplifier
- Package name on Simplifier: devdays.r4.example.conformanceresources

ORGANIZERS firely HL7 International

Exercise 1: Consume a package and validate using Torinox R4

Imagine that you are tasked to implement a laboratory results use case described in this dummy implementation guide. You are developer on the laboratory results server, which needs to be exposed to FHIR clients. Your implementation specs are based on the FHIR Conformance Resources in package *DevDays.R4.Example.ConformanceResources 1.0.0-beta* hosted in this Simplifier project.

Step 0: Prerequisites

- Install Torinox R4
- Create a directory for your project

Step 1: Download the package

- Command $> *fhir4 install devdays.r4.example.conformanceresources 1.0.0-beta*

Step 2: Validate instance named 'Example-Invalid-LabObservation.xml'

- Download this example and save it in your directory as LabObservation.xml
- Command $> *Fhir4 validate LabObservation.xml*

Looks like the profile is too strict! Your system does not always have a method qualifier. An issue was filed and a new package was created by the author: version 1.0.0.

Step 3: Download new version of the demo package

- Command $> *fhir4 install devdays.r4.example.conformanceresources 1.0.0*

Step 4: Validate the example again

- Command $> *Fhir4 validate LabObservation.xml*
- The validator should tell you the resource is valid now.

What are the validation results?

Validation might give this 1 incorrect error. Please ignore:
[ERROR] (no details)(further diagnostics: Endpoint returned a body with contentType 'text/plain', while a valid FHIR xml/json body type was expected. Is this a FHIR endpoint?) (at Observation.status[0])

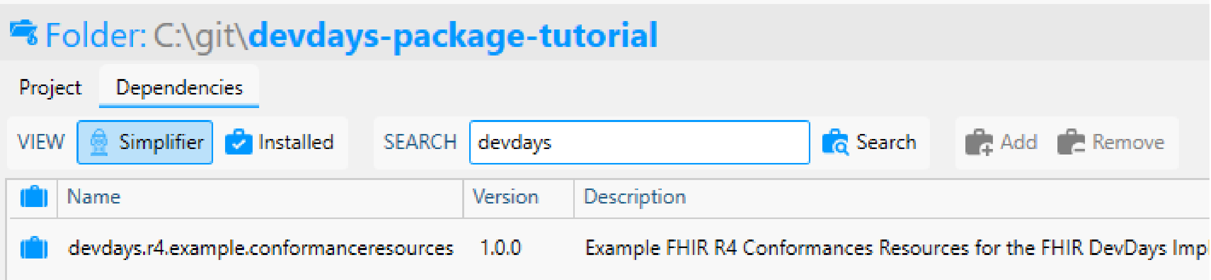
More on the next pages!

Exercise 2: Create a derived profile with Forge R4 based on the profile in package

For you similar system you want to reuse these profiles, however, your system mandates that the Observation.issued element is always filled with a date. So, you want to create a derived profile based on the LabObservation in package version 1.0.0 and set the .issues element cardinality to 1..1.

Step 1: Open Forge and install the package

- Open Forge R4
- Open profile folder → use a (temp) directory for example – same as exercise 1.
- Click on the dependencies tab, click on Simplifier icon and search for devdays, and install the correct package to your profiling folder by selecting it and clicking on 'Add'
-



Folder: C:\git\devdays-package-tutorial

SEARCH devdays

Name	Version	Description
devdays.r4.example.conformanceresources	1.0.0	Example FHIR R4 Conformances Resources for the FHIR DevDays Impl

Step 2: Create derived profile on LabObservation

- Select profile folder
- Click on Project tab
- Select the just downloaded package, double click on it, click on package and select the LabObservation profile
- Right mouse click, and select New Derived profile

Step 3: adjust cardinality of Observation.issues

- Go to profile and select Element Tree
- Select issued element
- Set cardinality to 1..1

Step 4: provide canonical url

- Move to the properties tab.
- Update the url to your liking.
- Save profile to your working director.

More on the next pages!

Exercise 3: Create a package using Simplifier.net

Step 0: Prerequisites

- Sign up or log in to Simplifier
- Create a Simplifier project

Step 1: Upload your derived profile to project

- Go to your project
- Click on upload and select your file

Step 2: Create package dependency

- Go to Dependencies tab
- Click on the green Manage button
- Search for 'DevDays.R4.Example.ConformanceResources'
- Select 1.0.0 version and click add
- Save

Step 3: Create package

- Go to Package tab
- Click on green Create button
- Fill in all the required meta data, like your package name
- Create!

Enjoy your own package!

Exercise 4: Validate against your own package

Either by adding an instance to your project or with Torinox! Remember the previously used instance? It was valid against package version 1.0.0. Let's try it out on your package in these steps:

Step 0: update the canonical URL to your profile

Step 1: Validate with Simplifier

- Upload the example instance to your own project
- Search for the instance in your project and open it
- Click the validate button in the top menu

Step 2: Validate with Torinox

- See exercise 1 for more the details
- Install your package
- Validate this example instance

Have fun, and remember to ask for help if you get stuck!