Resource Authoring

This is the technical documentation that describes what you do to author a resource that will be part of the FHIR specification. There is a also a Design Guide that addresses how resources should be designed.

Note: before attempting to author resources, you MUST be able to successfully run the FHIR build process. You must also sign up to the FHIR Committers Zulip chat at https://chat.fhir.org/#narrow/stream/committers

Background

All FHIR resources have both a lower case name [name], and a Pascal-case name [Name]. For example, for the StructureDefinition resource, the lower case name is structuredefinition and the Pascal-case name is StructureDefinition. Each resource has a sub-directory [name] in the source folder of the FHIR git repository, which contains all the files related to the resource. The build process looks for the following files:

- an excel spreadsheet [name]-spreadsheet.xml that defines the content and behavior of the resource
- several xhtml files that allow additional text documentation to be added for the resource
  - [name]-notes.xml - text documentation that goes below the formal resource definition on the resource page
  - [name]-introduction.xml - text documentation that goes above the formal resource definition on the resource page
  - others yet to be documented
- one or more [name]-(whatever)-example.xml which is an example of the resource (refer to #Example Elements Tab for guidance on naming example files.

Only the first file must exist, though at least one example must exist. Managing examples is discussed further below.

Creating a new resource

Creating a new resource is only done by the FHIR project team once a new resource has been proposed and accepted. This section documents the process that the project team follows. paths are relative to the source directory.

1. test and make sure your local copy of the build process completes without errors (so that if something breaks, you can be confident it's your fault . . .)
2. create the directory [name] in the source directory
3. copy templates/template-spreadsheet.xml to [name]/[name]-spreadsheet.xml
   a. open it and replace "[ResourceOrDataTypeName]" in the first column of the Data Elements tab with [Name])
   b. select an appropriate w5 category (e.g. clinical.general) (see source/w5.ini for a list of categories)
4. copy templates/template-notes.xml to [name]/[name]-notes.xml
5. copy templates/template-introduction.xml to [name]/[name]-introduction.xml
6. copy templates/template-example.xml [name]/[name]-example.xml
7. add the new directory and its files to git
8. edit fhir.ini
   a. add [name]=[Name] to the [resources] section
   b. add [name]=committee to workgroups section
   c. add [name]=0 to fmm section
   d. add a [name]=<securitycategory> to the security-categorization section, where <securitycategory>can be one of business, anonymous, patient, not-classified
   e. add [Name]=tla the tlas section
   f. add [name]=an entry in the security-categorization section of fhir.ini for the resource
9. edit hierarchy.xml and add your page under the correct place within the site's navigation
10. open compartments.xml with Excel and specify which search parameters can be used to place the resource in a Patient or Practitioner compartment (or leave empty if N/A)
11. if your resource has an element named "status", open status-codes.xml with Excel and add the appropriate mappings to the status. Use other resources in the same category (definition, request, event) as a pattern for the mappings.
12. add your resource to resourcelist.html and /resourceguide.html (both in the right category and under the right caption letter)
13. add your resource as appropriate in source/administration-module.html, source/clinicalreasoning-module.html, source/clinicalsummary-module.html, source/conformance-module.html, source/diagnostics-module.html, source/financial-module.html, source/foundation-module.html, source/implsupport-module.html, source/medication-module.html, source/ontology-module.html, source/secpriv-module.html, source/terminology-module.html, and/or source/workflow-module.html
14. add a translation for your resources name to implementations/translations.xml
15. edit the example to fill out the [Name] on the base node and add a <id value="xxx"/> where xxx is what you're going to call the example (usually "example")
16. test and make sure the build completes without errors
17. commit all changes to a git branch, push to GitHub, and create a Pull Request (see https://github.com/hl7/fhir/wiki/Get-Started-with-FHIR-on-GitHub]).

Editing a FHIR resource

Instructions for how to make use of the FHIR resource spreadsheet can be found here. Instructions on using the notes and introduction HTML pages are embedded as comments within the XHTML templates. If you run into issues, ask a question on the FHIR Committers list.