

Requirements - Page 3

"Uniform processing for FHIR requests" (new requirement)

The goal for this requirement is to encapsulate the "core" FHIR request processing steps into a "component" or two so that you're not duplicating things in each one of the FHIR controllers.

For example, the REST API services utilize the [RestControllerHelper](#) to parse a processing result and generate a HTTP response body and status code. The link above should direct you to the new static method I added and integrated with the Patient Service. You can look around in the file to see what the other API services are using. You will notice that validation and the corresponding processing result are in a "new" [validation namespace](#). Basically, I took the code that was in the patient service and extracted it to the new namespace. The validation functionality is provided via [Particle](#), a 3rd party library (this was also in the project when I reviewed the code).

High Level Recommendation

- Update the related "API Services" to utilize the new RestControllerHelper method.
- Add a new RestControllerHelperMethod to support FHIR responses. It can invoke RestControllerHelper.handleProcessingResult, evaluate the "processing result" and form up the FHIR response as needed (if this is practical)

Generate Globally Unique Resource ID for New Resources (new requirement)

OpenEMR needs to utilize and return a unique id for each resource it creates. Typically in other FHIR implementations this is implemented as UUIDv4. OpenEMR has a [PR](#) that got a bit off track, as it started to talk about some kind of mapping tables for UUIDs.

Anyway, the objective is to ensure that OpenEMR generates a "globally" unique identifier for its FHIR resources. The ID needs to be globally unique since the resource could be exported/imported into another FHIR compliant system.

At a high level I think this would require one of the following options:

- migrating the existing patient_data.pid column to a UUID field
- add a new column to patient_data
- create a new table for fhir_metadata that contains the UUID/id and other attributes of interest such as the version id or etag, etc

Validate Resources With Tools\Create validation operation(existing requirements)

OpenEMR has a [FHIR R4 Domain Model Namespace](#). These classes appear to have been generated using <https://github.com/dcarbone/php-fhir>. If the domain models do not have validation methods, I would recommend creating a "validation profile" with the Particle Validator library. You could extend the [BaseValidator](#) or provide a FHIRBaseValidator for extension, if it makes sense to have a separate "FHIR" base class.

Finally, the FHIR controller will need to invoke the validation component when the request is received. You may also want to consider implementing a FHIRControllerBase class possibly an abstract class so that you can specify the request processing workflow for all of the FHIR

controllers. It's most likely going to be . . .

Parse request -> validate request -> convert/map request to OpenEMR data type (probably an array) -> invoke appropriate OpenEMR service to interact with DB

You will need to validate each resource regardless of whether it's arriving at a fhir/<resource> endpoint (fhir/Patient) or it's being passed to fhir/Resource-validate.

Manually Testing APIs (existing requirement)

In addition to manual testing with Postman, cURL, fiddler, etc you can also write unit tests for the controller and service components. Brady and I did run into an issue with the CI server for some of the "nginx" related CI tests. I can continue to look at this issue to see if we can address it so that api tests can run for all "server" environments.

Common Operations (page 6)

I would assume that versioning is not supported at this time.

You may want to review the following sections of the FHIR REST documentation to determine if it's included within your project. In my opinion these are reasonable to include. If the Prefer header is not supported, then we should document the type of response that is returned.

- [Prefer Header](#)
- [Content Type/Encodings](#)
- [FHIR Version Parameter](#)