

# Generation of FHIR dart classes for openIMIS

Joel Staubach

04.08.2022, Mannheim





## Table of contents

1. Motivation
2. How does it work?
3. Limitations
4. Conclusion



## Motivation

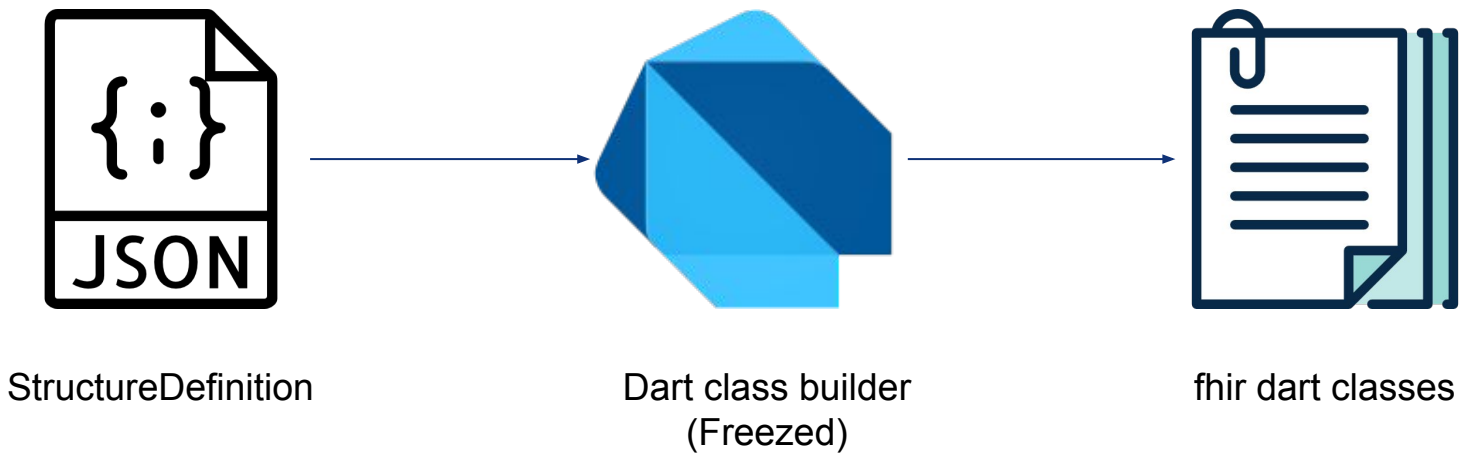
a fhir library for dart already exists, but..

- it has no type safe resource access (e.g. Bundles)
- it runs slow
- it has unnecessary attributes for derived classes (e.g value sets)

⇒ Build your own extensions from the FHIR library

Problem: Creating fhir classes manual is time consuming!

## How does it work?





## How does it work?

Structure of the dart builder:

- `./abstracts` (manual abstract dart data classes)
- `./pre_code` (files defining the outcome for a specific resource)
- `./json` (directory containing all json definitions)
- `./generated` (directory containing all generated files)



## How does it work? - example

```
"concept": [  
  {  
    "code": "UUID",  
    "display": "UUID unique identifier",  
    "definition": "The UUID V4 unique identifier  
that is generated by the openIMIS Database"  
  },  
  {  
    "code": "Code",  
    "display": "Unique code",  
    "definition": "An identifier issued by the  
implementer to identify an openIMIS entity"  
  }  
]
```



## How does it work? - example

```
"compose": {  
  "include": [  
    {  
      "system": "https://openimis.github.io/  
openimis_fhir_r4_ig/CodeSystem/  
openimis-identifiers"  
    }  
  ]  
}
```





## How does it work? - example

```
@frozen
class {RESOURCE_NAME} with _${RESOURCE_NAME} {
  factory {RESOURCE_NAME}({
    @Default(FhirUri.asConst("{DEFAULT_SYSTEM}", ConstUri(
      {DEFAULT_SYSTEM}")), true)) FhirUri system,
    required {CODE_SYSTEM_TYPE} code,
  }) = _{RESOURCE_NAME};

  factory {RESOURCE_NAME}.fromJson(Map<String, dynamic> json) =>
    _${RESOURCE_NAME}FromJson(json);
}
```





## How does it work? - example

```
enum OpenimisIdentifiersCode {  
    @JsonValue("UUID")  
    value_uuid,  
    @JsonValue("Code")  
    value_code,  
    @JsonValue("ACSN")  
    value_acsn,  
}
```



## How does it work? - example

```
@frezed
You, 4 weeks ago | 1 author (You)
class OpenimisIdentifiersCoding with _$OpenimisIdentifiersCoding {
  factory OpenimisIdentifiersCoding({
    @Default(FhirUri.asConst("https://openimis.github.io/openimis_fhir_r4_ig/
CodeSystem/openimis-identifiers", ConstUri("https://openimis.github.io/
openimis_fhir_r4_ig/CodeSystem/openimis-identifiers"),true)) FhirUri
system,
    required OpenimisIdentifiersCode code,
  }) = _$OpenimisIdentifiersCoding;

  factory OpenimisIdentifiersCoding.fromJson(Map<String, dynamic> json) =>
    | _$OpenimisIdentifiersCodingFromJson(json);
}
```



## Limitations

- Documentation of openIMIS fhir does not fit the implementation  
⇒ Workaround: changes in the json definitions itself
- BackboneElements can not (currently) be created dynamically  
⇒ Workaround: custom json defintions



## Conclusion

- Better typing and generic types for bundles
- Clear specification of required and optional parameters
- Automatic creation of all openIMIS FHIR data structures (already existing as a package)
- Less programming effort
- Extension by own generators possible
- Changes to the data structure can be updated immediately without manual changes
- The Project is open source  $\Rightarrow$  documentation available on GitHub!



**Thank you for your attention!**