
Article

[Lorenzo Scalese](#) · Oct 3, 2022 18m de lecture

Créons un profil FHIR à l'aide de SUSHI Partie 1

Bonjour, chers développeurs !

Dans cette série, je ne vous montrerai pas comment utiliser IRIS for Health, mais plutôt comment utiliser SUSHI, un outil de création de profils FHIR, en tant que technologie associée.

Quand on a les outils adéquats, les informations du profil (spécifications, limitations, extensions, etc.) d'un projet FHIR peuvent être bien organisées et publiées.

Avant de commencer, qu'est-ce que le SUSHI ? Je vais l'expliquer brièvement.

Qu'est-ce que le SUSHI ?

FHIR est un acronyme pour Fast Healthcare Interoperability Resources et est défini comme une norme pour l'échange de dossiers médicaux électroniques qui peut être mise en œuvre dans un court délai. REST est une technologie standard de communication web permettant d'échanger un ensemble de données (ressources) au format JSON/XML, très lisible et facile à manipuler.

En simplifiant, l'idée est d'utiliser un format commun pour représenter les données médicales afin de faciliter le transfert et l'échange d'informations entre les systèmes et les établissements.

Il existe plusieurs [ressources](#) définies dans FHIR. Par exemple, les informations sur les patients ont une définition appelée [Ressource Patient](#), qui représente les informations sur les patients.

Il existe de nombreux [modèles](#) sur le site officiel de FHIR, j'ai donc pensé que je pourrais en extraire quelques-uns.

Par exemple, les données sont exprimées au format JSON de la manière suivante. Le numéro du patient (identifiant), son nom, son sexe, etc. sont représentés.

```
{
  "resourceType": "Patient",
  "id": "pat1",
  "texte": {
    "statut": "génééré",
    "div": "<div xmlns=\"http://www.w3.org/1999/xhtml\">\n      \n      <p>Patient Do\nnald DUCK @ Acme Healthcare, Inc. MR = 654321</p>\n    </div>"
  },
  "identifiant": [
    {
      "utilisation": "commune",
      "type": {
        "codage": [
          {
            "système": "http://terminology.hl7.org/CodeSystem/v2-0203",
            "code": "MR"
          }
        ]
      }
    }
  ]
},
```

```
    "système": "urn:oid:0.1.2.3.4.5.6.7",
    "valeur": "654321"
  }
],
"active": true,
"nom": [
  {
    "utilisation": "officiel",
    "nom de famille": "Donald",
    "prénom": [
      "Duck"
    ]
  }
],
"sex": "masculin",
"* snip *"
}
```

Qu'est-ce que le profil FHIR ?

Dans FHIR, il existe des formats de représentation JSON et XML et des règles concernant les informations à répertorier avec le nom de clé JSON, le type de code à utiliser et le type de structure à utiliser. C'est ce qu'on appelle le [Profile FHIR](#).

Le profil est un terme utilisé de différentes manières.

Au sens le plus large :

Un ensemble de définitions pour les contraintes sur les ressources FHIR et les serveurs FHIR. Un artefact qui le représente.

Au sens plus restreint :

Au sens plus restreint, une ressource de conformité est une ressource à laquelle des contraintes spécifiques ont été appliquées.

Dans ce cas, un profil existe pour chaque ressource (par exemple, Profil de Patient, Profil d'observation)..

Pour plus de détails, veuillez regarder la vidéo du [Sommet virtuel japonais 2021] (https://youtu.be/B-B6ge0nHg_) sur les profils FHIR ici. (environ 20 minutes).

Bien que le site officiel FHIR fournisse des spécifications par défaut pour chaque ressource, le degré de liberté dans l'utilisation de chaque ressource est considérable. Malheureusement, le degré de liberté dans l'utilisation de chaque ressource est très important. Il est difficile de réaliser des échanges de données interopérables avec les ressources telles qu'elles sont. Il est donc important de définir de nouvelles règles pour la description des ressources sur la base de l'"accord" préalable. Cet "accord" et ces "règles" sont équivalents au Guide de mise en œuvre et au Profil.

Les directives de mise en œuvre sont pour la plupart rédigées en texte à l'aide de Word, Excel, HTML, etc. L'une des caractéristiques de FHIR est que le profil FHIR lui-même peut également être exprimé à l'aide de ressources FHIR. Par exemple, il est possible d'exprimer les spécifications au format JSON afin que des produits tels que IRIS for Health puissent intégrer les définitions et étendre les fonctions. D'autre part, le profil est écrit au format JSON, qui peut être traité à l'aide de la ressource StructureDefinition de FHIR.

L'Association HL7 des États-Unis a publié IG Publisher, qui génère automatiquement des directives de mise en œuvre à partir de profils. Grâce à cet outil, vous pouvez générer des fichiers HTML de directives de mise en œuvre

au format publié par l'Association HL7. La seconde partie de cet article vous montrera comment procéder.

À titre d'exemple, il s'agit d'une ressource appelée "StructureDefinition" qui représente les conventions de notation pour les Ressources Patient suggérées pour une utilisation standard aux États-Unis, appelée US Core.

([Référence](#))

```
{
  "resourceType" : "StructureDefinition",
  "id" : "us-core-patient",
  "texte" : {
    "statut" : "extensions",
    "div" : "(snip)"
  },
  "url" : "http://hl7.org/fhir/us/core/StructureDefinition/us-core-patient",
  "version" : "3.1.1",
  "nom" : "USCorePatientProfile",
  "titre" : "Profil du patient de base aux États-Unis",
  "statut" : "active",
  "expérimental" : false,
  "date" : "2020-06-27",
  "éditeur" : "Comité de pilotage HL7 US Realm",
  "contact" : [
    {
      "telecom" : [
        {
          "système" : "url",
          "valeur" : "http://www.healthit.gov"
        }
      ]
    }
  ],
  "(snip)"
}
```

Il existe également de nombreuses autres ressources qui décrivent le profil FHIR, comme l' [ImplementationGuide] (<http://hl7.org/fhir/implementationguide.html>) et le [CapabilityStatement] (<http://hl7.org/fhir/capabilitystatement.html>) qui résume la série de fonctions du serveur FHIR

Qu'est-ce que FHIR Shorthand?

Pour créer un profil FHIR, il suffit de compléter la structure JSON présentée ci-dessus ! Cependant, il serait difficile et fastidieux de le faire manuellement. Je ferais probablement une erreur.

Il existe des applications et des outils pour faciliter ce processus, notamment plusieurs produits commerciaux et des options à code source ouvert.

Voir [cette] (<https://confluence.hl7.org/pages/viewpage.action?pageId=35718864#Profile...>) page.

Un exemple est célèbre Forge de Firely aux Pays-Bas, bien que, plus récemment, une ** langue spécifique au domaine pour définir les artefacts FHIR** appelée FHIR Shorthand ([lien](#)) soit également devenue largement utilisée.

FHIR Shorthand est une langue qui vous permet de créer un profil FHIR tout en créant un fichier de définition = fichier FSH(fish) (exemple)

Un exemple de fichier FSH est présenté ci-dessous. Le paramètre [quote](#) contient le nom de ce profil (Profil : Cance.

Alias: LNC = <http://loinc.org>

Alias: SCT = <http://snomed.info/sct>

Profil: CancerDiseaseStatus

Parent: Observation

Id: mcode-cancer-disease-status

Titre: "Statut de la maladie cancéreuse"

Description: "Le jugement qualitatif d'un clinicien sur la tendance actuelle du cancer, par exemple, s'il est stable, s'il s'aggrave (progression) ou s'il s'améliore (répond)."

* ^statut = #ébauche

* l'extension contient un EvidenceType nommé evidenceType 0..*

* extension[evidenceType].valueCodeableConcept de CancerDiseaseStatusEvidenceTypeVS (requis)

* statut et code et sujet et effectif[x] et valeurCodeableConcept MS

* bodySite 0..0

* modèle 0..0

* device 0..0

* referenceRange 0..0

* hasMember 0..0

* composant 0..0

* interprétation 0..1

* sujet 1..1

* basé sur la seule référence (ServiceRequest ou MedicationRequest)

(snip)

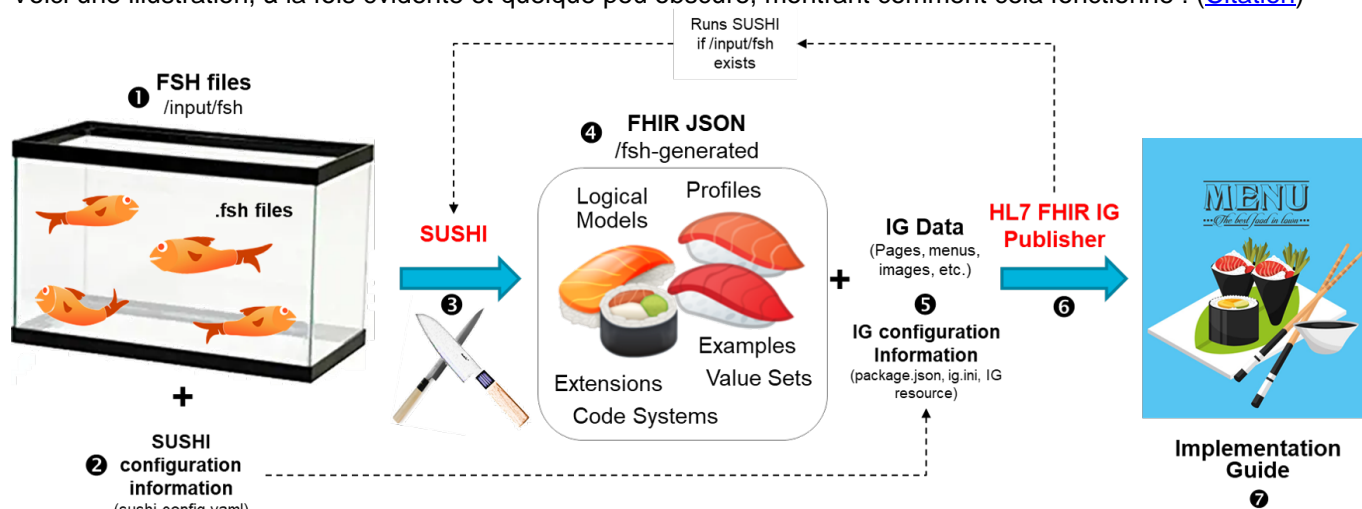
Qu'est-ce que SUSHI?

Après avoir expliqué le profil FHIR/FHIR/FHIR Shorthand, il est temps d'expliquer SUSHI.

SUSHI (acronyme de "SUSHI Unshortens SHorthand Inputs") (4) est une implémentation de référence d'un compilateur FSH qui traduit FSH en artefacts FHIR tels que des profils, des extensions et des ensembles de valeurs. SUSHI est installé sur votre propre ordinateur et s'exécute localement à partir de la ligne de commande ([Reference](#))

En bref, lorsque SUSHI traite le fichier FSH(fish) décrivant le FHIR Shorthand mentionné précédemment, des fichiers tels que StructureDefinition seront générés

Voici une illustration, à la fois évidente et quelque peu obscure, montrant comment cela fonctionne ! ([Citation](#))



(Dans cette image, il est décrit le processus (c'est-à-dire la compilation) de traitement du poisson pour faire du

sushi, mais le compilateur SUSHI fait la compilation, et le produit fini est constitué d'artefacts FHIR tels que des profils, ce qui est un peu différent.)

J'espère qu'après cette introduction, vous comprenez mieux en quoi consiste le SUSHI.

Allons à l'école FSH

Au lieu d'expliquer ici comment installer et utiliser fondamentalement SUSHI, je vous recommande de visiter le site officiel où vous trouverez une introduction très précise.

Le site s'appelle [École FSH] (<https://fshschool.org/>).

Les ressources (fichiers JSON) telles que StructureDefinition sont générées en utilisant SUSHI, et en utilisant l'outil "IG Publisher" également présenté sur ce site, vous pouvez générer la source HTML qui les combine. Vous pouvez également utiliser l'outil "IG Publisher", également présenté sur ce site, pour générer la source HTML pour l'ensemble d'entre eux.

Tout d'abord, nous vous recommandons de suivre le contenu de ce [Tutoriel SUSHI] (<https://fshschool.org/docs/tutorials/basic/>) pour vérifier les fonctions de base.

Si vous avez des problèmes, vous pouvez vous référer au répertoire FishExampleComplete, qui est la version complète du FSH Tank(!) inclus dans le dossier téléchargeable.

Je l'ai testé dans un environnement Windows, et comme je n'avais pas installé Node.js, j'ai utilisé les informations du [site](#) pour m'aider !

De plus, comme le stipule le tutoriel ci-dessous, vous aurez besoin d'un outil appelé Jekyll pour produire des fichiers HTML à l'aide d'IG Publisher.

Avertissement

Avant de passer à la commande suivante : Si vous n'avez jamais exécuté l'IG Publisher, il vous faudra peut-être d'abord installer Jekyll. Consultez la section Installation d'IG Publisher pour plus de détails.

Vous pouvez vous procurer le kit Jekyll sur le [site](#).

Exemple d'exécution de SUSHI

Voici les résultats de l'exécution de la commande SUSHI en utilisant la version complète du tutoriel dans mon environnement.

Pour plus d'informations sur la commande, etc., veuillez consulter ce site ([Running SUSHI](#)).

```
>sushi .
info  Running SUSHI v1.2.0 (implémentation de la spécification FHIR Shorthand v1.1.0)
info  Arguments:
info    C:\Users\kaminaka\Documents\Work\FHIR\SUSHI\fsh-tutorial-
master\FishExampleComplete
info  Aucune route de sortie spécifiée. Sortie vers .
info  Utilisation du fichier de configuration : C:\Users\kaminaka\Documents\Work\FHIR
\SUSHI\fsh-tutorial-master\FishExampleComplete\sushi-config.yaml
info  Importation du texte FSH...
info  Prétraitement de 2 documents avec 3 alias.
info  Importation de 4 définitions et 1 instance.
info  Vérification du cache local pour hl7.fhir.r4.core#4.0.1...
info  Trouver hl7.fhir.r4.core#4.0.1 dans le cache local
info  Paquet téléchargé hl7.fhir.r4.core#4.0.1
```

```
(node:26584) Avertissement : Accès à une propriété inexistante 'INVALID_ALT_NUMBER' d
es exportations du module à l'intérieur d'une dépendance circulaire
(Utilisez `node --trace-warnings ...` pour montrer où l'avertissement a été créé)
(node:26584) Avertissement : Accès à une propriété inexistante 'INVALID_ALT_NUMBER' d
es exportations du module à l'intérieur d'une dépendance circulaire
info Conversion des ressources FSH en ressources FHIR...
info Conversion de 3 StructureDefinitions FHIR.
info Conversion de 1 ValueSets FHIR.
info Conversion de 1 instance FHIR
info Exportation des ressources FHIR en JSON...
info Exportation de 5 ressources FHIR en JSON.
info Assemblage des sources du Guide de mise en œuvre...
info Génération d' ImplementationGuide-fish.json
info Sources du guide de mise en œuvre assemblées ; prêtes pour l'IG Editeur.
```

```
????????????????????????????????????????????????????????????????????????????????
? ??????????????????????????????????????????????????????????????????????????????
? ? Profils ? Extensions ? ValueSets ? CodeSystems ? Instances ? ?
? ??????????????????????????????????????????????????????????????????????????????
? ? 2 ? 1 ? 1 ? 0 ? 1 ? ?
? ??????????????????????????????????????????????????????????????????????????????
? ? ?
????????????????????????????????????????????????????????????????????????????????
? Il n'y a pas mieux que ça ! 0 Erreurs 0 Avertissements ?
????????????????????????????????????????????????????????????????????????????????
```

Une fois exécuté, il compilera le fichier FSH et affichera le nombre de profils et d'extensions générés au cours du processus. Si tout va bien, seul "info" sera imprimé, sinon " Avertissement et erreur " sera affiché. Les messages d'erreur sont relativement simples et permettent de déterminer facilement l'origine du problème.

Après son exécution, vous pouvez confirmer que le fichier JSON de StructureDefinition a été généré avec succès dans le dossier fsh-generated du dossier du projet.

Ensuite, essayons d'obtenir l'outil IG Publisher à l'aide de la commande "updatePublisher", de lancer IG Publisher à l'aide de la commande "genonce", et aussi de générer un groupe de fichiers HTML. Comme le journal d'exécution est long, je vais le sauter.

Après l'exécution, vérifiez le dossier de sortie dans le même dossier de projet, et vous verrez que de nombreux fichiers ont été générés. Et si vous ouvrez le fichier index.html, vous verrez que la page suivante a été développée.

Les pages de description des ressources que vous avez l'habitude de voir sur le site officiel de FHIR sont également générées automatiquement.

Rédigeons un guide de mise en œuvre

Bien que je ne travaille avec cet ensemble d'outils que depuis peu, je voudrais vous montrer comment rédiger un guide de mise en œuvre pour commencer.

Pour plus de détails sur la façon de les utiliser, veuillez vous reporter aux informations figurant sur le site Web de l'école FSH. Les [images FHIR DevDays et autres informations] (<https://fshschool.org/downloads/>) peuvent également vous être très utiles.

Pour commencer, utilisez la commande sushi --init pour créer un modèle pour la structure de votre projet.

```
C:\Users\kaminaka\Documents\Work\FHIR\SUSHI\TestProject>sushi --init
```

```
????????????????????????????????????????????????????????????????
? Cet outil interactif utilisera vos réponses pour créer      ?
? un projet SUSHI fonctionnel configuré avec les             ?
? informations de base de votre projet.                       ?
????????????????????????????????????????????????????????????

Nom (par défaut : ExampleIG) : MonPremierProjetSUSHIP
Id (Par default: fhir.example): myfirstsushi
Canonique (Par default: http://example.org): http://example.org/myfirstsushi
Statut (Par default: ébauche):
Version (Par default: 0.1.0):
Initialiser le projet SUSHI en C:\Users\kaminaka\Documents\Work\FHIR\SUSHI\TestProje
ct\MyFirstSUSHIPProject? [y/n]: y
Téléchargement des scripts de l'éditeur à partir de https://github.com/HL7/ig-
publisher-scripts
(node:13972) Avertissement : Accès à une propriété inexistante 'INVALID_ALT_NUMBER' d
es exportations du module à l'intérieur d'une dépendance circulaire.
(Utilisez `node --trace-warnings ...` pour montrer où l'avertissement a été créé)
(node:13972) Avertissement : Accès à une propriété inexistante 'INVALID_ALT_NUMBER' d
es exportations du module à l'intérieur d'une dépendance circulaire.

????????????????????????????????????????????????????????????????
? Le projet a été initialisé à : ./MyFirstSUSHIPProject      ?
????????????????????????????????????????????????????????????
? Maintenant, essayez ceci :                                  ?
?                                                            ?
? > cd MyFirstSUSHIPProject                                  ?
? > sushi .                                                  ?
?                                                            ?
? Pour des conseils sur la structure et la configuration     ?
? du projet, passez à la documentation SUSHI :              ?
? https://fshschool.org/docs/sushi                          ?
????????????????????????????????????????????????????????????
```

Après l'avoir exécuté, il générera le minimum de fichiers de configuration et de FSH requis.

Maintenant, apportons quelques modifications.

Ensuite, apportons quelques modifications.

Avant cela, je voudrais présenter l'éditeur ; je recommande d'utiliser Visual Studio Code, car il y a une [extension] (<https://marketplace.visualstudio.com/items?itemName=kmahalingam.vscode-l...>) disponible pour modifier les fichiers fsh.

Puisque c'est ce que je vais faire, j'aimerais entrer quelques informations japonaises et voir comment elles sont reflétées.

Tout d'abord, modifiez sushi-config.yaml.

Le titre du guide d'implémentation est ajouté, l'écran de menu est changé en japonais, et la page de la liste des contenus (tuc.html) et la page personnalisée (mycustompage.html) sont ajoutées.

sushi-config.yaml

```
# ??????????????????????????????????????????????????????????????Guide d'implémentation????????????????????????????????????
????????????????????
# ? Les propriétés ci-
```

Les pages d'index et les pages personnalisées peuvent être écrites en markdown :

```
# MyFirstSUSHIProject
```

N'hésitez pas à modifier cette page d'index avec votre propre contenu !

Historique du projet

pagecontent/index.md

Vous pouvez modifier le fichier pour changer la description dans le fichier html.

Pour décrire une page, vous pouvez utiliser la notation markdown.

```
### Liens vers des informations de référence
(omission)
```

mycustompage.md

```
## C'est une page personnalisée.
```

Si vous fournissez un fichier markdown, un fichier html sera généré.

Des pages spécifiques au projet peuvent être produites et incluses dans le guide d'implémentation.

Enfin, nous allons modifier le fichier FSH le plus essentiel. Ce modèle contient le fichier FSH pour le profil Patient, nous l'avons donc modifié légèrement.

patient.fsh

```
// Voici un exemple simple d'un fichier FSH.  
// Ce fichier peut être renommé, et des fichiers FSH supplémentaires peuvent être ajoutés.  
// SUSHI recherchera les définitions dans tout fichier utilisant la terminaison .fsh.  
Profil: MyPatient  
Parent: Patient  
Titre: "Profil du patient pour le projet ??"  
* nom 1..* MS  
// La partie descriptive de la liste peut être modifiée en changeant le ^short.  
* name ^short = "Contient le nom du patient.?"  
* name ^definition = "Élément permettant de stocker le nom du patient. Kanji et Katakana basés sur NeXEHR JP CORE"
```

Exécutez maintenant la commande suivante pour voir la page du guide d'implémentation générée.

sushi.

updatePublisher

genonce

Vous pouvez facilement générer une page qui contient les informations suivantes.

Il génère également des fichiers JSON tels que StructureDefinition et ImplementationGuide.

Résumé

Les résultats vous ont-ils satisfait ?

Cet outil génère des fichiers JSON de profils FHIR et peut créer des fichiers HTML. Ainsi, si cet outil est bien utilisé, il sera facile de créer un contenu capable de communiquer les spécifications FHIR d'une manière facile à comprendre.

Bien que cet outil ne soit pas directement lié à InterSystems, nous avons pensé qu'il serait utile de le présenter à la communauté des développeurs pour échanger des informations sur le projet FHIR.

Pour ceux d'entre vous qui l'ont essayé après avoir lu cet article, ou qui le connaissent déjà, je serais heureux d'entendre vos réflexions sur la façon de l'utiliser, ainsi que toute fonctionnalité ou syntaxe utile que vous souhaiteriez partager.

Dans l'article suivant de cette série, je voudrais étendre les paramètres de recherche en lisant le fichier de définition du paramètre de recherche généré par SUSHI dans IRIS for Health.

(2021/4/20 Nous avons corrigé la partie ambiguë de l'explication, notamment en ce qui concerne le profil. Merci de nous l'avoir signalé.)

[#FHIR](#) [#InterSystems](#) [IRIS for Health](#)

URL de la
source: <https://fr.community.intersystems.com/post/cr%C3%A9ons-un-profil-fhir-%C3%A0-l'aide-de-sushi-partie-1>