Analyse Technique du Guide d'Implémentation OSIRIS FHIR

Seenovate

Cr Lafayette 191-193,
69006 Lyon

Contact

Vincent MARGOT
vincent.margot@seenovate.com
0623576761

SOMMAIRE

I. Vue d'ensemble du projet

OSIRIS FHIR est un Guide d'Implémentation, hébergé sur GitHub, développé pour standardiser le partage de données en oncologie. Ce document présente une analyse technique détaillée du repository ImplementationGuide_OsirisFHIR ainsi que son processus de génération.

II. Architecture du Projet

Le projet se structure de la façon suivante :

ImplementationGuide_OsirisFHIR/

- ▪ input/ # Fichiers source FSH et configurations
- ▪ fsh-generated/ # Fichiers générés par SUSHI
- ▪ output/ # Site web du guide d'implémentation généré
- ▪ input-cache/ # Cache pour le publisher FHIR

III. Environnement de Developpement

1. Installer Java

- Aller sur https://www.java.com/download/
- Télécharger et installer Java Runtime Environment (JRE).
- Vérifier l'installation.
  Commande dans PowerShell :
  java --version

2. Installer Ruby

- Aller sur https://rubyinstaller.org/downloads/
- Télécharger et installer la dernière version Ruby+Devkit (ex : Ruby+Devkit 3.2.X (x64)).
- Pendant l'installation, cocher « Add Ruby executables to your PATH ».
- À la fin de l'installation, garder coché « Run 'ridk install' ».
- Quand l'invite de commande s'ouvre, appuyer sur ENTRÉE pour installer tous les composants.
- Vérifier l'installation.
  Commandes PowerShell :
  ruby -v
  gem -v

3. Installer Jekyll

- Ouvrir PowerShell et exécuter :
  Commande dans PowerShell :
  gem install jekyll bundler

- Vérifier l'installation.
  Commande dans PowerShell :
  jekyll -v

4. Installer Node.js

- Aller sur https://nodejs.org/
- Télécharger et installer la version LTS
- Vérifier l'installation.
  Commande dans PowerShell :
  node -v
  npm -v

Note importante
Si la commande npm -v retourne une erreur de sécurité PowerShell. Ouvrir PowerShell en tant qu'administrateur et exécuter la commande : *Set-ExecutionPolicy RemoteSigned -Scope CurrentUser*

5. Installer SUSHI

- Installer SUSHI puis PowerShell
  Commande dans PowerShell :
  npm install -g fsh-sushi

- Vérifier l'installation.
  Commande dans PowerShell :
  npx fsh-sushi -v

IV. PROCESSUS DE GENERATION

- Cloner le dépôt (en supposant que Git est installé)
  Commande dans PowerShell :
  git clone https://github.com/InstitutNationalduCancer/ImplementationGuide_OsirisFHIR.git
  cd ImplementationGuide_OsirisFHIR

- Générer les ressources FHIR
  Commande dans PowerShell :
  npx fsh-sushi .

- Mettre à jour l'outil de publication du Guide d'Implémentation.
  Commande dans PowerShell :
  ./_updatepublisher.bat

- Générer le Guide d'Implémentation.
  Commande dans PowerShell :
  ./_genonce.bat

- Visualiser le site web.
  Naviguer vers le dossier "output" dans l'Explorateur de fichiers et ouvrir `index.html` dans votre navigateur.

Installation de Git (si nécessaire)
Si vous n'avez pas installé Git :
- Aller sur https://git-scm.com/download/win
- Télécharger et installer Git pour Windows
- Vérifier l'installation
Commande dans PowerShell :
git --version

Résolution des problèmes courants

1. Commandes non reconnues :
   - Redémarrer PowerShell après les installations
   - Si nécessaire, redémarrer Windows
2. Java non reconnu :
   - Rechercher "Variables d'environnement" dans Windows
   - Sous Variables système, trouver "Path"
   - Ajouter le répertoire bin de Java (généralement C:\Program Files\Java\jre[version]\bin)
3. Si SUSHI échoue :

Commande dans PowerShell :
npm uninstall -g fsh-sushi # Désinstaller SUSHI
npm cache clean --force # Nettoyer le cache npm
npm install -g fsh-sushi --force # Réinstaller SUSHI

V. AUTOMATISATION DES TESTS DE DEPLOIEMENT

Cette section détaille le workflow GitHub Actions pour l'automatisation des tests et la génération du site sur GitHub Pages.
Workflow : Workflow Sushi Tests gitHubpages

```yaml
name: Workflow Sushi Tests gitHubpages
on:
  workflow_call:
  push:
  # Allows you to run this workflow manually from the Actions tab
  workflow_dispatch:
jobs:
  run-sushi-tests_gitHubPages:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v3
        with:
          path: igSource
      - uses: ansforge/IG-workflows@v0.2.0
        with:
          repo_ig: "./igSource"
          github_page: "true"
          github_page_token: ${{ secrets.GITHUB_TOKEN }}
          bake: "true"
          nos: "true"
          validator_cli: "true"
          generate_testscript: "false"
          generate_plantuml : "true"
```

Le workflow ci-dessus « fhir-workflows.yml » est déclenché lors de chaque push ou via une exécution manuelle dans l'onglet *Actions*. Il contient les étapes suivantes :

1. Checkout du code.

Utiliser l'action « actions/checkout@v3 » pour cloner le code source dans le répertoire `igSource`.

2. Exécution du Workflow.

Utiliser « ansforge/IG-workflows@v0.2.0 » pour exécuter la génération du guide et publier sur GitHub Pages. Voici les paramètres utilisés :

- repo_ig : spécifie le répertoire source (« ./igSource »).
- github_page : 'true' pour activer la publication GitHub Pages.
- github_page_token : autorise la publication via le token GitHub.
- bake : 'true', active une compilation particulière.
- nos : 'true', génère certaines données statiques.
- validator_cli : 'true', active la validation.
- generate_testscript : 'false', n'inclut pas de scripts de tests.
- generate_plantuml : 'true', génère des diagrammes PlantUML.

VI. Metriques du Site Genere

1. Analyse Automatisée

Notre script de diagnostic analyse les éléments suivants :

- Pages HTML : nombre total et structure.
- Images : utilisation et distribution.
- Liens :
    - Externes (pointant vers d'autres sites).
    - Internes (navigation locale).
    - Morts (non fonctionnels).

2. Rapports Générés

Le script produit plusieurs fichiers CSV :

- pages_[timestamp].csv : Inventaire des pages.
- images_[timestamp].csv : Utilisation des images.
- external_links_[timestamp].csv : Liens externes.
- internal_links_[timestamp].csv : Navigation interne.
- dead_links_[timestamp].csv : Problèmes de liens.
- summary_[timestamp].csv : Synthèse globale.

VII. Processus de Contribution

1. Workflow Git

1. Cloner le référentiel.
2. Créer une branche de fonctionnalité.
3. Effectuer les modifications.
4. Tester localement avec SUSHI.
5. Soumettre un pull request.

2. Tests

1. Générer localement le site.
2. Valider les ressources FHIR.
3. Exécuter le script de diagnostic.
4. Vérifier les liens et la navigation.

www.seenovate.com