

# Criando sua primeira API FHIR: Uma API de pacientes para o e-SUS APS

Italo Macêdo do A Costa<sup>a</sup>

<sup>a</sup>GOInterop Tecnologia, Recife, PE

## Resumo

*Este tutorial visa mostrar como implementar uma API FHIR de pacientes. Para isso, utilizaremos o e-SUS APS PEC como plataforma. Vamos descrever os passos para configuração do ambiente e ferramentas de desenvolvimento. Além disso, vamos explicar a estrutura básica de um projeto de API, mostrando a função que cada pacote exerce para o projeto de um sistema de interoperabilidade. Adicionalmente, vamos ensinar como criar um provedor de recursos e como realizar operações simples e estendidas de busca. O código do projeto utilizado neste tutorial será disponibilizado. Finalmente, buscamos mostrar como é simples e rápido conceber uma API como essa utilizando uma tecnologia que é open source e permite adoção rápida e aderente ao padrão.*

## Palavras-chaves:

Interoperabilidade, FHIR, API.

## Descrição do tutorial

O objetivo deste tutorial é ajudá-lo a criar sua primeira API FHIR em conformidade com padrão do HL7 Internacional seguindo as melhores práticas de desenvolvimento.

Este tutorial é, portanto, voltado para desenvolvedores de soluções de interoperabilidade.

Para acompanhar este tutorial, você precisará de uma máquina Windows 10 com um kit de desenvolvimento Java instalado (preferivelmente o OpenJDK11), de uma instância do e-SUS APS (PEC) instalada no seu computador, do kit de desenvolvimento (HAPI FHIR), além da ferramenta (IDE) de desenvolvimento Eclipse.

## Instrutores do Tutorial

Este tutorial será apresentado por Italo Macêdo. Nascido em Recife – PE. Ele é formado em Ciências da Computação pela Universidade Federal de Pernambuco (UFPE), com Mestrado em Engenharia de Software também pela UFPE e pós-graduação em *Foundations of Health Information Sciences* pela *School of Biomedical Informatics (University of Texas)*. Italo já trabalhou em grandes empresas de Informática em Saúde e hoje é fundador da GOInterop Tecnologia, a primeira editora brasileira de perfis FHIR.

## Palavras-chaves

Interoperabilidade; FHIR; HL7.

## Introdução

Fast Health Interoperability Resources (FHIR) é o mais novo padrão desenvolvido pelo Health Level Seven International (HL7). Esse padrão que já está na sua revisão 4 (R4) permite rápida implementação de Application Programming Interfaces (API) para troca de dados entre sistemas, aplicações e dispositivos e vem causando uma rápida transformação no cenário mundial de interoperabilidade em Saúde.

## Configuração do ambiente

A nossa API FHIR funcionará como sistema de interoperabilidade para o Prontuário Eletrônico do Cidadão (PEC). Portanto, antes de mais nada, precisamos configurar a instância do PEC que conterà os dados da nossa API. Se você ainda não tem PEC instalado, você poderá realizar o download em [1]. Para instruções sobre como realizar a instalação do PEC, você pode seguir as orientações em [2].

Além do PEC, para desenvolver uma API em FHIR, você precisará de um kit (Starter) de desenvolvimento da HAPI FHIR [5]. As orientações para obtenção desse kit e como instalá-lo poderão ser encontradas em [3].

## Template de projeto

Com o intuito de pular algumas etapas de estruturação do projeto, não vamos utilizar o kit de desenvolvimento FHIR em [3]. Isso porque o kit não tem algumas coisas fundamentais, como um conector Java para banco de dados (JDBC).

Portanto, foi disponibilizado um template de projeto em [4] que já contém funcionalidades fundamentais para este tutorial, como o conector JDBC para o PEC, além de Helpers para criar recursos e tradutores de conceitos.

Dessa forma, poderemos focar somente no que é fundamental para a criação dessa API que é a construção dos provedores de dados do paciente.

## Importando o template de projeto para a ferramenta de desenvolvimento

Uma vez concluído o download do projeto template, você poderá descompactá-lo na pasta que você preferir e importá-lo na ferramenta de desenvolvimento. Para realizar essa operação, basta selecionar a opção *Import Project* na tela inicial da ferramenta e selecionar a pasta em que o projeto foi descompactado.

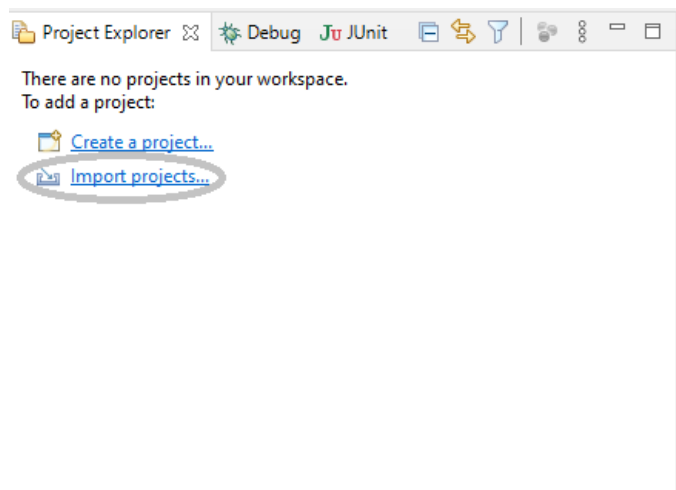


Figura 1 - Importando projetos no Eclipse.

Assim que for concluída a importação, você perceberá a seguinte estrutura de projeto:

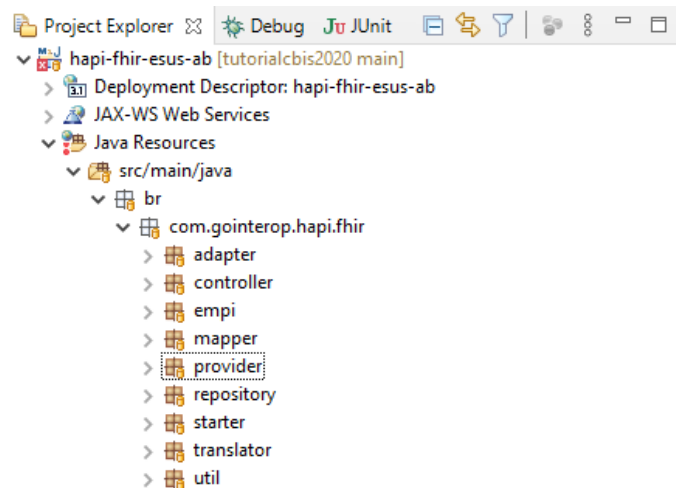


Figura 2 - Estrutura de pacotes do projeto.

## Entendendo a estrutura de pacotes do projeto template

O projeto template criado para acelerar a concepção da API FHIR neste tutorial já entrega uma série de componentes fundamentais para a criação da nossa API (vide Figura 2): Conexão com o banco de dados do PEC em *repository*, mapeamento de campos do paciente com colunas de banco de

dados em *mapper*, tradução de semântica FHIR para a semântica de banco de dados em *translator*, controladores de paciente em *controller*, adaptadores que transformam dados do banco em recursos FHIR em (*adapter*), além de Master Patient Index em *empi* e componentes do kit da HAPI FHIR em *starter*.

## Criando nosso provedor FHIR para pacientes.

Para criar nosso provedor de recursos FHIR para pacientes, basta, dentro do pacote *provider*, criarmos uma classe Java para manipular os dados de requisições para paciente (vide Figura 3).

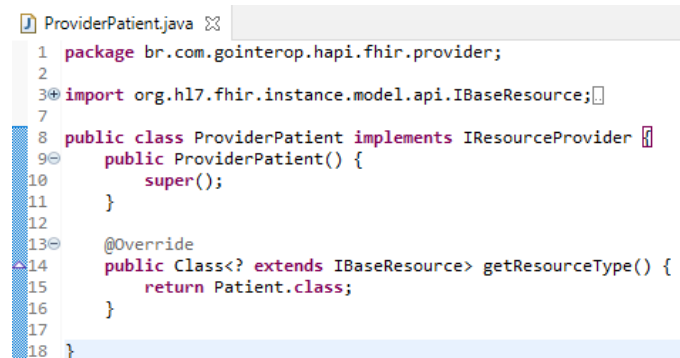


Figura 3 - Exemplo de provedor de recursos.

Um típico provedor de recursos contém uma definição de classe que implementa a interface *IResourceProvider*. Essa interface precisa ter a operação *getResourceType()* redefinida retornando o tipo de recurso FHIR que será implementado. No nosso exemplo, implementaremos o recurso *Patient*.

## Implementando uma operação de busca por Id na nossa API

Agora que nosso provedor de recursos está pronto, podemos implementar uma operação de busca por id na nossa API. Para isso, dentro do nosso provedor de recursos, podemos adicionar anotações que vão interceptar chamadas ao recurso paciente e realizar consultas para responder essas chamadas.



Figura 3 - Implementando uma busca pelo id do Paciente.

A *annotation Read* intercepta operações de leitura no nosso provedor de recursos, enquanto a *annotation IdParam* intercepta os parâmetros do tipo `Id` que foram passados na chamada da busca.

Então a única coisa que precisamos fazer é chamar o nosso controlador de pacientes utilizando a operação *readById()* que recebe a `id` do paciente que foi solicitado. O controlador retornará os dados do Paciente em conformidade com a FHIR R4.

Portanto, na sua máquina, se você disparar uma requisição em <http://localhost/hapi-fhir-esus-ab/fhir/Patient/1>, sua API FHIR fará a consulta por pacientes com `id` iguais a 1 e retornará o resultado no seu navegador.

### Implementando busca estendidas na nossa API

Para aumentar um pouco o nível de dificuldade, podemos criar agora uma busca estendida na nossa API. A busca estendida recebe um conjunto de parâmetros e realiza um filtro compreensivo de resultados baseado nesses parâmetros de entrada.

A busca estendida deve ser realizada utilizando a *annotation Search*. Sua operação de busca pode receber um variado número de parâmetros. Cada parâmetro deve ser declarado usando a *annotation OptionalParam* para parâmetros opcionais e *RequiredParam* para parâmetros obrigatórios. O binding desses parâmetros deverá ser realizado com o tipo respectivo de parâmetro, podendo ser: *StringParam* para texto, *DateParam* para datas e *NumberParam* para números.

Os parâmetros recebidos devem ser inseridos em um *HashMap*. Depois, é só chamar o controlador de pacientes, que foi disponibilizado junto com o template do projeto, utilizando a operação *search()*. Essa operação recebe o *HashMap* de parâmetros de busca e retorna dados do Paciente também em conformidade com a FHIR R4.

### Resultados

Neste tutorial, disponibilizamos uma estrutura básica de projeto para uma API FHIR, explicando a função de cada um dos pacotes e o papel que exercem dentro do projeto de sistema de interoperabilidade.

Além disso, ensinamos como criar e registrar um provedor de recursos, como implementar operações de buscas simples e estendidas, além de controlar a opcionalidade e obrigatoriedade dos parâmetros de busca.

### Conclusão

Vimos neste tutorial como é rápido criar uma API FHIR para uma aplicação. Além disso, mostramos como é simples e de fácil compreensão a sua estrutura de pacotes. Adicionalmente,

apresentamos como é fácil desenvolver novas funcionalidades para a sua API.

O objetivo desse tutorial é, portanto, incentivá-los a adotar essa tecnologia que é *open source* e promove imediata adoção aderente ao padrão.

### Referências

- [1] Portal da Secretaria de Atenção Primária. Página de download do e-SUS APS (PEC). <https://aps.saude.gov.br/ape/esus/download>. Último acesso em 12/11/2020.
- [2] Portal da Secretaria de Atenção Primária. Página de perguntas frequentes do e-SUS APS (PEC). <https://aps.saude.gov.br/ape/esus/perguntasfrequentas>. Último acesso em 12/11/2020.
- [3] Get Started - HAPI FHIR Documentation. [https://hapifhir.io/hapi-fhir/docs/server\\_jpa/get\\_started.html](https://hapifhir.io/hapi-fhir/docs/server_jpa/get_started.html). Último acesso em 12/11/2020.
- [4] Template de projeto de API FHIR para tutorial no CBIS 2020. <https://github.com/italomacedo/tutorialcbis2020>. Último acesso em 12/11/2020.
- [5] HAPI FHIR: The open source FHIR API for Java. <https://hapifhir.io/>. Último acesso em 12/11/2020.